PRESTO: PRIORITIZED REASONING FOR SAFE TRAJECTORY OPTIMIZATION WITH FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Ensuring safety in open-world embodied systems is difficult not only because novel objects and hazards arise at test time, but because existing formal methods treat all safety constraints uniformly–leading to overly conservative or infeasible behavior in complex scenes. This paper introduces *PRESTO*, a framework that leverages multimodal large language models (MLLMs) to generate interpretable safety constraints directly from visual observations, and embeds that into a prioritized safe planner. Our framework enables (i) open-vocabulary safety reasoning over previously unseen objects, (ii) priority-aware safety constraints based on semantic risk, (iii) planner-agnostic deployment as a safety layer, and (iv) interpretable safety constraints. We evaluate *PRESTO* across autonomous driving and robotic manipulation tasks, reducing collision rates by 59.25% when applied as a safety layer to existing planners, while also generalizing to manipulation settings requiring semantic reasoning about fragile, hazardous, and low-risk objects. Our results highlight the potential of combining high-level multimodal reasoning with formal safety guarantees for scalable and interpretable autonomy. More details are available at: https://sites.google.com/view/prestoiclr.

1 INTRODUCTION

Modern embodied systems increasingly rely on foundation models for high-level reasoning, planning, and interaction with the physical world. Recent advances in vision-language and language-conditioned policies have enabled robots to perform increasingly complex manipulation and navigation tasks from natural language instructions (Ahn et al., 2022; Kawaharazuka et al., 2025; Liang et al., 2023). By leveraging large-scale pretrained models, these systems exhibit strong generalization to unseen tasks, objects, and environments.

Despite these advances, ensuring the safety of embodied foundation-model-based systems remains a major challenge. Most existing approaches optimize average task performance without explicit safety guarantees, while methods that incorporate formal safety constraints often rely on manually specified rules, fixed object taxonomies, or task-specific safety formulations. Such approaches are difficult to scale to open-world environments where robots must reason about previously unseen objects and context-dependent hazards. Moreover, enforcing all constraints uniformly can lead to overconservative behavior or infeasible planning problems in cluttered scenes with conflicting safety requirements.

Humans, however, do not reason about safety through rigid geometric constraints alone. Instead, they prioritize risks semantically and contextually: colliding with a balloon is treated differently from colliding with a pedestrian, and navigating near a fragile vase differs from navigating near a cardboard box. Recent multimodal large language models (MLLMs) have demonstrated strong capabilities in semantic reasoning (Tang et al., 2023), commonsense understanding (Zhao et al., 2023), 3D grounding (Team, 2025), and open-vocabulary scene interpretation (Brown et al., 2020; Radford et al., 2021). These capabilities suggest that MLLMs may provide a mechanism for adaptive safety reasoning in complex embodied environments.

In this work, we propose *PRESTO* that combines semantic reasoning from MLLMs with control-theoretic safety constraints through Control Barrier Functions (CBFs). Given visual observations of a scene, an MLLM first identifies objects, reasons about their semantic safety criticality, and generates a prioritized set of safety constraints. These priorities are then incorporated into a modified CBF-based

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



Figure 1: Given RGB observations from the robot, our method leverages a multimodal large language model (MLLM) to reason about the semantic safety criticality of objects and generate prioritized safety constraints. Collision-critical objects (orange) are assigned stricter safety constraints, while lower-risk objects (blue) admit greater relaxation to preserve feasibility in cluttered environments. The resulting prioritized constraints are integrated into a Control Barrier Function (CBF)-based optimization framework to generate safe trajectories across diverse embodied tasks.

optimization framework that adaptively relaxes lower-priority constraints while preserving strict enforcement for safety-critical obstacles. By integrating high-level semantic reasoning with formal control-theoretic guarantees, our approach enables robots to reason about safety in a context-aware and interpretable manner while remaining robust to novel environments. We evaluate our framework across autonomous driving and robotic manipulation tasks. In driving scenarios, our method reduces collision rates by 59.25% when used as a safety layer for existing planners while remaining planner-agnostic. We further demonstrate that the framework generalizes across manipulation settings requiring semantic reasoning about fragile, hazardous, or low-risk objects.

Our contributions are as follows:

- We propose a novel framework that bridges semantic reasoning from multimodal large language models with control-theoretic safety constraints through Control Barrier Functions;
- An interpretable priority-based constraint generation system that handles conflicting safety requirements;
- We demonstrate that our method improves safety across autonomous driving benchmarks while remaining compatible with existing planners.
- Comprehensive evaluation showing 59.25% reduction in collision rates across autonomous driving scenarios while,
- Demonstration of generalizatiblity across robotic manipulation tasks with semantic understanding of object safety criticality.

2 Related Works

Our approach is at the intersection of several areas of intense research,

Control Barrier Functions for Safety-Critical Learning and Control. Control barrier functions (CBFs) provide a rigorous, Lyapunov-inspired framework for guaranteeing safety in continuous dynamical systems by enforcing forward invariance of a safe set (Ames et al., 2016; 2019). Their applicability to real-world robotics has been demonstrated by approaches that embed CBF constraints as a learnable layer within neural robot controllers (Xiao et al., 2023) (Wang et al., 2023) (Wong et al., 2026) While recent works have explored combining vision with CBFs (Xiao et al., 2023; Wang et al., 2023), these approaches rely on fixed object detection taxonomies and manually designed safety rules. In contrast, our framework leverages the open-vocabulary reasoning capabilities of MLLMs to automatically generate context-aware safety constraints that adapt to novel objects and scenarios not seen during training. This semantic flexibility represents a fundamental advance beyond prior CBF and vision approaches.

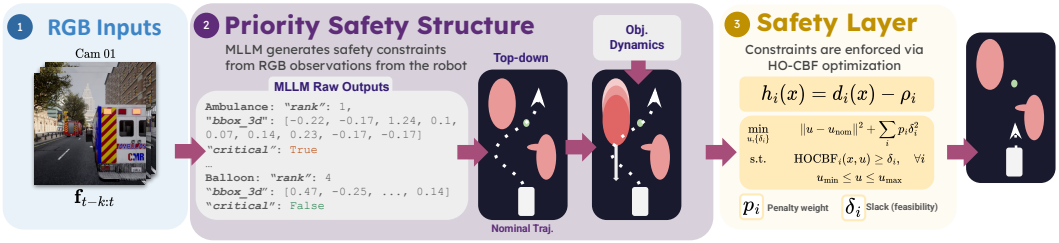

Figure 2: **Framework overview** (*Left*) RGB camera input from robot; (*Middle*) MLLM generates prioritized safety constraints with semantic labels and 3D bounding boxes; (*Right*) CBF safety layer produces safe control commands. Orange objects indicate collision-critical obstacles requiring strict avoidance, while blue objects are non-collision-critical (e.g., balloon).

In the driving domain, CBFs have been applied both to vision-based end-to-end control (Xiao et al., 2022) and to shared-control settings (Dallas et al., 2025). A complementary direction combines CBFs with reactive task specifications such as behavior trees, enabling compositional behavior with safety guarantees (Özkahraman & Ögren, 2020; Xiao & Belta, 2022).

Safety constraint prioritization. Safety constraint prioritization has been studied through predefined rule hierarchies for autonomous driving, including Signal Temporal Logic specifications that can be enforced through corresponding barrier-function formulations (Veer et al., 2023). Related learned safety-specification approaches, such as AnySafe (Agrawal et al., 2025) and Safety on the Fly (Knoedler et al., 2025), adapt or construct safety specifications from observations at runtime. More recently, LLMs and VLMs have been used to define or adapt safety constraints online, including augmenting vision-language-action models with safety layers (Hu et al., 2025; Pandya et al., 2025), grounding contextual safety reasoning into the control stack (Ravichandran et al., 2026), extracting socially-aware spatial constraints (Ruo et al., 2025), performing safety-aware task planning (Khan et al., 2025), and regulating CBF parameters from vision-language observations (Chen & Chandra, 2026). Our approach differs from these works by using an MLLM to infer a context-dependent semantic ordering over multiple safety constraints. Rather than relying on a fixed rule hierarchy or adapting constraints independently, this enables the system to resolve competing constraints online when they cannot all be satisfied simultaneously.

Safe Reinforcement Learning. A related field is safe reinforcement learning (RL), with early approaches characterized the problem of safe policy learning, e.g. with constrained MDPs or risk-sensitive objectives (Garcıa & Fernández, 2015), or reasoning about stability/learned values functions (Chow et al., 2018). Recent approaches broadly advance constrained policy optimization, shielding methods, and formal verification for learned policies (Gu et al., 2024), or update optimization approaches so as to embed hard safety constraints directly into the policy update (Milosevic et al., 2024). In contrast, our approach enforces safety explicitly via CBF constraints whose parameters are reasoned about semantically, without requiring unsafe interactions during training.

Large Language Models and Vision-Language-Action Models for Embodied Control. Finally language models have seen extensive use for embodied agents planning, either by grounding LLM-generation of instructions (Ahn et al., 2022; Liang et al., 2023), direct language-conditioned visuomotor policies (Zitkovich et al., 2023), or vision-language-action models for both general robotics (Kawaharazuka et al., 2025) and autonomous driving applications (Jiang et al., 2025). Our approach addresses the complementary question of how the semantic knowledge embedded in MLLMs can be harnessed to enforce safety with common sense.

3 Method

Problem Statement. Let the robot be governed by control-affine, known, dynamics $\dot{x} = f(x) + g(x)u$, where x is the robot state, u is the control input, and f, g are locally Lipschitz. Given a stream of RGB observations $\mathbf{o}_{t-k:t}$ over a time horizon t, our goal is to synthesize a control policy such that the robot avoids all the constraints informed by the observation images' semantics while completing the assigned task.

3.1 Learning for Safety Constraints

In many robotic applications, explicit safety constraints are not directly observable and must instead be inferred from high-dimensional sensory inputs such as RGB images. Existing control barrier function (CBF) methods typically assume access to manually specified safety constraints for obstacles in the environment. To address this limitation, we propose ***PRESTO***, a framework that leverages multimodal large language models (MLLMs) to infer structured safety constraints directly from visual observations. As illustrated in Figure 2, our framework consists of two components. First, an MLLM reasons over scene observations to construct a prioritized constraints list that captures object grounding, semantic importance, and collision criticality (Section 3.2). Second, these representations are translated into geometric safety regions, together with estimates of future object states, and enforced through a High-Order Control Barrier Function (HOCBF) optimization layer that generates a safe trajectory satisfying the inferred safety constraints (Section 3.3).

3.2 Reasoning for Priority Safety Structure

When multiple safety constraints are present $\{h_i(x) \geq 0, i \in S_o\}$, enforcing all of them equally may prevent the controller from finding a feasible policy for completing the task. Moreover, different obstacles may carry different levels of safety significance. For example, in autonomous driving, violating a constraint associated with a pedestrian should be penalized more heavily than violating one associated with a less critical object, such as a balloon. We therefore introduce a reasoning module that assigns a priority structure to the perceived obstacles before they are passed to the downstream control barrier function layer.

Given a sequence of RGB observations $\mathbf{o}_{t-k}$, the multimodal policy `mllm` produces a *prioritized constraints list* $\mathcal{C} = \{(r_i, c_i, \mathbf{b}_i)\}_{i=1}^{N}$. where $r_i \in 1, \ldots, N$ denotes its predicted priority rank, $c_i \in 0, 1$ is its collision-criticality flag, and $\mathbf{b}_i \in \mathbb{R}^6$ denotes the spatial grounding of object i, such as a 3D bounding box. A smaller rank indicates higher priority, with $r_i = 1$ corresponding to the highest-priority object. When objects are dynamic, their spatial grounding can be propagated overtime using a standard 3D object tracker, such as (Weng et al., 2020). The predicted ranks are converted into normalized priority weights w_i through rank normalization. For each collision-critical objects ($c_i = 1$), the spatial grounding $\mathbf{b}_i$ is converted into a geometric safety constraint $h_i(x) \geq 0$. The semantic priority weight w_i determines the relative slack penalty assigned to constraint i in the downstream HOCBF optimization, such that higher-priority constraints incur a larger cost when relaxed. We use a Qwen3-VL-4B-Instruct (Team, 2025) backbone, which was already fine-tuned for 3D visual grounding. Inference is deterministic (greedy decoding, fixed random seed) with `max_new_tokens`=256. Weights are loaded in 4-bit precision using NF4 quantization. Further implementation details and prompt templates are provided in the Appendix.

To ensure geometric consistency with the controlled layer, the spatial groundings $\mathbf{b}_i$ are aggregated across all available camera views and projected into a bird's eye-view and parameterized as ellipitical safety regions regions $\{o_x, o_y, a, b, \psi\}$, where (o_x, o_y) is the object center, a and b are the semi-major and semi-minor axes, and ψ is the orientation angle. This explicit geometric representation yields interpretable safety constraints that are passed to the HOCBF layer.

3.3 Safety Layer

We now describe how the high-level constraints are translated into control-theoretic guarantees. Given a prioritized constraints list $\mathcal{C}$ and a nominal reference trajectory $\tau_{\text{ref}} \sim \pi$ generated by a planner π, the safety layer modifies the corresponding nominal control u_{nom} only as needed to satisfy the inferred safety constraints, which are enforced using a High-Order Control Barrier Functions (HOCBFs) formulation (Xiao & Belta, 2022). For each collision-critical object $v_i \in \mathcal{V}$ with $\delta = 1$, we define a barrier function $h_i(x) = d_i(x) - \rho_i$, where $d_i(x)$ is the distance from the ego agent to the obstacle and ρ_i is the effective safety radius derived from the elliptical safety region $\mathbf{b}_i$, inflated to also account the ego geometry. Suppose the system dynamics are given in the form: $\dot{x} = f(x) + g(x)u$, where u denotes the control input of the system and f, g are locally Lipschitz. For barrier functions of relative degree m, the HOCBF condition is constructed as:

$$L_f \psi_{i,m-1}(x) + L_g \psi_{i,m-1}(x) u + \alpha_m(\psi_{i,m-1}(x)) \geq 0, \tag{1}$$

where $L_f\psi_{i,m-1}(x) = \frac{d\psi_{i,m-1}(x)}{dx} f(x), L_g\psi_{i,m-1}(x) = \frac{d\psi_{i,m-1}(x)}{dx} g(x)$. α_m is a class $\mathcal{K}$ function (strictly increasing function that passes through the origin), and $\psi_{i,m-1}(x)$ is defined as

$$\psi_{i,j}(x) := \dot{\psi}_{i,j-1}(x) + \alpha_j(\psi_{i,j-1}(x)), j \in \{1, \dots, m-1\}, \tag{2}$$

and $\psi_{i,0}(x) := h_i(x)$. $\alpha_j, j \in \{1, \dots, m-1\}$ are also class $\mathcal{K}$ functions. It can be shown that the satisfaction of (1) under a control policy u implies the satisfaction of the original safety constraint $h_i(x) \geq 0$ using the HOCBF theory (Xiao & Belta, 2022).

Safety and tracking objectives are unified into a single QP:

$$\begin{aligned} &\arg\min_{u,\delta_i} ||u - u_{nom}||^2 + \sum_{i \in S} p_i \delta_i^2, \\ &\text{s.t.} \\ &L_f\psi_{i,m-1}(x) + L_g\psi_{i,m-1}(x)u + \alpha_m(\psi_{i,m-1}(x)) \geq \delta_i, \\ &u_{min} \leq u \leq u_{max}, \end{aligned} \tag{3}$$

where u_{nom} is a nominal control input obtained by tracking the reference trajectory τ_{ref} and u_{min}, u_{max} denote the minimum and maximum control bounds. S denotes the set of obstacles that are safety-critical, and δ_i is a slack variable that ensures the feasibility of the above QP. The coefficient p_i denotes the slack penalty associated with constraint i. We parameterize $p_i = \lambda w_i$, where w_i is the semantic priority weight obtained from the MLLM-predicted rank and $\lambda > 0$ controls the overal relaxation penalty. Consequently, the semantic ordering is preserved in the QP: higher-priority constraints receive larger slack penalties and therefore relaxed only after lower-priority constraints when simultaneous satisfaction is infeasible. When $p_i \to \infty$, the corresponding HOCBF constraint becomes a hard constraint. Manually tuning the parameters p_i is non-trivial. We could employ the BarrierNet model (Xiao et al., 2023) to optimally learn the parameters for $p_i, i \in S$. We use the OSQP solver (Stellato et al., 2020).

4 Experiments

In this section, we first present case studies in Section 4.2 to analyze the capabilities of the our method. We then evaluate our method as a safety layer for existing planners in Section 4.3, demonstrating reduced collision rates across multiple planning architectures. Next, in Section 4.4 we validate our method through real-world hardware experiments on a robotic manipulator. Finally, Section 4.5 provides ablation studies on key components of the framework.

4.1 Experimental setup

Driving. For autonomous driving, we use the CARLA simulator (Dosovitskiy et al., 2017). A reference trajectory is provided by a planner (Hu et al., 2023; Jiang et al., 2023) or generated via A* search. Low-level control is implemented using a PID controller that tracks the trajectory with a heuristic target speed profile. We query the `mllm` at $1.\overline{3}$ Hz in closed-loop evaluation, i.e., every 0.75s. For the case study, we use a single front-facing camera; for Bench2Drive, we additionaly incorporate a left camera to improve lateral coverage. More implementation details in Appendix A.1.

Manipulation. For manipulation, we use use IsaacSim (NVIDIA) in the Lightwheel Kitchen environment (Lightwheel, 2025) with a Franka Emika arm. A reference trajectory is generated as a sequence of BEV waypoints. Low-level control is implemented using an RMPFlow controller (Cheng et al., 2019a). We use a single fixed overhead camera.

4.2 Case Study Analysis

We conduct case studies in both autonomous driving and robotic manipulation, shown in Figure 3 and Figure 4 with results summarized in Table 1 and Table 2. In Table 1, our approach achieves full route completion. In total, we evaluate 12 driving scenarios and 10 manipulation scenarios spanning cluttered environments, out-of-distribution (OOD) objects, and semantically grounded safety constraints. For the TRPO, SafeRL, and HOCBF baselines, we provide oracle ground-truth object bounding boxes to isolate the planning and control performance from perception errors. Across

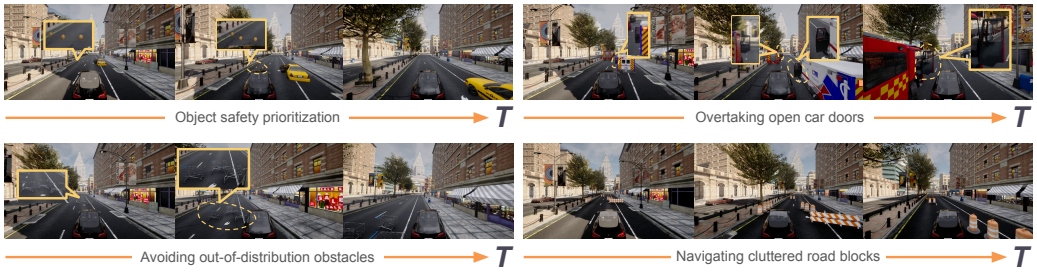

Figure 3: ***PRESTO* for driving.** In autonomous driving, our framework demonstrates semantic safety reasoning across diverse scenarios, including object safety prioritization, overtaking vehicles with open car doors, avoiding out-of-distribution obstacles, and navigating cluttered road blocks. The MLLM reasons about the semantic safety criticality of scene elements to generate prioritized safety constraints, enabling context-aware trajectory generation in complex environments.

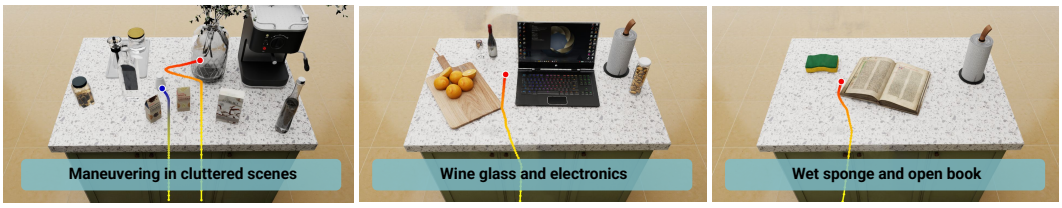

Figure 4: ***PRESTO* for manipulation.** In robotic manipulation, our framework is demonstrated on tabletop tasks requiring semantic understanding of object safety and interaction constraints. Examples include maneuvering in cluttered scenes, avoiding fragile or valuable objects such as wine glasses and electronics, and reasoning about held-object constraints such as avoiding an open book while carrying a wet sponge. The resulting trajectories balance task completion and safety through prioritized constraint reasoning.

both domains, our framework achieves the lowest collision rate while maintaining full task completion. In driving, our method achieves 5.00% collision rate with 98.69% route completion, outperforming prior Safe RL and CBF-based baselines that either remain overly conservative or fail in scenarios that require semantic reasoning, such as the balloon scenario. Similarly, in manipulation, our framework achieves perfect task completion without collisions.

Visual Grounding and Semantic Reasoning. We demonstrate that our method enables reasoning beyond purely geometric obstacle avoidance. In driving, the model distinguishes between semantically low-risk and collision-critical objects. For example, in a scenario containing both a balloon and rigid obstacles, the MLLM outputs "a collision with the balloon would not cause any physical damage to the vehicle or pose a safety risk to occupants," while prioritizing avoidance of the vehicle ahead. Existing CBF-based approaches typically treat all detected obstacles uniformly within the downstream optimization, which can produce overly conservative behavior or infeasible constraints in semantically ambiguous scenes, as observed in the balloon scenario in Table 1.

In manipulation, the generated constraints similarly depend on task context and object semantics. For example, when the robot executes a pick-and-place task while holding a wet sponge or glass of wine, the MLLM infers that traversing above electronic devices such as a laptop introduces spill risk despite remaining geometrically feasible, as shown in Figure 4. These examples highlight that the proposed framework reasons not only about geometry, but also about physical consequences and interaction semantics within the environment.

The use of MLLMs additionally enables open-vocabulary grounding for out-of-distribution (OOD) objects, following the definition in (Aasi et al., 2025). We observe that current state-of-the-art driving policies such as TransFuser++ exhibit elevated collision rates in scenarios containing uncommon obstacles, as shown in Table 1. In particular, the policy failed to maneuver around a shopping cart, likely because such objects are underrepresented in the training distribution, leading to a catastrophic collision. In contrast, the MLLM provides open-vocabulary scene understanding that allows previously unseen objects to be incorporated into the downstream safety constraint generation pipeline, enabling safer behavior in environments containing obstacles outside the training distribution.

Table 1: **Navigation (driving) case studies.** Comparison against prior safe reinforcement learning and CBF-based approaches on representative driving scenarios. Our method achieves the lowest collision rate while maintaining full route completion. Unlike prior methods that enforce uniformly strict geometric constraints, our framework leverages semantic prioritization of safety constraints, enabling safer and less conservative behavior in cluttered environments.

Methods	**Collision Rate (%)** ↓	**Route Completion (%)** ↑	**Conserv.** (≥ 0 **&** ↓)	
			b_c	Avg. min. dist.
Transfuser++ (Jaeger et al., 2023)	53.85	64.54	–	0.35
TRPO (Schulman et al., 2017)	100.00	48.72	–	0.33
SafeRL (TRPO) (Cheng et al., 2019b)	83.33	53.06	–	2.22
HO-CBF (Xiao & Belta, 2022)	8.33	68.67	2.06	2.71
Ours	**5.00** $\pm$ **7.45**	**98.69** $\pm$ **2.96**	0.39 $\pm$0.08	4.58 $\pm$ 0.87

Table 2: **Manipulation case studies.** Comparison on manipulation tasks requiring safe motion in cluttered tabletop environments. Our method achieves perfect task completion while maintaining zero collisions. The results highlight the ability of the proposed framework to reason about semantic object safety, including fragile, valuable, and interaction-sensitive objects, while integrating these priorities within a planning framework.

Methods	**Collision Rate (%)** ↓	**Route Completion (%)** ↑
TRPO (Schulman et al., 2017)	80.00	92.98
SafeRL (TRPO) (Cheng et al., 2019b)	0.00	10.00
Ours	**0.00**	**100.00**

Object Priority and Constraint Feasibility. We further show that object prioritization improves feasibility in cluttered environments. Without a priority structure between visual grounding and the downstream CBF layer, all perceived obstacles are treated as equally collision-critical, which can produce overly conservative constraints and infeasible optimization problems in dense scenes. In the balloon scenario, the baseline HOCBF formulation becomes infeasible due to enforcing strict avoidance of all detected objects, including semantically low-risk obstacles.

In contrast, our framework introduces semantically informed slack modulation, allowing lower-risk objects to admit greater relaxation while preserving strict enforcement for safety-critical constraints. For example, in crowded driving scenes, a pedestrian approaching a vehicle door is assigned higher priority than a parked motorcycle in an adjacent lane. Similar behavior is observed in cluttered manipulation environments. As shown in Figure 4, the MLLM identifies the small biscuit package as "small and lightweight; no significant safety risk," allowing the planner to generate trajectories that pass nearby without unnecessarily constraining the robot motion. In contrast, when operating near fragile objects such as a vase, the generated safety constraints enforce a larger safety margin, causing the trajectory to route around the object. These examples demonstrate that incorporating semantic object understanding into the safety optimization improves both task feasibility and safety-critical behavior. We additionally observe that densely cluttered scenes require longer context lengths and increased inference time due to the larger number of objects and interactions that must be jointly reasoned over.

4.3 Safety Layer for Autonomy

Table 3 shows that applying our framework as a safety layer significantly reduces collision rates across multiple planners on the Bench2Drive benchmark. When integrated with UniAD (Hu et al., 2023), our method reduces the collision rate by 59.25% over 220 driving scenarios while maintaining strong route completion performance. The largest improvements occur in forward-hazard scenarios requiring overtaking, particularly in scenarios involving accidents, construction obstacles, and interurban actor flow. Qualitatively, the proposed framework provides an additional safety mechanism during execution by enforcing control barrier function constraints on top of the nominal planner trajectory. In one scenarios, for example, the planner may initiate an overtaking maneuver even when an approaching vehicle is in the adjacent lane. The safety layer detects the impending constraint violation and modifies the control input to prevent collision. A detailed breakdown of performance across scenario categories is provided in Appendix A.3.4.

Table 3: **Driving Safety Benchmarks.** Evaluation on the Dev10 and Bench2Drive autonomous driving benchmarks (Jia et al., 2024). When applied as a safety layer to existing planners, our framework consistently reduces collision rates. On Dev10, our method reduces collision rates by up to 75.00% across VAD and UniAD planners, while on Bench2Drive it achieves a 59.25% reduction in collisions over UniAD. These results demonstrate that *PRESTO* is planner-agnostic and can improve safety across different planners.

Benchmark	Methods	Collision Rate (%) ↓	Driving Score ↑	Route Completion (%) ↑
Dev10 (Jia et al., 2024)	VAD (Jiang et al., 2023)	40.00	52.37	68.24
	UniAD-Base (Hu et al., 2023)	40.00	53.42	66.34
	UniAD-Tiny	40.00	58.90	76.22
	VAD + Ours	10.00 (↓ 75.00%)	52.59(↑ 0.42%)	61.47 (↓ 9.92%)
	UniAD-Base + Ours	10.00 (↓ 75.00%)	57.23 (↑ 7.13%)	73.50 (↑ 10.79%)
	UniAD-Tiny + Ours	10.00 (↓ 75.00%)	60.60 (↑ 2.89%)	77.60 (↑ 1.81)
Bench2Drive (Jia et al., 2024)	UniAD	49.08	46.23	68.17
	UniAD+Ours	**20.00** (↓ 59.25%)	**58.23** (↑ 25.96%)	**69.46** (↑ 1.89%)

Table 4: **Effect of Semantic Prioritization.** We isolate semantic prioritization by fixing the bounding-box source and CBF formulation within each comparison. Semantic priority improves route completion for both MLLM-predicted and oracle boxes while preserving collision-free performance with oracle geometry, indicating less conservative resolution of conflicting safety constraints.

Method	Collision Rate (%) ↓	Route Completion (%) ↑	Conserv. (≥ 0 & ↓)	
			b_c	Avg. min. dist.
Oracle Boxes w/o Semantic Priority	0.00 ± 0.00	82.16 ± 4.23	0.92 ± 0.47	5.37 ± 1.66
Oracle Boxes w/ Semantic Priority	0.00 ± 0.00	100.00 ± 0.00	0.34 ± 0.02	3.19 ± 0.06
MLLM Perception w/o Semantic Priority	8.33 ± 8.33	84.70 ± 2.03	0.74 ± 0.02	3.23 ± 0.64
MLLM Perception w/ Semantic Priority	5.00 ± 7.45	98.68 ± 2.96	0.39 ± 0.08	4.58 ± 0.87

For comparison, we additionally apply our safety layer to VAD (Jiang et al., 2023) and evaluate using the Dev10 subset following prior work (Jia et al., 2025; Gao et al., 2026). Across planners, our method consistently reduces collision rates, demonstrating that the proposed safety layer is model-agnostic and can be integrated with diverse planning architectures. The lower route completion observed with VAD is primarily attributable to its conservative driving behavior, as the planner often fails to resume motion after coming to a stop. We additionally observe several limitations of the current perception stack. In particular, noisy yaw estimation from the MLLM-generated obstacle representations can produce inaccurate safety constraints in dense traffic scenes.

4.4 Real-world Experiments

We conducted nine real-world hardware experiments across different tasks and objects. Figure 5 shows one hardware experiment using a UR5 robot equipped with a RealHand L6 end effector. The robot is tasked with carrying a sponge across a table while avoiding a laptop positioned along the nominal path. We first execute an unsafe nominal trajectory, shown in Figure 5(left), in which the sponge passes over the laptop. We then apply *PRESTO* to modify the nominal trajectory according to the inferred safety constraints. The resulting safe trajectory is executed on the physical robot, successfully carrying the sponge around the laptop rather than over it Figure 5(right). In a second example, the robot navigates between a cup containing water and an empty mug. *PRESTO* assigns higher priority to the cup containing water due to the greater consequence of contact and potential spillage. This demonstrates the ability of the framework to infer context-dependent safety priorities. Additional hardware experiments across different tasks and objects are provided in Appendix A.2.

4.5 Ablation

Effect of Semantic Prioritization. In Table 4 we ablate the proposed semantic priority mechanism. With MLLM-predicted boxes, semantic priority improves route completion from 84.70% to 98.68%, while with oracle boxes it improves route completion from 82.16% to 100%, with both oracle variants remaining collision-free. The oracle-box result removes MLLM perception error and therefore reflects the benefit of semantic prioritization under accurate geometric grounding. Together, these results show that semantic priority primarily improves progress by relaxing less safety-critical constraints when all constraints cannot be simultaneously satisfied.

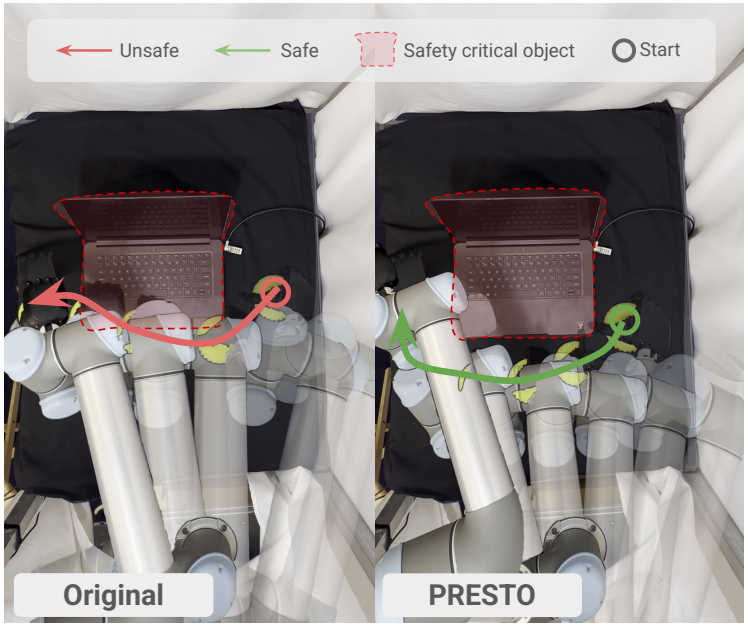

Figure 5: **Hardware experiment on UR5 robot.** The robot transports a wet sponge across the table. The original unsafe trajectory passes over a safety-critical laptop, whereas *PRESTO* modifies the nominal trajectory to avoid the laptop while preserving task completion.

Runtime Breakdown. Table 5 provides a breakdown of the computational cost of our framework. The MLLM inference stage requires 564.15 ± 68.12 ms. Its runtime varies with scene complexity and generally increases with the number of objects in the scene, which affects the number of generated tokens. The HOCBF optimization requires 155.70 ± 1.90 ms. Overall, the framework achieves a total runtime of 914.61 ± 66.94 ms, corresponding to a rate of 1.13 Hz. These results are obtained using LFM2.5-VL model (Amini et al., 2025) implemented with vLLM (Kwon et al., 2023); additional runtime analysis is provided in Appendix A.3.2. All experiments were conducted on an NVIDIA GeForce RTX 4090 GPU. Further efficiency improvements, such as quantization and model distillation, could reduce inference latency without sacrificing significant performance.

Table 5: **Runtime Breakdown.**

Component	Time (ms)	Rate (Hz)
MLLM Inference	564.15 ± 68.12	1.79 ± 0.21
CBF Optimization	297.63 ± 1.75	3.36 ± 0.02
Total	914.61 ± 66.94	1.13 ± 0.07

5 Discussion

Assumptions and Limitations. The efficacy of *PRESTO* is conditional on correct object classification and sufficiently accurate estimates of object geometry and obstacle dynamics. These assumptions are common in CBF-based safety filtering, while accounting for perception and state-estimation uncertainty remains an active area of research (Yun & Azizan, 2026; Abdi et al., 2023). Addressing these uncertainties is beyond the scope of this work; instead, we empirically characterize their impact in the Appendix through localization perturbations, narrow-gap feasibility tests, and comparisons of MLLM classification and prioritization against human annotations in Appendix A.3.1.

6 Conclusion

This work demonstrates that foundation models can provide semantic understanding for formal safety guarantees, opening new directions for safe AI systems. The combination of MLLM reasoning with CBF constraints provides both interpretability (humans can understand why certain safety decisions were made) and formal guarantees essential for safety-critical deployment.

AI USE STATEMENT

In this work, we used generative AI tools for language editing and refinement, as well as for assisting with code development and debugging. We have not used generative AI tools to generate or modify experimental results, and other required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to assist with brainstorming, technical writing, and code implementation. All AI-assisted text was reviewed and revised by the authors, and AI-assisted code was inspected, tested, and validated through the reported experiments. All experimental results, analyses, figures, and conclusions were independently verified by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Human annotators were recruited solely to evaluate model outputs, including object classification accuracy and collision-criticality labels. Annotators were informed of the annotation task and provided judgments based on predefined evaluation criteria. No sensitive personal information was collected, and annotator responses were used only in aggregate for quantitative evaluation. The study did not involve interaction with autonomous systems or expose participants to physical risk.

REPRODUCIBILITY STATEMENT

We will release the code, prompts, and additional implementation details upon publication to facilitate reproducibility. We provide implementation and experimental details throughout the main paper and appendix, including model configurations, prompts, system parameters, evaluation protocols, and hardware specifications. We report the models and software frameworks used, inference settings, benchmark subsets, and evaluation metrics required to reproduce our experiments. Where applicable, experiments are repeated across multiple trials or seeds and results are reported with summary statistics.

REFERENCES

Erfan Aasi, Phat Nguyen, Shiva Sreeram, Guy Rosman, Sertac Karaman, and Daniela Rus. Generating out-of-distribution scenarios using language models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10616–10623, 2025. doi: 10.1109/ICRA55743.2025.11127950.

Hossein Abdi, Golnaz Raja, and Reza Ghabcheloo. Safe control using vision-based control barrier function (v-cbf). In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 782–788, 2023. doi: 10.1109/ICRA48891.2023.10160805.

Sankalp Agrawal, Junwon Seo, Kensuke Nakamura, Ran Tian, and Andrea Bajcsy. Anysafe: Adapting latent safety filters at runtime via safety constraint parameterization in the latent space, 2025. URL https://arxiv.org/abs/2509.19555.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8): 3861–3876, 2016.

Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*, pp. 3420–3431. Ieee, 2019.

Alexander Amini, Anna Banaszak, Harold Benoit, Arthur Böök, Tarek Dakhran, Song Duong, Alfred Eng, Fernando Fernandes, Marc Härkönen, Anne Harrington, Ramin Hasani, Saniya Karwa, Yuri Khrustalev, Maxime Labonne, Mathias Lechner, Valentine Lechner, Simon Lee, Zetian Li, Noel Loo, Jacob Marks, Edoardo Mosca, Samuel J. Paech, Paul Pak, Rom N. Parnichkun,

Alex Quach, Ryan Rogers, Daniela Rus, Nayan Saxena, Bettina Schlager, Tim Seyde, Jimmy T. H. Smith, Aditya Tadimeti, and Neehal Tumma. Lfm2 technical report, 2025. URL https://arxiv.org/abs/2511.23404.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Fuhao Chang, Shuxin Li, Yabei Li, and Lei He. Vlm-3d:end-to-end vision-language models for open-world 3d perception, 2025. URL https://arxiv.org/abs/2508.09061.

Jeffrey Chen and Rohan Chandra. Dynamic control barrier function regulation with vision-language models for safe, adaptive, and realtime visual navigation. *arXiv preprint arXiv:2603.21142*, 2026.

Ching-An Cheng, Mustafa Mukadam, Jan Issac, Stan Birchfield, Dieter Fox, Byron Boots, and Nathan Ratliff. Rmpflow: A computational graph for automatic motion policy generation, 2019a. URL https://arxiv.org/abs/1811.07049.

Richard Cheng, Gábor Orosz, Richard M. Murray, and Joel W. Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019b. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33013387. URL https://doi.org/10.1609/aaai.v33i01.33013387.

Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

James Dallas, John Talbot, Makoto Suminaka, Michael Thompson, Thomas Lew, Gábor Orosz, and John Subosits. Control barrier functions for shared control and vehicle safety. In *2025 American Control Conference (ACC)*, pp. 4203–4210. IEEE, 2025.

Daniel DeTone, Tianwei Shen, Fan Zhang, Lingni Ma, Julian Straub, Richard Newcombe, and Jakob Engel. Boxer: Robust lifting of open-world 2d bounding boxes to 3d, 2026. URL https://arxiv.org/abs/2604.05212.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator, 2017. URL https://arxiv.org/abs/1711.03938.

Tian Gao, Celine Tan, Catherine Glossop, Timothy Gao, Jiankai Sun, Kyle Stachowicz, Shirley Wu, Oier Mees, Dorsa Sadigh, Sergey Levine, and Chelsea Finn. Steervla: Steering vision-language-action models in long-tail driving scenarios, 2026. URL https://arxiv.org/abs/2602.08440.

Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. A review of safe reinforcement learning: Methods, theories, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):11216–11235, 2024.

Songqiao Hu, Zeyi Liu, Shuang Liu, Jun Cen, Zihan Meng, and Xiao He. Vlsa: Vision-language-action models with plug-and-play safety constraint layer. *arXiv preprint arXiv:2512.11891*, 2025.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17853–17862, 2023.

Bernhard Jaeger, Kashyap Chitta, and Andreas Geiger. Hidden biases of end-to-end driving models. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2023.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *Advances in Neural Information Processing Systems*, 37:819–844, 2024.

Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving, 2025. URL https://arxiv.org/abs/2503.07656.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. VAD: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8350, 2023.

Sicong Jiang, Zilin Huang, Kangan Qian, Ziang Luo, Tianze Zhu, Yang Zhong, Yihong Tang, Menglin Kong, Yunlong Wang, Siwen Jiao, et al. A survey on vision-language-action models for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4524–4536, 2025.

Kento Kawaharazuka, Jihoon Oh, Jun Yamada, Ingmar Posner, and Yuke Zhu. Vision-language-action models for robotics: A review towards real-world applications. *IEEE Access*, 2025.

Azal Ahmad Khan, Michael Andrev, Muhammad Ali Murtaza, Sergio Aguilera, Rui Zhang, Jie Ding, Seth Hutchinson, and Ali Anwar. Safety aware task planning via large language models in robotics. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 21024–21031. IEEE, 2025.

Luzia Knoedler, Oswin So, Ji Yin, Mitchell Black, Zachary Serlin, Panagiotis Tsiotras, Javier Alonso-Mora, and Chuchu Fan. Safety on the fly: Constructing robust safety filters via policy control barrier functions at runtime. *IEEE Robotics and Automation Letters*, 10(10):10058–10065, October 2025. ISSN 2377-3774. doi: 10.1109/lra.2025.3597847. URL http://dx.doi.org/10.1109/LRA.2025.3597847.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention, 2023. URL https://arxiv.org/abs/2309.06180.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International conference on robotics and automation (ICRA)*, pp. 9493–9500. IEEE, 2023.

Lightwheel. Lightwheel kitchen: 3d kitchen asset collection for nvidia isaac sim. GitHub repository, 2025. URL https://github.com/LightwheelAI/Lightwheel_Kitchen. Open-source 3D assets under CC BY-NC 4.0.

Yunze Man, Shihao Wang, Guowen Zhang, Johan Bjorck, Zhiqi Li, Liang-Yan Gui, Jim Fan, Jan Kautz, Yu-Xiong Wang, and Zhiding Yu. Locateanything3d: Vision-language 3d detection with chain-of-sight, 2026. URL https://arxiv.org/abs/2511.20648.

Nikola Milosevic, Johannes Müller, and Nico Scherf. Embedding safety into RL: A new take on trust region methods. *arXiv preprint arXiv:2411.02957*, 2024.

NVIDIA. Isaac Sim. URL https://github.com/isaac-sim/IsaacSim.

Özer Özkahraman and Petter Ögren. Combining control barrier functions and behavior trees for multi-agent underwater coverage missions. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 5275–5282. IEEE, 2020.

Ravi Pandya, Madison Bland, Duy P Nguyen, Changliu Liu, Jaime Fernández Fisac, and Andrea Bajcsy. From refusal to recovery: A control-theoretic approach to generative ai guardrails. *arXiv preprint arXiv:2510.13727*, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Zachary Ravichandran, David Snyder, Alexander Robey, Hamed Hassani, Vijay Kumar, and George J Pappas. Contextual safety reasoning and grounding for open-world robots. *arXiv preprint arXiv:2602.19983*, 2026.

Andrea Ruo, Jonathan Cacace, Magí Dalmau-Moreno, Lorenzo Sabattini, and Valeria Villani. An llm-based architecture for socially intelligent robot navigation based on social cues. In *2025 34th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1494–1500. IEEE, 2025.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization, 2017. URL https://arxiv.org/abs/1502.05477.

B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4):637–672, 2020. doi: 10.1007/s12532-020-00179-2. URL https://doi.org/10.1007/s12532-020-00179-2.

Xiaojuan Tang, Zilong Zheng, Jiaqi Li, Fanxu Meng, Song-Chun Zhu, Yitao Liang, and Muhan Zhang. Large language models are in-context semantic reasoners rather than symbolic reasoners, 2023. URL https://arxiv.org/abs/2305.14825.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Sushant Veer, Karen Leung, Ryan Cosner, Yuxiao Chen, Peter Karkus, and Marco Pavone. Receding horizon planning with rule hierarchies for autonomous vehicles, 2023. URL https://arxiv.org/abs/2212.03323.

Tsun-Hsuan Wang, Wei Xiao, Makram Chahine, Alexander Amini, Ramin Hasani, and Daniela Rus. Learning stability attention in vision-based end-to-end driving policies. In *Learning for Dynamics and Control Conference*, pp. 1099–1111. PMLR, 2023.

Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. Ab3dmot: A baseline for 3d multi-object tracking and new evaluation metrics, 2020. URL https://arxiv.org/abs/2008.08063.

Kiwan Wong, Wei Xiao, and Daniela Rus. Posafenet: Safe learning with poset-structured neural nets. *arXiv preprint arXiv:2601.22356*, 2026.

Wei Xiao and Calin Belta. High-order control barrier functions. *IEEE Transactions on Automatic Control*, 67(7):3655–3662, 2022. doi: 10.1109/TAC.2021.3105491.

Wei Xiao, Tsun-Hsuan Wang, Makram Chahine, Alexander Amini, Ramin Hasani, and Daniela Rus. Differentiable control barrier functions for vision-based end-to-end autonomous driving. *arXiv preprint arXiv:2203.02401*, 2022.

Wei Xiao, Tsun-Hsuan Wang, Ramin Hasani, Makram Chahine, Alexander Amini, Xiao Li, and Daniela Rus. Barriernet: Differentiable control barrier functions for learning of safe robot control. *IEEE Transactions on Robotics*, 39(3):2289–2307, 2023.

Runsen Xu, Zhiwei Huang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Vlm-grounder: A vlm agent for zero-shot 3d visual grounding. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 3961–3985. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/xu25c.html.

Kai S. Yun and Navid Azizan. Atom-cbf: Adaptive safe perception-based control under out-of-distribution measurements. In Gaurav Sukhatme, Lars Lindemann, Stephen Tu, Adam Wierman, and Nikolay Atanasov (eds.), *Proceedings of The 8th Annual Learning for Dynamics and Control Conference*, volume 331 of *Proceedings of Machine Learning Research*, pp. 1382–1395. PMLR, 17–19 Jun 2026. URL https://proceedings.mlr.press/v331/yun26a.html.

Zirui Zhao, Wee Sun Lee, and David Hsu. Large language models as commonsense knowledge for large-scale task planning, 2023. URL https://arxiv.org/abs/2305.14078.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.

A APPENDIX

A.1 IMPLEMENTATION DETAILS

Model. We tested two MLLM backbones. For Qwen, we tested `Qwen3-VL-2B,4B-Instruct` through Hugging Face `transformers` with 4-bit NF4 quantization using `bitsandbytes`, bf16 computation, and SDPA attention. Inference uses greedy decoding, KV caching, and a maximum of 384 generated tokens. For LFM, we use `LiquidAI/LFM2.5-VL-1.6B` in bf16 with a 4096-token context window and one image per prompt. Decoding is greedy with temperature 0, repetition penalty 1.1, and a generation limit of $100N_{\max}$ tokens, where $N_{\max} = 8$ is the maximum number of objects.

Prompts. We provide the template prompt below.

Prompt: Priority Safety Structure

```
SYSTEM PROMPT: You are a helpful autonomous vehicle assistant
    equipped with a forward-facing RGB camera. You are tasked with
    generating bounding boxes with ranking objects in the scene
    based on their safety importance by considering several
    factors, as you would if you were a human driver.
    NOTE: Each rank can only have 1 object assigned to it, and the
    ranks should be ordered from most important (1) to least
    important (N)

Prompt: You are given 2 consecutive frames ordered oldest to
    newest over a time horizon of 1 second. First, list what you
    see in the images. Then rank these objects from highest to
    lowest priority, i.e., how critical it is to satisfy the safety
    constraint for that object. For each object in the most recent
    frame, provide its 3D bounding box and output in JSON format:
    [{{"bbox_3d":[x_center, y_center, z_center, x_size, y_size,
    z_size, roll, pitch, yaw], "label": "name_of_object",
    "rank":<num>, "collision_critical":<boolean>}}]. The image
    resolution is 1600x900.

    The difference between collision-critical and
    non-collision-critical objects is as follows:
    * Collision-critical: Objects that the driver should avoid
    because a collision could cause harm to the vehicle, its
    occupants, or other road users  including damage that may
    impair driving safety.
    * Non-collision-critical: Objects that the driver could safely
    ignore because a collision would not cause noticeable physical
    damage or safety risk to the vehicle.

    The bounding box should be a safety bubble around the object.
    Make sure bounding boxes are unique. Base all bounding boxes
    only on the most recent frame. Answer concisely and precisely.
```

Object Dynamic. For future-state prediction, we use a simple heuristic that propagates detected objects under constant-velocity motion in the horizontal 2D plane. This heuristic is used only in our implementation and can be replaced with 3D object tracking methods, such as AB3DMOT (Weng et al., 2020).

A.2 HARDWARE EXPERIMENT

Figure 6 shows several of the real-world experimental setup. We use a UR5 robotic arm equipped with a RealHand L6 end-effector. An external RGB camera is positioned to the left of the robot to provide visual observations of the workspace and manipulated objects. Across the hardware experiments, we evaluate on 16 unique objects, including a peanut butter jar, large LEGO block, sliced bread, mug, jam jar, wine glass, electric drill, book, pink tennis ball, sponge, water bottle, and paper roll.

A.2.1 RESULTS

We provide some examples. Figure 5(2nd row, 1st column) shows a scenario in which the model assigns higher priority to the plastic cup than the ceramic cup. Although the cup is more fragile, the plastic cup contains water and therefore poses a greater risk to the surrounding workspace if spilled.

Example 1: Plastic cup with water vs. Cup


```
[
   {
      "bbox_3d": [-0.12, -0.05, 0.61, 0.08, 0.1, 0.09, 0.2, 0.05, 0.05],
      "label": "plastic_cup",
      "why": "Contains liquid; risk of spillage or breakage if disturbed.",
      "collision_critical": true,
      "rank": 1
   },
   {
      "bbox_3d": [0.05, -0.07, 0.65, 0.08, 0.08, 0.11, 0.2, 0.05, 0.05],
      "label": "ceramic_cup",
      "why": "Contains no liquid; but ceramic material may crack under force.",
      "collision_critical": true,
      "rank": 2
   }
]
```


Example 2: Identifying non-critical objects


```
[
  {
    "bbox_3d": [-0.24, -0.01, 0.71, 0.1, 0.1, 0.12, 0.23, 0.15, 0.1],
    "label": "cup",
    "why": "Small and light, but can spill if knocked, posing minor safety risk.",
    "collision_critical": true,
    "rank": 1
  },
  {
    "bbox_3d": [-0.07, -0.1, 0.75, 0.08, 0.12, 0.09, 0.23, 0.15, 0.1],
    "label": "jar",
    "why": "Contains liquid; tipping could cause spillage, posing minor safety risk.",
    "collision_critical": true,
    "rank": 2
  },
  {
    "bbox_3d": [0.15, -0.15, 0.81, 0.1, 0.12, 0.09, 0.23, 0.15, 0.1],
    "label": "glass",
    "why": "Glassware; potential for breakage if impacted, posing minor safety risk.",
    "collision_critical": true,
    "rank": 3
  },
  {
    "bbox_3d": [-0.12, -0.15, 0.78, 0.07, 0.07, 0.07, 0.23, 0.15, 0.1],
    "label": "ball",
    "why": "Small and light; minimal risk if touched, but could roll if disturbed.",
    "collision_critical": false,
    "rank": 4
  },
  {
    "bbox_3d": [-0.25, -0.05, 0.75, 0.08, 0.1, 0.08, 0.23, 0.15, 0.1],
    "label": "lego_block",
    "why": "Plastic; low risk of damage if touched, but may be handled with care.",
    "collision_critical": false,
    "rank": 5
  }
]
```

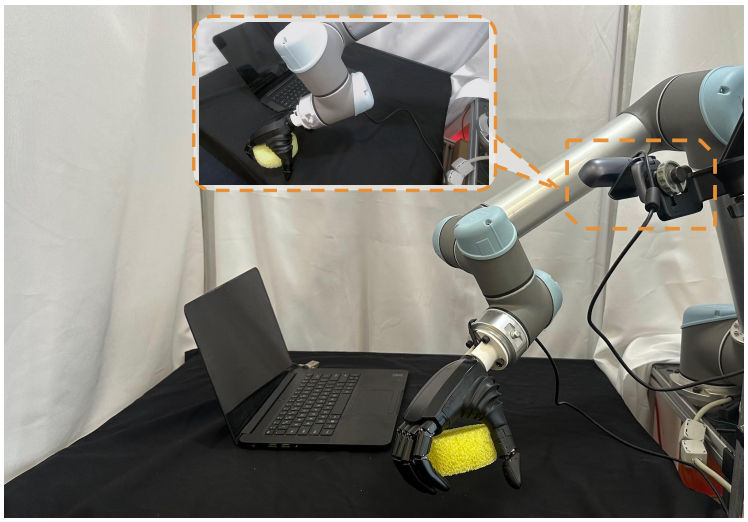

Figure 6: **Hardware Setup.** Real-world experimental setup consisting of a UR5 robotic arm equipped with a RealHand L6 end-effector and an external RGB camera positioned to the left of the robot.



Figure 7: **Scenarios in Real World.** Set of hardware experiments across diverse objects

A.3 Additional Experiments and Analysis

A.3.1 MLLM Output

Table 6: **Human evaluation of the MLLM's output.** Results across 15 hardware scenes evaluating for: (a) Collision-critical classification using participant responses as ground truth. (b) Fraction of objects correctly labeled by the LLM.

Metric	Score (%) ↑
Accuracy	96.3 ± 3.1
Precision	95.7 ± 3.7
Recall	100.0 ± 0.0
F1	97.8 ± 1.9

(a) Collision-critical classification

Metric	Score (%) ↑
Accuracy	100.0 ± 0.0

(b) Label accuracy

Using the hardware scenarios shown above, we evaluate the LLM's safety reasoning across different object configurations. We measure (i) collision-critical classification, comparing the predicted labels against human annotations, and (ii) object-label accuracy, indicating whether the model correctly

identifies each object. Four human annotators provide the ground-truth labels. As shown in Table 6, the model achieves 96.3% classification accuracy and 97.8% F1 score, while correctly identifying all objects in the evaluated scenes. Furthermore, recall was 100% which indicates that collision-critical objects were never identified as non-collision critical.

A.3.2 Runtime Analysis.

Table 7: **Runtime analysis.** Comparison of VLM backbones in terms of model size, inference latency and rate.

Backbone MLLM	Model Size	Inference Time (ms) ↓	FPS (Hz) ↑	ms/token ↓
Qwen3-VL	4B	10155.39 ± 76.11	0.10 ± 0.00	30.7
Qwen3-VL	2B	5212.57 ± 32.47	0.19 ± 0.00	25.2
LFM2.5-VL	1.6B	1341.37 ± 10.70	0.75 ± 0.01	12.1
LFM2.5-VL + vLLM	1.6B	346.10 ± 1.19	2.89 ± 0.01	3.1

We evaluate the runtime using an image and the CBF pipeline under the model configurations described in Appendix A.1. As shown in Table 5, LFM2.5-VL with vLLM provides the lowest latency among the evaluated VLM backends.

We therefore evaluate LFM2.5-VL + vLLM in CARLA on a subset of the case studies, consisting of one scenario from each of the four categories. The MLLM inference stage requires 564.15 ± 68.12 ms. Since LFM2.5-VL does not directly predict 3D bounding boxes, it outputs 2D detections that are lifted to 3D using Boxer (DeTone et al., 2026), which requires an additional 53.44 ± 2.58 ms. Including the remaining components of the safety pipeline, the overall system runs at 914.61 ± 66.90 ms per iteration, corresponding to 1.13 ± 0.07 Hz. On this subset, the vehicle achieves 100% route completion with a 0% collision rate, an average minimum clearance of 0.51 ± 0.02 m, and $b_c = 2.36 \pm 0.11$.

The proposed framework does not depend on a specific MLLM architecture; it assumes access to a perception module that provides object localization and semantic-priority outputs in the required format. We selected these models because it is openly available and works off the shelf, allowing us to prioritize an easy-to-understand implementation over a cost-optimized deployment, consistent with prior work (Xu et al., 2025). Recent work on specialized and fine-tuned vision-language models for 3D grounding suggests that faster inference is possible with model specialization and further systems optimization (Chang et al., 2025; Man et al., 2026).

A.3.3 Robustness to Perception Module

Table 8: **Feasibility Under Decreasing Gap Width.** Evaluation of *PRESTO* under progressively narrower traversable gaps. The ego vehicle has a width of approximately 2.16 m.

Gap Width (m)	4.79	3.79	3.29	2.54	2.29
Outcome	✓	✓	✓	✓	Infeasible

Table 9: **Robustness to Perception Errors.** We evaluate robustness to localization errors by perturbing the position and orientation of oracle bounding boxes. *PRESTO* maintains high collision-free rates and route completion under moderate perturbations, while performance degrades under increasingly severe localization errors.

Noise	Collision Rate (%) ↓	Route Completion (%) ↑	Conserv. (≥ 0 & ↓)	
			b_c	Avg. min. dist.
0.0 m, 0°	0.00 ± 0.00	100.00 ± 0.00	0.47 ± 0.09	2.83 ± 0.11
0.1 m, 5°	0.00 ± 0.00	100.00 ± 0.00	0.53 ± 0.05	3.25 ± 1.56
0.3 m, 10°	8.33 ± 14.43	96.14 ± 4.26	0.52 ± 0.23	3.81 ± 2.46
0.5 m, 45°	33.33 ± 28.87	75.93 ± 9.30	0.56 ± 0.43	3.44 ± 1.12
1.0 m, 90°	41.67 ± 14.33	60.69 ± 19.72	0.88 ± 0.58	6.07 ± 3.19

We further evaluate the sensitivity of *PRESTO* to perception accuracy. First, we test whether the MLLM-generated obstacle representations are sufficiently precise for constrained navigation by

progressively narrowing the traversable gap between obstacles. As shown in Table 8, the vehicle remains feasible for gaps as narrow as 2.54,m, compared with an ego-vehicle width of approximately 2.16 m, while the optimization becomes infeasible at 2.29 m. Between the gap width of 4.79 to 2.54 m the vehicle is able to safely navigate without collision.

Second, we directly perturb the estimated obstacle locations to evaluate robustness to perception error. Table 9 shows that performance is preserved under moderate perturbations and degrades as localization error increases. Together, these results indicate that the perception outputs are sufficiently accurate for tight navigation while the overall framework remains sufficiently robust to moderate perception errors.

A.3.4 Safety Improvements for Autonomy

Breakdown of scenario improvements.

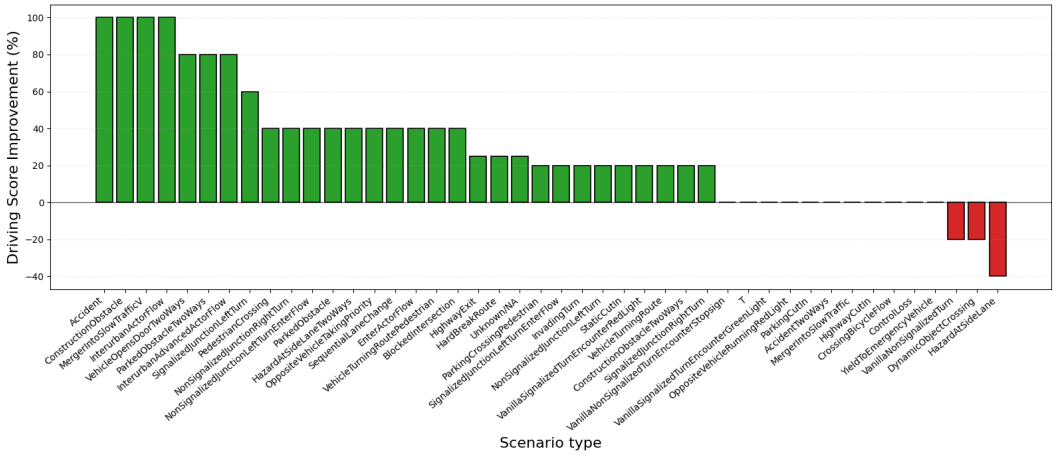

Figure 8: B2D Perfomrance Improvements.

A.4 More Case Visualization

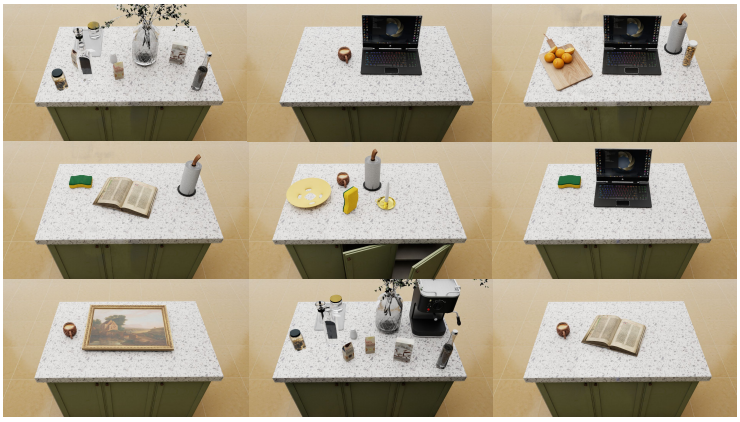

Figure 9: **Scenarios from Isaac Sim.**