2025 IEEE International Conference on Robotics and Automation (ICRA)
May 19-23, 2025. Atlanta, USA

Updating Robot Safety Representations Online from Natural Language Feedback

Leonardo Santos[1*], Zirui Li[2*], Lasse Peters[3], Somil Bansal[4†], Andrea Bajcsy[5†]

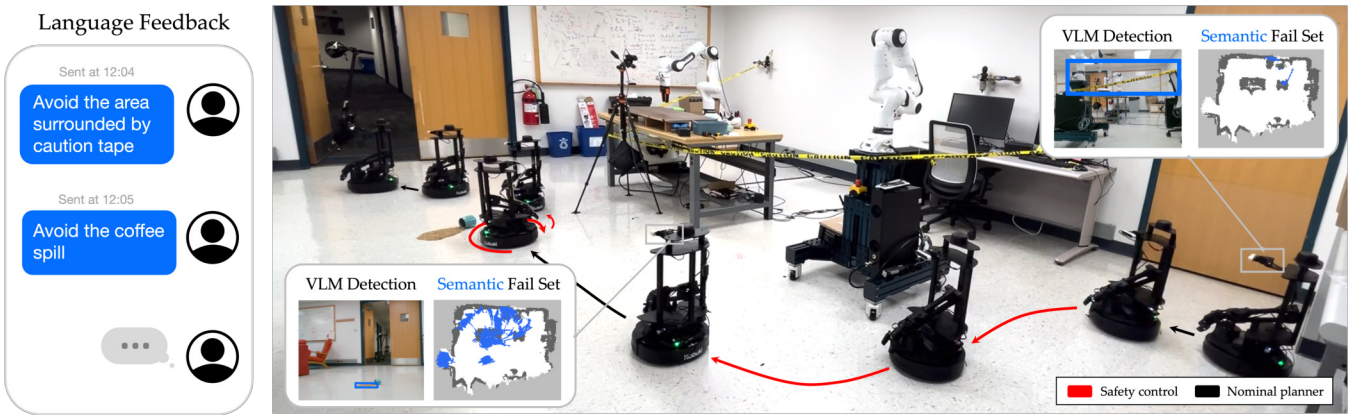

Fig. 1: Natural language provides an intuitive interface for people to specify constraints they care about online, like restricted areas behind caution tape or coffee spills. We leverage advances in vision-language models to interpret multimodal language and image data, infer semantically-meaningful constraints, and update robot safety controllers online. Video results and code at the project website: https://cmu-intentlab.github.io/language-informed-safe-navigation/.

***Abstract*— Robots must operate safely when deployed in novel and human-centered environments, like homes. Current safe control approaches typically assume that the safety constraints are known *a priori*, and thus, the robot can precompute a corresponding safety controller. While this may make sense for some safety constraints (*e.g.*, avoiding collision with walls by analyzing a floor plan), other constraints are more complex (*e.g.*, spills), inherently personal, context-dependent, and can only be identified at deployment time when the robot is interacting in a specific environment and with a specific person (*e.g.*, fragile objects, expensive rugs). Here, language provides a flexible mechanism to communicate these evolving safety constraints to the robot. In this work, we use vision language models (VLMs) to interpret language feedback and the robot's image observations to continuously update the robot's representation of safety constraints. With these inferred constraints, we update a Hamilton-Jacobi reachability safety controller online via efficient warm-starting techniques. Through simulation and hardware experiments, we demonstrate the robot's ability to infer and respect language-based safety constraints with the proposed approach.**

I. Introduction

As robots are increasingly integrated into human environments, ensuring their safe operation is critical. Designing safe controllers for robots is a well-studied problem in robotics; however, the current approaches often assume that the safety constraints are known in advance, and thus, a safety controller can be synthesized offline. While this approach may be effective for static and well-defined constraints (e.g., walls or fixed obstacles), it is insufficient in complex, human-centered environments, where safety requirements are often personalized and context-dependent. For example, one may not want a cleaning robot to drive through a workout area during exercise, and a warehouse robot should avoid entering areas temporarily blocked with caution tape (Figure 1).

In such cases, language provides a flexible communication channel between the robot and the operator who can easily describe constraints they care about (*e.g.*, "Avoid the area surrounded by caution tape"). In this work, we develop a framework for updating robot safety representations *online* through such natural language feedback. Our key idea is that pre-trained open-vocabulary vision-language models (VLMs) are not only a useful interface for constraint communication, but they provide an easy way to convert multimodal data observed online (RGB-D and language) into updated safety representations. With this, the robot can detect hard-to-encode constraints such as a workout zone, coffee spills, or designated no-go zones (Figure 1). To ensure safety with respect to both pre-defined and new language constraints, we leverage Hamilton-Jacobi reachability analysis [1], [2] to compute a *policy-agnostic* safety controller for the robot which is constantly updated online via efficient warm-starting techniques [3], [4]. The safety controller intervenes only when the robot's nominal planner is at risk of violating either the physical or semantic safety constraints, and provides a corrective safe action. Through simulation studies and

* Equal Contribution. † Equal Advising.
[1]School of Engineering, Federal University of Minas Gerais, Brazil. Email: leohmcs@ufmg.br. Work done during Robotics Institute Summer Scholars (RISS) Program at Carnegie Mellon University. [2]Electrical and Computer Engineering, University of Rochester. Email: zli133@u.rochester.edu. [3]Department of Cognitive Robotics, Delft University of Technology. Email: l.peters@tudelft.nl. [4]Department of Aeronautics and Astronautics, Stanford University. Email: somil@stanford.edu. [5]Robotics Institute, Carnegie Mellon University. Email: abajcsy@cmu.edu

2025 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3315-4139-2/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICRA55743.2025.11127680

experiments on a hardware testbed, we demonstrate the ability of our framework to enable the robot to operate safely, even when new language-based constraints are introduced during deployment.

II. Related Work

Language-Informed Robot Planning. While this topic has been explored for over a decade (see review [5]), advances in internet-scale language and vision-language models have significantly grown language-informed robot planning approaches. Recent works use language for high-level semantic or motion planning [6], [7], [8], [9], providing corrective feedback [10], [11], for low-level control primitives [12], and for language-conditioned end-to-end policies [13], [14]. One common theme in these lines of work is that language provides a flexible mechanism to interact with the robot. Building upon this observation, we use language feedback in our work to enhance robot safety during deployment time.

Safety Constraint Inference from Human Feedback. There is a relatively smaller body of work focused on constraint learning from human feedback. Prior works have inferred state constraints offline from human demonstrations [15], [16], [17], [18], [19], and inferred constraints represented as logical (LTL or STL) specifications from natural language [20], [21]. Our framework focuses on inferring novel state constraints *online* based on multimodal data of image observations and natural language feedback.

Safety Filtering. Safety filters are a popular mechanism to ensure safety for autonomous robots under *any* off-the-shelf planner [22], [23]. The key idea is to use a nominal planner whenever it is safe for the system and intervene with a safety-preserving action whenever the system's safety is at risk. The most popular paradigms to construct safety filters are control barrier functions (CBFs) [24], [25], [26], [27], [28], [29], Hamilton-Jacobi (HJ) reachability analysis [30], [31], [32], [33], and model predictive shielding [34]. We leverage HJ reachability, as it can be easily applied to general nonlinear systems, accounts for control constraints and system dynamics uncertainty, and is associated with a suite of numerical tools [35]. We build on prior work [4], [3], [36] which proposed algorithms for efficiently updating reachability-based safety filters online as the safety constraints change. Our key innovation is incorporating multimodal data of language and images to this online update.

III. Problem Formulation

Robot and Environment. We model the robot as a continuous-time dynamical system $\dot{s}(t) = f(s, a, d)$, where $t \in \mathbb{R}$ is the time, $s \in \mathcal{S}$ is the robot state (e.g., planar position and heading), $a \in \mathcal{A}$ is the robot's control input (e.g., linear and angular velocity). Here, $d \in \mathcal{D}$ is the disturbance which can be an exogenous input (e.g., wind for an aerial vehicle) or represent model uncertainty (e.g., unmodelled tire friction) that we want to be robust to. We assume that the flow field $f : \mathcal{S} \times \mathcal{A} \times \mathcal{D} \to \mathcal{S}$ is uniformly continuous in time, and Lipschitz continuous in s for fixed a and d. The robot is operating in an environment E that it shares with a human, and we assume that the two agents do not expect to physically interact. We use the term "environment" here broadly to refer to factors that are external to the robot (e.g., a building that the robot is navigating in or the surrounding lighting conditions). We also assume that we are given a nominal robot policy $\pi_{\mathcal{R}}(s; E)$ that maps the robot state to control inputs. $\pi_{\mathcal{R}}$ is typically designed to obtain a desired robot behavior, such as reaching a particular goal location for a navigation robot.

Robot Sensor and Perception. The robot has a sensor $\sigma : \mathcal{S} \times E \to \mathcal{O}$ that yeilds (high-dimensional) RGB-D observations. At any time $t \in [0, T]$ during the deployment horizon, let $o^t \in \mathcal{O}$ be the robot's observation.

Human Language Feedback. A human can augment the robot's constraint set at any time during deployment via language commands. More formally, let the human's language command be denoted by $\ell^t \in \mathcal{L}$, where $t \geq 0$ is any time during deployment. In this work, $\mathcal{L}$ are open-vocabulary commands and the set also includes null in which case the person does not describe a new constraint.

Safety Representation: Failure Set. Let $\mathcal{F}_E^* \subset \mathcal{S}$ be the failure set in the human's mind consisting of both physical constraints that are known a priori (e.g., floorplan geometry), as well as the semantically-meaningful constraints that the human describes in the language (e.g., caution tape, spill, etc.). Intuitively, the failure set captures the state constraints that our system must avoid. Traditionally, this failure set is assumed to be specified a priori, and then utilized to compute a safe set and corresponding safety controller automatically. Our work precisely aims to relax this assumption. Thus, $\mathcal{F}_E^*$ can change online as the robot is operating in the environment.

Objective. We seek to design a robot controller $\pi_{\mathcal{R}}^*$ for the robot that respects the safety constraints $\mathcal{F}_E^*$ at all times while following the nominal policy $\pi_{\mathcal{R}}$ as closely as possible.

IV. Background: Hamilton-Jacobi Reachability

Our approach builds upon Hamilton-Jacobi (HJ) reachability analysis [2], [37]. This framework provides robust assurances, yields minimally invasive safety filters compatible with any nominal robot policy (e.g., a neural network), nonlinear systems, and non-convex safety constraints. Here we provide a brief background on the key components of HJ reachability and how to synthesize safety filters with this technique (see these surveys for more details [1], [38]).

Computing the Safety Filter. Given a failure set, $\mathcal{F}$, and the robot dynamics, HJ reachability computes a backward reachable tube (BRT), $\mathcal{S}^\dagger \subset \mathcal{S}$, which characterizes the set of initial states from which the robot is doomed to enter $\mathcal{F}$ despite its best control effort. The computation of the BRT can be formulated as a zero-sum, differential game between the control and disturbance, where the control attempts to

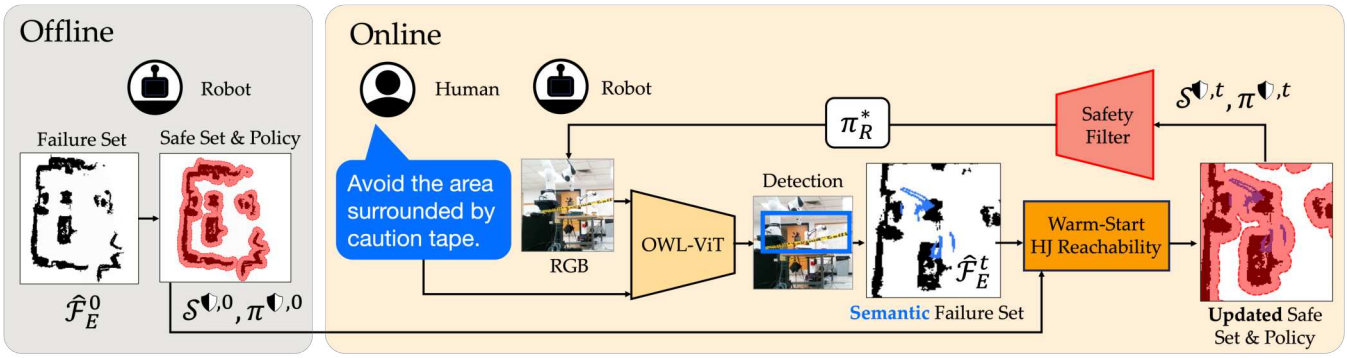

Fig. 2: **Updating Robot Safety Representations Online from Language Feedback.** (left) Offline, the robot has an initial failure set ($\hat{\mathcal{F}}_E^0$) and computes the corresponding safe set ($\mathcal{S}^{\blacklozenge,0}$) and safety policy ($\pi_{\mathcal{R}}^{\blacklozenge,0}$). (right) Online, the person describes their semantic constraint. Using a vision-language model, the robot converts the language-image data into a new failure set. This, along with the previously-computed safe set, are used to efficiently update the safety filter that shields the robot.

avoid the failure region, whereas the disturbance attempts to steer the system inside it. This game can be solved using dynamic programming which, ultimately, amounts to solving the Hamilton Jacobi-Isaacs Variational Inequality (HJI-VI) [39], [37] to compute the value function V that satisfies

$$\begin{aligned} &\min\{D_\tau V(\tau,s) + H(\tau,s,\nabla V(\tau,s)), g(s) - V(\tau,s)\} = 0 \\ &\quad V(0,s) = g(s), \quad \tau \leq 0. \end{aligned} \tag{1}$$

Note that the function $g(s)$ is the implicit surface function representing our failure set $\mathcal{F} = \{s : g(s) \leq 0\}$. Here, $D_\tau V(\tau,s)$ and $\nabla V(\tau,s)$ denote the time and spatial derivatives of the value function $V(\tau,s)$ respectively. The Hamiltonian, $H(\tau,s,\nabla V(\tau,s))$, encodes the role of system dynamics, robot control, and disturbance, and is given by

$$H(\tau,s,\nabla V(\tau,s)) = \max_{a\in\mathcal{A}} \min_{d\in\mathcal{D}} \nabla V(\tau,s)\cdot f(s,a,d). \tag{2}$$

The HJI-VI in (1) can be solved offline via a variety of numerical tools, such as high-fidelity grid-based PDE solvers [35] or neural approximations that leverage self-supervised learning [40] or adversarial reinforcement learning [41]. Once the value function $V(\tau,s)$ is computed, the BRT can be extracted from the value function's sub-zero level set

$$\mathcal{S}^\dagger(\tau) = \{s : V(\tau,s) \leq 0\}, \tag{3}$$

As $\tau \to -\infty$, the BRT represents the infinite time control-invariant set (denoted $\mathcal{S}^\dagger$ here on), which is what we use to construct the safety filter. Importantly, note that $\mathcal{F} \subseteq \mathcal{S}^\dagger$.

Shielding the Robot's Nominal Planner. Along with $\mathcal{S}^\dagger$, HJ reachability yields a corresponding policy-agnostic safety feedback controller $\pi_{\mathcal{R}}^{\blacklozenge}(s)$ that guarantees to keep the robot *outside* the BRT and *inside* the safe states, $\mathcal{S}^{\blacklozenge} = (\mathcal{S}^\dagger)^{\mathsf{c}}$.

$$\pi_{\mathcal{R}}^{\blacklozenge}(s) = \arg\max_{a\in\mathcal{A}} \min_{d\in\mathcal{D}} \nabla V(s)\cdot f(s,a,d), \tag{4}$$

where $V(s)$ represents the value function as $\tau \to -\infty$. Using this, we can design a minimally-invasive control law (i.e., *safety filter*) that *shields* $\pi_{\mathcal{R}}$ from danger:

$$\pi_{\mathcal{R}}^*(s) = \begin{cases} \pi_{\mathcal{R}}(s;E), & \text{if } s \in \mathcal{S}^{\blacklozenge} \\ \pi_{\mathcal{R}}^{\blacklozenge}(s), & \text{otherwise.} \end{cases} \tag{5}$$

V. Updating Robot Safety Representations Online from Natural Language Feedback

While foundational safe control methods like the one in Section IV are powerful, they assume that the robot's safety representation (i.e., the failure set $\mathcal{F}$) is perfectly specified a priori. Our key idea is that vision-language models (VLMs) are not only a useful interface for people to specify unique constraints that they care about, but provide a flexible way to automatically convert multimodal data (vision and language) observed *online* into constraint representations that are compatible with safe control tools. In this section, we detail the core components of our framework (in Figure 2): (1) a VLM-based approach to updating the failure set and (2) an efficient warm-starting approach to update the safety filter online.

Updating the Failure Set Online from Natural Language. We design a constraint predictor

$$\mathcal{P}(o^{0:t}, \ell^{0:t}, \hat{\mathcal{F}}_E^t; E) \to \hat{\mathcal{F}}_E^{t+1}, \tag{6}$$

that updates the inferred failure set based on the sequence of robot observations, human language commands, and last inferred failure set. We assume the initial inferred failure set, $\hat{\mathcal{F}}_E^0$, is given to us; e.g., from mapping the robot's operating environment by running an off-the-shelf SLAM algorithm and extracting an occupancy map. The core of our constraint predictor is a VLM that takes the current image observation (o_t) and the concatenation of all the human's language commands ($\ell_{0:t}$) so far, and produces bounding boxes ($\mathcal{B}_\mathcal{I}$) in the robot's image space associated with the language commands[1]:

$$\phi(o^t, \ell^{0:t}) \to \mathcal{B}_\mathcal{I}. \tag{7}$$

Utilizing the depth information from the RGB-D image o_t, these bounding boxes are projected into the ground plane via:

$$\text{proj}(\mathcal{B}_\mathcal{I}; \lambda) \to \mathcal{B}_{XY} \subset \mathcal{S} \tag{8}$$

[1]Note that the bounding box is an over-approximation. Future work should use semantic segmentation for tighter failure constraint inference.

where proj$(\cdot)$ is the standard camera projection operation depending on the camera intrinsics (λ) that we assume to be known. The predicted failure set, $\hat{\mathcal{F}}_E^{t+1}$, is the prior failure set augmented with $\mathcal{B}_{XY}$. In total, $\mathcal{P}$ is the composition of the VLM ϕ and the operations for converting and augmenting the failure set: $\mathcal{P}(\cdot,\cdot,\mathcal{F}_E^t;E) := \hat{\mathcal{F}}_E^t \cup (\text{proj} \circ \phi)(\cdot,\cdot)$.

Updating the Safety Filter Online via Warm Starting. Every time the predicted failure set changes, $\hat{\mathcal{F}}_E^{t+1}$, we need to also update the corresponding safety controller, $\pi_{\mathcal{R}}^{\blacktriangledown,t+1}$. However, this presents a computational challenge as we need to re-compute online a *new* safety value function $V^{t+1}(s)$ (in Equation 1) so that the robot always has a valid safety-preserving control law. To tackle this, we leverage the approach of warm-starting from [3], [4]. The intuition behind this approach is that since the failure set changes *incrementally* and in a smaller region of the state space, the robot's corresponding safety value function should also only change in a smaller region of the state space. Prior work has precisely demonstrated this property, where warm-starting enables significantly faster updates of the BRT because fewer state values have to be updated [3]. Thus, we leverage the value function computed at the prior timestep (t) to bootstrap the computation of the new value function $(t+1)$ by initializing $V^{t+1}(0,s) = V^t(s)$ (instead of the typical $g(s)$) in Equation 1.

VI. Experimental Setup

Robot Dynamics. In simulation and hardware experiments, we model the robot as a 3D unicycle where the robot controls the linear and angular velocity:

$$\dot{p}_x = v\cos\theta + d_x, \quad \dot{p}_y = v\sin\theta + d_y, \quad \dot{\theta} = \omega, \tag{9}$$

where (p_x, p_y) is the planar position, θ is the heading, and v is the speed of the robot. The robot controls $a := (v,\omega)$ and for reachability analysis, we also model the disturbance to ensure a robust safety filter, $d = (d_x, d_y)$. In simulation, sim we used $0.1\,\mathrm{m\,s^{-1}} \le v \le 1\,\mathrm{m\,s^{-1}}$, $|\omega| \le 1\,\mathrm{rad\,s^{-1}}$ and $|d_i| \le 0.1\,\mathrm{m}, i \in \{x,y\}$. In hardware, we changed the robot linear velocity bounds to $0.0\,\mathrm{m\,s^{-1}} \le v \le 0.5\,\mathrm{m\,s^{-1}}$.

Deployment Scenarios (E). We first deploy our framework the Habitat 3.0 simulator [42] in two different environments from the HSSD-HAB home dataset [43]. The **home gym** scenario features a workout zone consisting of a floor mat and set of barbells in the corner of a living room (top left, Figure 3). The person wants the robot to avoid this area; e.g., because they are working out there or because they want the mat to stay clean. The **hallway** scenario features an expensive rug in the center of the room that the person doesn't want dirtied (bottom left, Figure 3)[2]. The rugs, workout mat, and weights pose a challenge for standard SLAM systems since their geometry alone is not sufficient to distinguish them from free-space. Instead, their subjective value to a person renders them a *semantic* constraint that is communicated verbally.

[2]We modified the original map slightly by removing the center bench.

VLM Model (ϕ). In both simulated and hardware experiments, we use a pre-trained OWLv2 VLM [44] which is an open-vocabulary object detector capable of identifying uncommon objects from natural language descriptions.

Nominal Robot Policy ($\pi_{\mathcal{R}}$). Our approach is agnostic to the nominal robot policy. However, in experiments we use a Model Predictive Path Integral (MPPI) planner [45]. The cost function consists of a goal-reaching term (sum of the Euclidean distance to a goal location, called *cost to goal*) and a collision cost term (where, given the map of obstacles, the robot receives a high penalty for entering the obstacle zone and zero otherwise). Note that the MPPI planner does not model the disturbance in the dynamics; $d = 0$ in Equation 9.

Methods. We compare two ways of inferring the failure set (SLAM-only vs. VLM-informed) and two robot policy designs (with and without a safety filter). We use the RTAB-Map SLAM module [46]. In total, we compare four methods: (1) **Plan-SLAM**: MPPI planner without a safety filter that plans around obstacles detected only by a SLAM module, (2) **Plan-Lang**: MPPI planner without a safety filter that plans around obstacles inferred by our VLM constraint inference predictor, (3) **Safe-SLAM**: MPPI planner shielded by a safety filter that only knows of obstacles detected by SLAM, (4) **Safe-Lang**: our approach that uses a language-informed safety filter.

Deployment Details. We always keep the robot start and goal fixed. When projecting the semantic constraint detections to the ground plane (Equation 8), we only include pixels within a distance threshold τ_{dist} from the robot, ensuring that distant, free-space pixels are not incorrectly treated as part of the obstacle. All modules run on individual threads, and the VLM and BRT run asynchronously on a NVIDIA RTX A6000. To address delays in action execution or the network, we apply the safety filter at a small super-zero level set (i.e. a slight under-approximation of the safe *zero* level set in Equation 5).

VII. Simulation Results

A. On the Accuracy of Failure Set Inference from Language

For the same constraint, users may give varying language descriptions. Thus, we first study the accuracy of our VLM-based failure set inference to varying language inputs.

Independent Variables. We test language commands with varying levels of detail about the failure set $\mathcal{F}_E^*$. In the **home gym** scenario, the language follows a template: ℓ = "`Avoid the` X" where we vary X={`floormat and weights`, `free weights area`, `workout room`, `exercise station`}. In the **hallway** scenario, the language follows a template: ℓ = "`Don't drive over the` X" where we vary X={`carpet`, `expensive rug`, `rug`, `rug in the hallway`}.

Metrics. We compare the ground-truth $\mathcal{F}_E^*$ obtained in the simulator to the final inferred $\hat{\mathcal{F}}_E^T$ at the end of deployment. We measure the ***IoU (Intersection over Union)*** = $\frac{|\hat{\mathcal{F}}_E^T \cap \mathcal{F}_E^*|}{|\hat{\mathcal{F}}_E^T \cup \mathcal{F}_E^*|}$

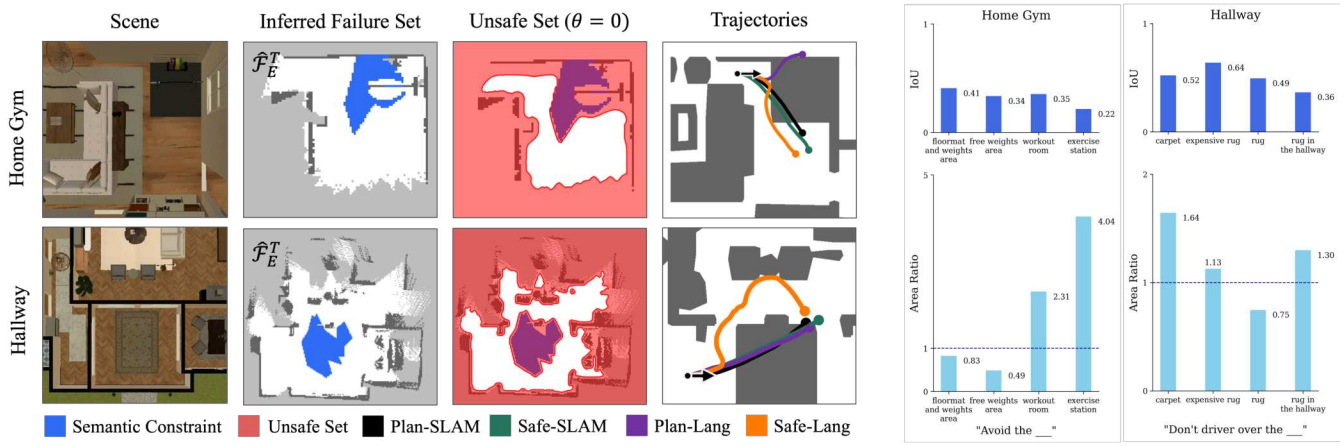

Fig. 3: **Simulation: Closed-Loop Behavior.** (left) Two simulated scenes from HSSD-HAB dataset [43], the final physical and semantic failure set and corresponding unsafe set, and the closed-loop trajectories of all methods. (right) Failure set inference accuracy as function of language command. Metrics compare the ground-truth failure $\mathcal{F}_E^*$ set and the inferred failure $\hat{\mathcal{F}}_E^T$.

which measures the accuracy of the inferred failure set by quantifying the alignment with the ground truth failure set. The closer to $IoU = 1$, the more accurate. We also measure ***Area Ratio*** $= \frac{|\hat{\mathcal{F}}_E^T|}{|\mathcal{F}_E^*|}$ which measures how over-conservative (ratio > 1) or under-conservative (ratio < 1) the inferred failure set is.

Results. Figure 3 (right) shows the IoU and area ratio results for both the home gym and hallway scenarios. In the **home gym**, we find that as the language command becomes more vague, the VLM becomes *over-conservative*, detecting a majority of the room as the constraint rather than just the floormat workout area (0.83 when commanded "floormat and weights" while 4.04 when commanded "exercise station"). However, in the **hallway** scenario, the area ratio is fairly consistently close to 1. Across both methods, however, the IoU scores are relatively low. This is because we only project pixels within the threshold distance, τ_{dist}, from the VLM detections onto $\hat{\mathcal{F}}_E$. Thus, we tend to include less of the very distant parts of the failure set in our inferred set. In practice, however, we found that this does not severely impact the robot's behavior, which largely relies on reliable detection *nearby*.

B. On the Closed-Loop Robot Performance

We next study the closed-loop performance of a robot navigating through our scenarios when using each method: **Plan-SLAM**, **Safe-SLAM**, **Plan-Lang**, and **Safe-Lang**. The language command is always kept the same (**home gym** is "`Avoid the free weights area`" and **hallway** is "`Don't drive over the rug`") and is given at $t = 0$.

Metrics. We measure the speed of generating a robot action via the average ***Plan Time***. For **Plan-SLAM** and **Plan-Lang** it is the plan time required by MPPI, whereas for **Safe-SLAM** and **Safe-Lang** it includes the plan time of MPPI and the safety filtering. Note that the VLM calls and BRT updates are computed asynchronously, so they do not contribute to this metric. We measure the goal reaching efficiency via the average ***Cost to Goal*** over the executed trajectory, where lower means more efficient. We also measure an indicator ***Abides*** $\mathcal{F}_E^*$ if the robot ever violates $\mathcal{F}_E^*$, and report $\pi_{\mathcal{R}}$ ***Active*** for **Safe-SLAM** and **Safe-Lang** to measure the % of time the safety controller intervened during the trajectory.

Results: Quantitative & Qualitative. Table I shows quantitative results in both scenarios. Among the four methods, only ours was able to avoid both physical and semantic constraints. The planning time is not significantly increased by calling the safety filter, but the robot is slightly less efficient at goal reaching. We show qualitative results of closed-loop robot performance in Figure 3. We observe that **Plan-SLAM** and **Safe-SLAM** reach the goal while respecting the physical constraints, but completely violate the semantic constraints, as they can't be detected by SLAM alone. When language is included, **Plan-Lang** fails to adapt and ignores the semantic constraints detected in runtime: it either fails to find a feasible alternative path and ends up colliding (see **home gym** in Figure 3), or slows down until it can't find an alternative path and move towards the goal ignoring the semantic constraint whatsoever (see **hallway** in Figure 3). In contrast, our approach **Safe-Lang** ensures that the robot will execute the optimal control action to avoid *both* the semantic constraints and the physical constraints, so long as the BRT is updated fast enough. We study this more in Section VII-C.

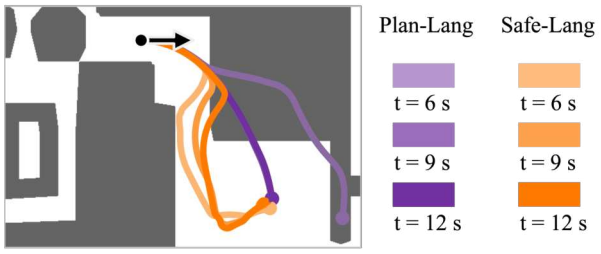

Fig. 4: **Simulation: Language Timing.** Our **Safe-Lang** method is more robust to feedback timing than **Plan-Lang**.

Method	Home Gym Scenario				Hallway Scenario			
	Plan Time (s)	Cost to Goal	Abides $\mathcal{F}_E^*$	$\pi_{\mathcal{R}}^{\blacklozenge}$ Active	Plan Time (s)	Cost to Goal	Abides $\mathcal{F}_E^*$	$\pi_{\mathcal{R}}^{\blacklozenge}$ Active
Plan-SLAM	0.033 ($\pm$ 0.011)	1.83	✗	N/A	0.083 ($\pm$0.053)	3.39	✗	N/A
Safe-SLAM	0.071 ($\pm$0.045)	1.95	✗	0.0%	0.090 ($\pm$1.34)	3.40	✗	12.37%
Plan-Lang	0.115 ($\pm$0.054)	∞ (collision)	✗	N/A	0.056 ($\pm$0.055)	3.19	✗	N/A
Safe-Lang (ours)	0.105 ($\pm$0.057)	2.10	✔	23.83%	0.172 ($\pm$0.099)	4.06	✔	61.67%

TABLE I: **Simulation: Closed-Loop Metrics.** Our approach consistently respects physical and semantic constraints.

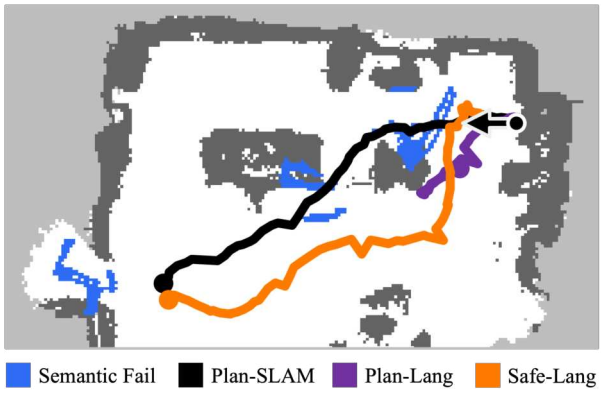

Fig. 5: **Hardware: Closed-Loop Motion.** Without semantic constraints, **Plan-SLAM** cuts through the caution tape zone. **Safe-Lang** respects both the physical and semantic constraints.

C. On the Robustness to Language Feedback Timing

Next, we study the robustness of **Safe-Lang** to language constraints added at some time $t > 0$ during deployment. For brevity, we present results only in **home gym**.

Independent Variables. We use the language command, ℓ^t = "`Avoid the floor mat and weights`", but vary the time when it is specified to the robot: $t = \{6s, 9s, 12s\}$.

Results. Figure 4 shows the closed-loop trajectories of the methods that use language feedback across all language command time points. Similar to prior studies [47], we found that **Plan-Lang** fails to avoid constraints when the language feedback is obtained near the boundary of the constraint, especially when the free passage is narrow as in this study. If the language constraint is added when the robot is closer to the rug ($t = 9$), **Plan-Lang** fails to find a way around it (the robot slows down, but can't turn fast enough to avoid). In contrast, our approach **Safe-Lang** is more robust to language command timing. As long as the language command is given early enough so that the BRT can be updated in time (which took 3 s in this specific study), our method is able to avoid the new semantic constraints. Even when the robot was not able to completely avoid the constraint due to timing (as in $t = 12s$) our framework ensures the robot always has a best-effort action to leave the unsafe set as fast as possible.

VIII. Hardware Experiments

We deployed our framework in hardware on a LoCoBot ground robot equipped with an Intel RealSense camera.

Deployment Scenarios. We study a scenario that a real robot may face but is hard to simulate: avoiding areas marked by **caution tape**. The person specifies their desired constraint via the utterance ℓ^t = "`Avoid the area surrounded by caution tape`" at $t = 0$ of robot deployment. We also qualitatively test a scenario with *both* caution tape and coffee spill language constraints (Figure 1) and another scene with ℓ^t ="`Avoid the dog toys and the laundry`". Videos are on the project website.

Method	Caution Tape Scenario			
	Plan Time (ms)	t-to-Goal	Abides $\mathcal{F}_E^*$	$\pi_{\mathcal{R}}^{\blacklozenge}$ On
Plan-SLAM	7 ($\pm$1)	17.647	✗	N/A
Plan-Lang	40 ($\pm$30)	∞	✗	N/A
Safe-Lang	31 ($\pm$29)	26.176	✔	29.37%

TABLE II: **Hardware: Closed-Loop Metrics.** We see similar trends in hardware as in simulation: informing a safety controller with language enables efficient task completion while respecting both physical and semantic constraints.

Metrics. We use the same metrics as in Section VII-B except we measure Time-to-Goal (in s) and Plan Time (in ms).

Results: Quantitative & Qualitative. We observed that our method was the only one able to avoid both physical and semantic constraints (see Table II and Figure 5). While the base planner **Plan-SLAM** avoids physical constraints and reach the goal, it could not avoid the semantic constraint as it is not detectable by the SLAM system alone. The language informed base planner **Plan-Lang** was in fact able to detect and plan around the semantic constraint, but ended up colliding with the physical obstacles, since it provides no guarantees in terms of safety and is not robust to new constraints added in runtime. Our method **Safe-Lang** was able to react and avoid both physical and semantic constraints and reach the goal safely due to its strong safety guarantees.

IX. Conclusion

We propose a framework that enables robots to continuously update their safety representations online from natural language feedback. By leveraging pre-trained VLM, multimodal sensor observations are converted into constraints compatible with safety-oriented control tools such as Hamilton-Jacobi reachability. Across a suite of simulation and hardware experiments in ground navigation, we show that this is a promising first step toward enabling robots to continually refine their understanding of safety. Looking ahead, calibrating the VLM output (e.g., [48]) is crucial for reliable constraint inference. Studying a reach-avoid variant can also ensure task completion, and exploring semantic segmentation or improved VLM inference may address over-conservativeness in failure-set construction.

REFERENCES

[1] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-jacobi reachability: A brief overview and recent advances," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 2242–2253.

[2] I. Mitchell, A. Bayen, and C. J. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on Automatic Control (TAC)*, vol. 50, no. 7, pp. 947–957, 2005.

[3] A. Bajcsy, S. Bansal, E. Bronstein, V. Tolani, and C. J. Tomlin, "An efficient reachability-based framework for provably safe autonomous navigation in unknown environments," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 1758–1765.

[4] S. L. Herbert, S. Bansal, S. Ghosh, and C. J. Tomlin, "Reachability-based safety guarantees using efficient initializations," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 4810–4816.

[5] S. Tellex, N. Gopalan, H. Kress-Gazit, and C. Matuszek, "Robots that use language," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 25–55, 2020.

[6] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[7] D. Shah, M. R. Equi, B. Osiński, F. Xia, B. Ichter, and S. Levine, "Navigation with large language models: Semantic guesswork as a heuristic for planning," in *Conference on Robot Learning*. PMLR, 2023, pp. 2683–2699.

[8] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 523–11 530.

[9] W. Liu, C. Paxton, T. Hermans, and D. Fox, "Structformer: Learning spatial structure for language-guided semantic rearrangement of novel objects," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6322–6329.

[10] P. Sharma, B. Sundaralingam, V. Blukis, C. Paxton, T. Hermans, A. Torralba, J. Andreas, and D. Fox, "Correcting robot plans with natural language feedback," *arXiv preprint arXiv:2204.05186*, 2022.

[11] Y. Cui, S. Karamcheti, R. Palleti, N. Shivakumar, P. Liang, and D. Sadigh, "No, to the right: Online language corrections for robotic manipulation via shared autonomy," in *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, 2023, pp. 93–101.

[12] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9493–9500.

[13] C. Lynch and P. Sermanet, "Language conditioned imitation learning over unstructured data," *arXiv preprint arXiv:2005.07648*, 2020.

[14] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[15] D. R. Scobee and S. S. Sastry, "Maximum likelihood constraint inference for inverse reinforcement learning," *International Conference on Learning Representations*, 2019.

[16] G. Chou, D. Berenson, and N. Ozay, "Learning constraints from demonstrations," in *Algorithmic Foundations of Robotics XIII: Proceedings of the 13th Workshop on the Algorithmic Foundations of Robotics 13*. Springer, 2020, pp. 228–245.

[17] K. Kim, G. Swamy, Z. Liu, D. Zhao, S. Choudhury, and S. Z. Wu, "Learning shared safety constraints from multi-task demonstrations," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[18] D. Lindner, X. Chen, S. Tschiatschek, K. Hofmann, and A. Krause, "Learning safety constraints from demonstrations with unknown rewards," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2024, pp. 2386–2394.

[19] A. Shah, M. Vazquez-Chanlatte, S. Junges, and S. A. Seshia, "Learning formal specifications from membership and preference queries," *arXiv preprint arXiv:2307.10434*, 2023.

[20] C. Finucane, G. Jing, and H. Kress-Gazit, "Ltlmop: Experimenting with language, temporal logic and robot control," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 1988–1993.

[21] J. Pan, G. Chou, and D. Berenson, "Data-efficient learning of natural language to linear temporal logic translators for robot task specification," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 554–11 561.

[22] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 269–296, 2020.

[23] K.-C. Hsu, H. Hu, and J. F. Fisac, "The safety filter: A unified view of safety-critical control in autonomous systems," *Annual Review of Control, Robotics, and Autonomous Systems*, 2023.

[24] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[25] Z. Qin, K. Zhang, Y. Chen, J. Chen, and C. Fan, "Learning safe multi-agent control with decentralized neural barrier certificates," *International Conference on Learning Representations*, 2021.

[26] Y. Chen, A. Singletary, and A. D. Ames, "Guaranteed obstacle avoidance for multi-robot operations with limited actuation: A control barrier function approach," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 127–132, 2020.

[27] J. Li, Q. Liu, W. Jin, J. Qin, and S. Hirche, "Robust safe learning and control in an unknown environment: An uncertainty-separated control barrier function approach," *IEEE Robotics and Automation Letters*, 2023.

[28] C. Dawson, Z. Qin, S. Gao, and C. Fan, "Safe nonlinear control using robust neural lyapunov-barrier functions," in *Conference on Robot Learning*. PMLR, 2022, pp. 1724–1735.

[29] S. Liu, C. Liu, and J. Dolan, "Safe control under input limits with neural control barrier functions," in *Conference on Robot Learning*. PMLR, 2023, pp. 1970–1980.

[30] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin, "Fastrack: A modular framework for fast and guaranteed safe motion planning," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 1517–1522.

[31] S. Singh, A. Majumdar, J.-J. Slotine, and M. Pavone, "Robust online motion planning via contraction theory and convex optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[32] R. Tian, L. Sun, A. Bajcsy, M. Tomizuka, and A. D. Dragan, "Safety assurances for human-robot interaction via confidence-aware game-theoretic human models," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 11 229–11 235.

[33] D. P. Nguyen, K.-C. Hsu, J. F. Fisac, J. Tan, and W. Yu, "Gameplay filters: Robust zero-shot safety through adversarial imagination," in *8th Annual Conference on Robot Learning*, 2024.

[34] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, pp. 411–444, 2022.

[35] I. Mitchell, "A toolbox of level set methods," *http://www. cs. ubc. ca/mitchell/ToolboxLS/toolboxLS. pdf, Tech. Rep. TR-2004-09*, 2004.

[36] J. Borquez, K. Nakamura, and S. Bansal, "Parameter-conditioned reachable sets for updating safety assurances online," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[37] K. Margellos and J. Lygeros, "Hamilton–jacobi formulation for reach–avoid differential games," *IEEE Transactions on automatic control*, vol. 56, no. 8, pp. 1849–1861, 2011.

[38] K. P. Wabersich, A. J. Taylor, J. J. Choi, K. Sreenath, C. J. Tomlin, A. D. Ames, and M. N. Zeilinger, "Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems," *IEEE Control Systems Magazine*, vol. 43, no. 5, pp. 137–177, 2023.

[39] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Sastry, "Reach-avoid problems with time-varying dynamics, targets and constraints," in *Proceedings of the 18th international conference on hybrid systems: computation and control*, 2015, pp. 11–20.

[40] S. Bansal and C. J. Tomlin, "Deepreach: A deep learning approach to high-dimensional reachability," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 1817–1824.

[41] K.-C. Hsu, D. P. Nguyen, and J. F. Fisac, "Isaacs: Iterative soft adversarial actor-critic for safety," in *Learning for Dynamics and Control Conference*. PMLR, 2023, pp. 90–103.
[42] X. Puig, E. Undersander, A. Szot, M. D. Cote, R. Partsey, J. Yang, R. Desai, A. W. Clegg, M. Hlavac, T. Min, T. Gervet, V. Vondrus, V.-P. Berges, J. Turner, O. Maksymets, Z. Kira, M. Kalakrishnan, J. Malik, D. S. Chaplot, U. Jain, D. Batra, A. Rai, and R. Mottaghi, "Habitat 3.0: A co-habitat for humans, avatars and robots," 2023.
[43] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, *et al.*, "Simple open-vocabulary object detection," in *European Conference on Computer Vision*. Springer, 2022, pp. 728–755.
[44] M. Minderer, A. Gritsenko, and N. Houlsby, "Scaling open-vocabulary object detection," 2024. [Online]. Available: https://arxiv.org/abs/2306.09683
[45] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1433–1440.
[46] M. Labbé and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of field robotics*, vol. 36, no. 2, pp. 416–446, 2019.
[47] E. Trevisan and J. Alonso-Mora, "Biased-mppi: Informing sampling-based model predictive control by fusing ancillary controllers," *IEEE Robotics and Automation Letters*, 2024.
[48] A. Dixit, Z. Mei, M. Booker, M. Storey-Matsutani, A. Z. Ren, and A. Majumdar, "Perceive with confidence: Statistical safety assurances for navigation with learning-based perception," in *8th Annual Conference on Robot Learning*, 2024.