From Refusal to Recovery: A Control-Theoretic Approach to Generative AI Guardrails

Ravi Pandya[1,*] **Madison Bland**[2] **Duy P. Nguyen**[2] **Changliu Liu**[1]
Jaime Fernández Fisac[2,3] **Andrea Bajcsy**[1,3]
[1]Robotics Institute, Carnegie Mellon University
[2]Department of Electrical and Computer Engineering, Princeton University
[3]Equal Advising
[*]Corresponding author: ravi.pandya922@gmail.com

Abstract

Generative AI systems are increasingly assisting and acting on behalf of end users in practical settings, from digital shopping assistants to next-generation autonomous cars. In this context, safety is no longer about blocking harmful content, but about preempting downstream hazards like financial or physical harm. Yet, most AI guardrails continue to rely on output classification, based on labeled datasets and human-specified criteria, making them brittle to new hazardous situations. Even when unsafe conditions are flagged, this detection offers no path to recovery: typically, the AI system simply refuses to act—which is *not* always a safe choice. In this work, we argue that agentic AI safety is fundamentally a *sequential decision problem*: harmful outcomes arise from the AI system's continually evolving interactions and their downstream consequences on the world. We formalize this through the lens of safety-critical control theory, but within the AI model's latent representation of the world. This enables us to build *predictive guardrails* that (i) monitor an AI system's outputs (actions) in real time and (ii) proactively correct risky outputs to safe ones, all in a model-agnostic manner so the *same guardrail* can be wrapped around *any* AI model. We also offer a practical training recipe for computing such guardrails at scale via safety-centric reinforcement learning. Our experiments in simulated driving and e-commerce settings demonstrate that control-theoretic guardrails can reliably steer LLM agents clear of catastrophic outcomes (from collisions to bankruptcy) while preserving task performance, offering a principled dynamic alternative to today's flag-and-block guardrails.

1 Introduction

Autonomous agents powered by modern generative AI promise to assist users across a wide range of settings. Typically built around a large language model (LLM), these systems can process rich context and produce sophisticated outputs to negotiate financial deals [1, 2], make purchases [3], write software [4, 5], provide driver assist suggestions [6, 7], and even issue control commands to robots and autonomous vehicles [8–10]. Their rapid progress and widespread availability raise urgent questions about AI safety [11, 12].

Among approaches to AI safety [13], guardrails are appealing because they are simple, post-hoc mechanisms that filter model inputs or outputs at deployment time [14–16], refusing generations deemed "unsafe" (e.g., step-by-step instructions to build a bomb). Unfortunately, today's guardrails are typically trained on textual proxies [17, 18] or labels from one-step simulation [1]. These signals miss what ultimately matters: not just the *content* of an AI output, but its downstream *consequences*,

Article presented at the Second IASEAI Conference (IASEAI'26).

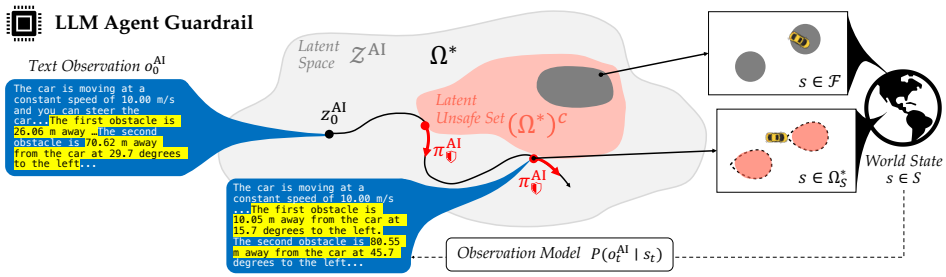

Figure 1: **Overview of Control-Theoretic Predictive Guardrails for AI Agents.** This guardrail, which shields an LLM agent, operates on text-based observations of the world, but learns a latent-space predictive safety monitor and recovery policy from real-world signals reflecting the outcomes of its actions (e.g., obstacle collisions).

such as financial loss, privacy breaches, or physical harm. Moreover, when a guardrail flags an unsafe situation, the default response is refusal, blocking the agent's output altogether. Yet inaction can itself be dangerous: an AI agent that refuses to steer a self-driving car approaching an obstacle at high speed can cause an accident *through* inaction.

In this paper, we argue that *generative AI safety is inherently a sequential decision problem*: harmful outcomes arise from an AI agent's evolving interactions with its environment. This agentic view applies whether the AI outputs executable commands (e.g., function calls) or user-facing content (e.g., text), and we demonstrate its value in both settings. It also connects directly to *safe control theory* [19], especially *safety filters* [19–23], which detect when a base policy's proposed actions will violate safety constraints and replace them with safe alternatives.

We instantiate our control-theoretic *predictive guardrail* through a scalable safety-centric reinforcement learning framework that learns (i) a monitor predicting whether current LLM agent generations will cause future safety violations, and (ii) a fallback policy that supplies safe alternatives. Crucially, the resulting guardrail is model-agnostic: the same guardrail can shield multiple LLM agents without access to their weights or training data. We evaluate this synthesis approach in three simulated domains where LLM actions affect world outcomes: autonomous vehicle control, agentic e-commerce, and AI driver assistance. Figure 1 shows the autonomous vehicle domain (a standard toy problem in control systems safety), where we can measure ground-truth downstream consequences and compare against optimally safe behavior. We use e-commerce to study whether control-theoretic guardrails can be learned in an agent's latent world representation, and driver assistance to test whether they can learn safe strategies when actions only *indirectly* affect the environment. Across all settings, consequence-aware guardrails trained on downstream outcomes are more reliable monitors than proxy-based guardrails, achieving higher predictive accuracy without sacrificing task performance.

Our work bridges two traditionally disjoint communities. For control systems experts, it shows that safety filters can extend to text-based world representations rather than only physical state representations. For AI safety experts, it shows that guardrails can do more than refuse: they can jointly learn to detect danger and recover with safe alternatives, using safety-centric RL instead of supervised proxy labels or hard-coded heuristics.

Statement of Contributions. To summarize, our contributions are: 1) We formalize generative AI safety as a sequential decision problem and introduce *predictive guardrails* as a generalization of control-theoretic *safety filters* to LLM-based representations. 2) We propose a scalable recipe for training LLM guardrails via *safety-centric reinforcement learning*, yielding a single safety mechanism that can filter *any* base LLM agent model. 3) In experiments with state-of-the-art LLMs, we show that predictive guardrails trained on downstream outcomes reliably detect and correct unsafe behavior in self-driving, e-commerce, and assistive AI settings, surpassing flag-and-block baselines in both safety and performance.

2 Related Work

Safety Guardrails for LLMs and LLM Agents. AI guardrails are deployment-time mechanisms for ensuring that pre-trained foundation models behave safely [15]. *Input guardrails* act before

inference, analyzing user prompts to identify or reject malicious, unsafe, or policy-violating inputs [24]. *Output guardrails* act after generation, preventing unsafe outputs from being shown to users or executed [25]. For non-agentic LLMs, guardrails have used string perplexity [26] or text classifiers [14, 17, 27–29] for detection. Recent control-inspired methods steer LLM generations toward safer output regions [18, 30–32] and learn enforceable decoding constraints [33]. For agentic LLMs, existing work mainly classifies and blocks unsafe actions using text proxies or one-step outcome labels [34–36]. These guardrails rely on *refusal*: rejecting inputs or blocking generations, without *recovering* from unsafe states. We instead co-optimize detection and recovery for LLM agents via safety-centric reinforcement learning over multi-step outcomes, yielding guardrails that identify unsafe behavior and prescribe corrective actions.

Safety-Critical Control. Safe control studies how embodied systems can make safe decisions, often using tools from dynamical systems and differential games. A central mechanism is the *safety filter*, which detects and minimally modifies actions that risk future constraint violations [19, 20, 37]. Safety filters combine a value function that identifies states from which failure is unavoidable [22, 38–45] with a fallback policy that steers away from violations [23, 46]. They can be implemented via Hamilton-Jacobi reachability [21, 47], control barrier functions [48, 49], or model predictive shielding [50]. Recent work has scaled these methods using reinforcement learning [51–53], self-supervised learning [54], and high-dimensional observations such as LiDAR [55, 56] or RGB images [57, 58]. We extend this line of work by computing safety filters directly in the high-dimensional *textual representation* of LLM agents.

Reinforcement Learning for LLMs. Reinforcement learning from human feedback (RLHF) has largely focused on single-turn preference optimization [59, 60], a paradigm inherited by early agentic LLMs. However, LLM agents often require multi-step environment interaction to accomplish complex tasks. Recent work applies multi-turn RL to elicit such behaviors [61–65], training agents to navigate web environments, use tools, and plan over long horizons. While this work uses multi-turn RL to improve *task performance*, we use it to train safety-focused guardrails. Our approach learns agents that explicitly pursue *safety* objectives: detecting states from which violations are unavoidable and executing recovery policies that steer back to safety.

3 Formalism: A Control-Theoretic Model of LLM Agent Guardrails

In this work, we argue that agentic AI safety is fundamentally a *sequential decision-making problem*: unsafe outcomes arise from the evolving sequence of actions and their downstream consequences on the world. We first formalize the necessary components of this model as a partially observable Markov decision process (POMDP). We then characterize the conditions under which an LLM agent can maintain safety (i.e. abide by constraints) and prescribe the most effective AI policy to do so. This culminates in our formulation of LLM agent guardrails as control-theoretic *safety filters*.

3.1 Dynamical System Model of LLM Agent Interaction

To formulate the computation of generative AI guardrails as a decision-making problem, we use the terminology of POMDPs. Let our *safety-crticial* POMDP be a tuple $\langle \mathcal{S}, \mathcal{A}^{\texttt{AI}}, \mathcal{U}, T, \mathcal{O}^{\texttt{AI}}, \ell_{\mathcal{F}}, \rho, \gamma \rangle$.

World State. $s \in \mathcal{S}$ is the true state of the world. For example, this can include the physical geometry of the environment in the case of the AI controlling a physical robot, or the human's true internal state (e.g., how risk tolerant they are) in the case of an AI assistant.

AI Agent Action. $a^{\texttt{AI}} \in \mathcal{A}^{\texttt{AI}}$ is the agent's action for interaction with the world. For LLM agents, this is a sequence of tokens (up to a max length K) representing their desired action, like clicking on a webpage element, and $\mathcal{A}^{\texttt{AI}}$ is the set of all sequences of length $\leq K$ from the agent's vocabulary.

Physical Action. $u \in \mathcal{U}$ is the physical action. In many (particularly embodied) environments, the AI agent's action $a^{\texttt{AI}}$ must be converted to a physical action that changes the world state, s. For example, if the AI agent is placed in direct control of a car, its text output (e.g., $a^{\texttt{AI}} =$ ‘‘`steer left`’’) will be converted to physical steering commands (e.g., $u = -1$ rad/s). Or, if the AI acts as an assistant providing advice to a human user (who could be, e.g., driving a car), then the physical actions will be the human's control commands executed in response to the AI's output (e.g., the human driver deciding to steer left based on the AI's suggestion).

Dynamics. $T\big(s_{t+1} \mid s_t, f_{\mathcal{U}}(s_t, a_t^{\texttt{AI}})\big)$ is the discrete-time dynamics function (or state transition map) which evolves the state of the world based on the physical action, which itself is influenced by the AI's actions. For example, in the case of an LLM agent participating in e-commerce, then the true dynamics include the outcomes of the agent purchasing items, like the total amount of money in the user's checking account decreasing. The world state and AI action induce a physical action via the *action map*, $f_{\mathcal{U}} : \mathcal{S} \times \mathcal{A}^{\texttt{AI}} \to \mathcal{U}$.

AI Observation. $o^{\texttt{AI}} \in \mathcal{O}^{\texttt{AI}}$ is the AI agent's observation. The agent never gets perfect access to the true world state s but instead observes it via a suite of sensors. For example, for LLM agents, which operate primarily on text inputs, $o^{\texttt{AI}}$ would be a textual description of the world, like the HTML of a website or a textual description of a physical scene. The corresponding observation model $P(o^{\texttt{AI}} \mid s)$ produces these textual observations of the world, conditioned on the true state.

Failure Margin Function. $\ell_{\mathcal{F}} : \mathcal{S} \to \mathbb{R}$ is akin to the reward function in traditional POMDPs, but in safety-critical control is called a *margin function*. It encodes distance to the failure set $\mathcal{F} \subset \mathcal{S}$, which defines the safety specification. For example, if an LLM shopping assistant must keep the user's bank balance above their monthly cost of living, then $\mathcal{F}$ contains balances below that threshold, and $\ell_{\mathcal{F}}$ returns the signed distance to it. Crucially, failure is specified in *outcome space*, not text-token space, allowing the guardrail to account for long-term real-world consequences, such as how digital purchases affect the user's bank balance.

Initial State & Discounting. $\rho \in \Delta(\mathcal{S})$ is initial state distribution and $\gamma \in [0, 1]$ is discount factor.

3.2 When Can A Guardrail Ensure Safety? Characterizing an AI Agent's Safe Set

When can we rely on a guardrail to prevent an AI agent from causing harm? The answer requires us to characterize the set of states from which a sequence of safe actions exists at all. This also forms the foundation for our connection to control-theoretic safety. In this subsection, we formalize the AI agent's *safe set*, i.e., the maximal region of states from which safety can be maintained, along with the associated notion of optimal safety policy. These two entities—safe set and optimal safety policy—will establish our control-theoretic guardrail instantiated in Section 3.3.

Latent Agent State & Policy. While the true state and transitions from Section 3.1 are not generally known to the AI, the AI may (explicitly or implicitly) estimate them during interaction. Specifically, we model the AI agent as maintaining an internal state representation of the world based on its experience (e.g., all prior conversations, textual descriptions of the state of the world, and any sensor measurements). Formally, let this latent state at any time t be denoted by $z_t^{\texttt{AI}} := \mathcal{E}(o_{t-H:t}^{\texttt{AI}}, a_{t-H:t-1}^{\texttt{AI}})$ which is the encoding of the H-step history of observations and actions up to time t (e.g., H can be the context window size of the LLM). Thus, we can denote any agent policy by $\pi(a_t^{\texttt{AI}} \mid z_t^{\texttt{AI}})$.

Maximal Latent Safe Set. Recall that in our formulation from Section 3.1, safety was encoded as a state constraint $s \notin \mathcal{F} \subset \mathcal{S}$; for example, a person's bank account being depleted, or an autonomous vehicle colliding with obstacles. Since from the AI's standpoint it is only making decisions within its latent representation of the world, $z^{\texttt{AI}}$, then we will characterize the safe set from it's perspective. Specifically, we want to find the set of all safe latent states $z_0^{\texttt{AI}} \in \mathcal{Z}^{\texttt{AI}}$ from which there ***exists*** a best-effort AI policy which can keep the system away from a future failure. Mathematically, this **maximal latent safe set** is characterized as

$$\Omega^* := \{z_0^{\texttt{AI}} \in \mathcal{Z}^{\texttt{AI}} : \exists \pi^{\texttt{AI}}, \forall \tau \geq 0,\ \mathcal{D}(z_\tau^{\texttt{AI}}) \notin \mathcal{F}\}, \tag{1}$$

where $\mathcal{D} : \mathcal{Z}^{\texttt{AI}} \to \mathcal{S}$ is a decoder which maps from the latent space into the world state space[1]. If $z_0^{\texttt{AI}} \in \Omega^*$, then there exists some AI policy $\pi^{\texttt{AI}} : z^{\texttt{AI}} \mapsto a^{\texttt{AI}}$ that keeps the AI agent within $z_\tau^{\texttt{AI}} \in \Omega^*$, and thus away from the failure set $\mathcal{F}$ for all time $\tau \geq 0$. In other words, if the AI agent's initial latent state is within the latent safe set, then there exists a safety-preserving action the guardrail can take.

[1] In general, the mapping from latent space to real state space could be set-valued, since one latent state can imply multiple real states due to the agent's uncertainty. In this case, Equation (1) would require $\mathcal{D}(z_\tau^{\texttt{AI}}) \cap \mathcal{F} = \varnothing$. To streamline the paper's exposition, we treat the decoded state as a singleton estimate, noting that the robust set-valued case corresponds to the well-studied minimax safety value function [40, 66].

Safety Value Function. We now turn to the computational question: how can the maximal safe set from Equation (1) be determined? It turns out that both this set and its corresponding safety-preserving AI policy are jointly characterized by the solution of an optimal control problem. Specifically, we will utilize the failure margin function $\ell_{\mathcal{F}}(s)$ from Section 3.1 to encode *distance to failure* in $\mathcal{S}$. Then, the control problem is to determine the *closest* the AI agent ever gets to failure starting from any initial latent state $z_0^{\mathtt{AI}}$ and trying its hardest to *avoid* this failure. This is captured by the *safety value function* [38, 40, 66–69]:

$$V^{\shield}(z_0^{\mathtt{AI}}) := \max_{\pi^{\mathtt{AI}}} \left(\min_{t \geq 0} \ell_{\mathcal{F}}(s_t) \right), \qquad \Omega^* = \left\{ z_0^{\mathtt{AI}} \in \mathcal{Z}^{\mathtt{AI}} : V^{\shield}(z_0^{\mathtt{AI}}) \geq 0 \right\}, \tag{2}$$

where $\mathcal{D}(z_t^{\mathtt{AI}}) = s$ and the super-zero level set of the value function encodes the maximal safe set. If the value $V^{\shield}(z_0^{\mathtt{AI}})$ is negative (i.e., $z_0^{\mathtt{AI}} \notin \Omega^*$), this means that, no matter what the AI agent chooses to do, it cannot avoid eventually entering $\mathcal{F}$. Critically, the optimization posed in Equation (2) quantifies the best the AI system could ever do to maintain safety—hence, the *maximal* safe set.

The value function defined above satisfies the fixed-point Bellman equation relating the current safety margin $\ell_{\mathcal{F}}$ to the minimum-margin-to-go V after one discrete timestep, and enables the extraction of a corresponding safety-centric policy:

$$V^{\shield}(z^{\mathtt{AI}}) = \max_{a^{\mathtt{AI}} \in \mathcal{A}^{\mathtt{AI}}} \underbrace{\min \left\{ \ell_{\mathcal{F}}(s_+), \mathop{\mathbb{E}}_{o_+^{\mathtt{AI}}, s_+} \left[V^{\shield}\left(z_+^{\mathtt{AI}}\right) \right] \right\}}_{Q^{\shield}(z^{\mathtt{AI}}, a^{\mathtt{AI}})}, \qquad \pi_{\shield}^{\mathtt{AI}}(z^{\mathtt{AI}}) = \operatorname*{argmax}_{a^{\mathtt{AI}} \in \mathcal{A}^{\mathtt{AI}}} Q^{\shield}(z^{\mathtt{AI}}, a^{\mathtt{AI}}), \tag{3}$$

where $z_+^{\mathtt{AI}}$ is the encoding including the most recent observation $o_+^{\mathtt{AI}} \sim P(\cdot \mid s_+)$ generated from the next state $s_+ \sim T(\cdot \mid s, a^{\mathtt{AI}})$ that is influenced by the AI's actions.

3.3 AI Predictive Guardrails as Control-Theoretic Safety Filters

The systematic detect-and-recover AI guardrail we aim to achieve parallels the safety filter mechanisms long established in control theory [19, 20]. A safety filter continuously monitors an agent's proposed actions and, when necessary, overrides them with safe alternatives to prevent future constraint violations. Formally, a **safety filter** is a tuple $(\pi_{\shield}^{\mathtt{AI}}, \Delta, \phi)$ where: the *fallback policy* $\pi_{\shield}^{\mathtt{AI}} \colon \mathcal{Z}^{\mathtt{AI}} \to \mathcal{A}^{\mathtt{AI}}$ provides last-resort recovery actions; the *safety monitor* $\Delta \colon \mathcal{Z}^{\mathtt{AI}} \times \mathcal{A}^{\mathtt{AI}} \to \mathbb{R}$ evaluates whether a candidate action preserves safety; and the *intervention scheme* $\phi \colon \mathcal{Z}^{\mathtt{AI}} \times \mathcal{A}^{\mathtt{AI}} \to \mathcal{A}^{\mathtt{AI}}$ implements the intervention when the monitor detects an unsafe action.

Our framework from Equation (3) provides a principled derivation of all three components jointly. The safety value function and its optimal policy implicitly define the *least-restrictive* safety filter–one granting maximal autonomy to the task policy while preempting all safety violations. Specifically, we obtain $\pi_{\shield}^{\mathtt{AI}}$ from the safety value optimization, $\Delta \leftarrow Q^{\shield}(\cdot, \cdot)$, and $\phi \leftarrow \mathbb{1}_{\Delta > 0} \cdot \pi_{\checkmark}^{\mathtt{AI}} + \mathbb{1}_{\Delta \leq 0} \cdot \pi_{\shield}^{\mathtt{AI}}$. This distinguishes our approach from existing LLM guardrails [14, 17, 36], which typically provide only detection mechanisms (i.e., Δ) without prescribing recovery policies ($\pi_{\shield}^{\mathtt{AI}}$), leaving the critical question of "what to do when unsafe" unresolved.

4 A Practical Training Recipe for Computing LLM Agent Safety Guardrails

Section 3 characterizes the ideal safety guardrail, but leaves open the question of how to practically compute it. Here, we establish a *safety-centric reinforcement learning* (RL) methodology that approximates the safety value function and its corresponding policy, resulting in our control-theoretic predictive guardrail. We adopt the terminology and structure of double deep Q-learning (DDQN) [70], although many ingredients we present here naturally extend to a broader class of RL algorithms. We discuss a few key details here.

Time-Discounted Safety Bellman Equation. Since Equation (3) does not inherently induce a contraction mapping, we cannot directly apply off-the-shelf RL solvers. Following prior work [51, 52], we instead train the model to minimize the *time-discounted* safety Bellman equation, which does induce a contraction and is therefore compatible with a large suite of RL solvers:

$$Q^{\shield}(z^{\mathtt{AI}}, a^{\mathtt{AI}}) = (1-\gamma)\ell_{\mathcal{F}}(s_+) + \gamma \min \left\{ \ell_{\mathcal{F}}(s_+), \max_{a_+^{\mathtt{AI}} \in \mathcal{V}} Q^{\shield}\left(z_+^{\mathtt{AI}}, a_+^{\mathtt{AI}}\right) \right\}. \tag{4}$$

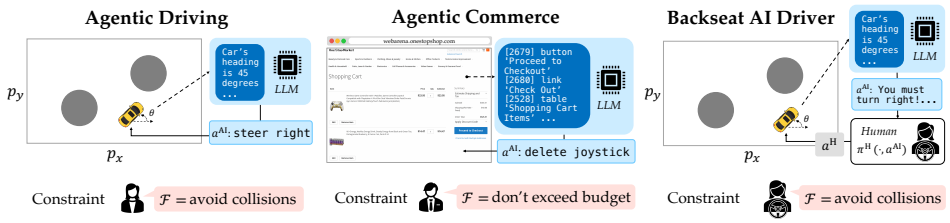

Figure 2: **Environments.** Three LLM agent environments where we evaluate our guardrails.

Training-Time vs. Inference-Time Decoding. At training time, we generate tokens directly from the safety-centric LLM model: $a^{\texttt{AI}}_{t,k+1} = \text{argmax}_{a \in \mathcal{V}'} Q^{\textcircled{\scriptsize S}}(z^{\texttt{AI}}_{t,k}, a)$. This enables the RL optimization to focus purely on maximizing the safety objective. At inference time, we blend samples from the base model—to generate diverse, coherent text—but guide the sampling based on the learned safety value. Specifically, $a^{\texttt{AI}}$ is generated by blending the safety value function's logits with the base model's logits for standard stochastic generations:

$$a^{\texttt{AI}}_{t,k} \sim \pi^{\texttt{AI}}_{\textcircled{\scriptsize S}}(\cdot \mid z^{\texttt{AI}}_{t,k}) \propto \exp\left\{\frac{1}{T}\left(Q_{\text{b}}(z^{\texttt{AI}}_{t,k}, a^{\texttt{AI}}_{t,k}) + \beta Q^{\textcircled{\scriptsize S}}(z^{\texttt{AI}}_{t,k}, a^{\texttt{AI}}_{t,k})\right)\right\}, \tag{5}$$

where $T, \beta \in \mathbb{R}$ are the temperature parameter and blending hyperparameter respectively.

5 Experimental Setup

We study our approach in three scenarios: (1) a LLM agent that directly controls an embodied system, (2) digital LLM commerce agent, and (3) a LLM agent that assists a user doing an embodied task. Throughout, our proposed control-theoretic LLM agent guardrail is a fine-tuned Llama-3.2-1B-Instruct model [71] with LoRA (Low-Rank Adaptation [72]).

Agentic Driving. The LLM agent steers a planar vehicle through obstacles (Fig. 2). The state $s = [p_x, p_y, \theta]$ follows discrete-time dynamics. Each step, the agent answers a multiple-choice query $a^{\texttt{AI}}$ mapped to $u \in \{-u_{\max}, 0, u_{\max}\}$ [73, 74]. The goal is to drive rightward. The observation $o^{\texttt{AI}}$ is a text-only summary of the task, pose (position, heading), relative angles/distances to obstacles, and distances to road boundaries, tokenized into $z^{\texttt{AI}}$ as a 370×1024 embedding sequence for Llama-3.2-1B-Instruct. Safety requires no collisions or off-road; the failure margin $\ell_{\mathcal{F}}$ is the minimum distance to any obstacle or boundary. Details of the environment and prompts are in the appendix.

Agentic Commerce. Building on WebArena [1] and inspired by the recent ChatGPT's e-commerce integration [3], we simulate a shopping site. The observation $o^{\texttt{AI}}$ is the page's accessibility tree in text ($\approx$ 500-900 tokens), encoded as $z^{\texttt{AI}}$. The LLM agent guardrail must keep the cart under the user's budget, with a fixed window to modify the cart before checkout , inducing time-critical, long-horizon reasoning. The failure margin function $\ell_{\mathcal{F}}(z^{\texttt{AI}}) = \text{budget} - \text{cart total}$.

Backseat AI Driver. Extending the Agentic Driving scenario, the LLM now serves as an advisor rather than the actuator. We model a user who takes physical actions in the environment after receiving advice from the AI. Let the human be modeled by the policy $\pi^{\texttt{H}} : \mathcal{O}^{\texttt{H}} \times \mathcal{A}^{\texttt{AI}} \to \mathcal{A}^{\texttt{H}}$ mapping from the human's observations and the AI agent's recommendation to the human's action space. The AI's available actions, $a^{\texttt{AI}} \in \mathcal{A}^{\texttt{AI}}$, are full open-vocabulary sentences and the transition function $T(s_{t+1} \mid s_t, \pi^{\texttt{H}}(o^{\texttt{H}}, a^{\texttt{AI}}))$ is influenced *implicitly* by the AI assistant through the human.

6 Experimental Results

Our experiments study the following series of questions: (1) How well can control-theoretic LLM guardrails be learned within a textual representation of the world?, (2) How well do guardrails balance safety and efficiency when used in-the-loop with LLM agents?, and (3) Can control-theoretic LLM guardrails learn safety strategies when actions *indirectly* influence the environment?

Table 1: **Agentic Driving.** Quantitative metrics show that training with long-term consequences yields a more accurate and less conservative guardrail.

Model	Success Rate (↑)	Failure Rate (↓)	Value F1 (↑)	Accuracy (↑)	Conservativeness (↓)
Privileged	85.2%	13.2%	1.00	TP: 1.00 TN: 1.00	FN: 0.00
Myopic-Privileged	81.9%	16.2%	N/A	N/A	N/A
GPT-5 (Think: minimal)	39.5%	59.4%	0.29	TP: 0.02 TN: **0.92**	FN: 0.94
GPT-4o	31.7%	68.2%	0.29	TP: 0.00 TN: 0.01	FN: 1.00
GPT-4o-mini	23.4%	70.2%	0.29	TP: 0.25 TN: 0.74	FN: 0.77
Llama-3.1-70B-Instruct	22.8%	76.8%	0.0	TP: 0.95 TN: 0.05	FN: 0.03
Qwen-2.5-72B-Instruct	22.3%	72.5%	0.78	TP: 0.52 TN: 0.58	FN: 0.58
Myopic-Realistic	44.4%	55.6%	N/A	N/A	N/A
ReGuard (ours)	**77.3%**	**18.8%**	**0.99**	TP: **0.99** TN: 0.56	FN: **0.01**

6.1 Control-Theoretic Guardrails Learn Safe Sets and Recovery Policies from Text Inputs

6.1.1 Agentic Driving

Setup In the **Agentic Driving** environment, we can tractably approximate the ideal safety filter (which intervenes only when necessary to prevent a future failure) assuming privileged access to the true world space. Our prompts used for all models are in the Appendix.

Baselines. We compare **ReGuard** (**Re**covery **Guard**rail) against eight baselines. **Privileged** is an upper-bound guardrail with access to the true world state s. It uses the same reach–avoid RL algorithm as our method, but its safety filter observes s directly and outputs physical controls u. We also evaluate five zero-shot foundation-model baselines, **ZeroShot-**X, where $X \in \{$**GPT-5**, **GPT-4o**, **GPT-4o-mini**, **Llama3-70B**, **Qwen2.5-72B**$\}$. We prompt each **ZeroShot-**X baseline to act as a safety filter by *implicitly* inferring both a monitor Δ and a safety-preserving policy $\pi^{\texttt{AI}}_{\textbf{0}}$. Finally, we compare to **Myopic**, inspired by prior work [18, 32, 33], which uses a text-based reward to classify a single generation $a^{\texttt{AI}}$ as safe or unsafe, then finetunes the recovery policy to produce high-reward action generations. We train two variants: one policy that is supervised by the action labels from the **Privileged** policy $\pi^{\texttt{AI}}_{\textbf{0}}$ (**Myopic-Privileged**), and one that is trained via an LLM-as-a-judge [75] which scores whether $a^{\texttt{AI}}$ will keep the system safe in the future (called **Myopic-Realistic**). Both approaches are fine-tuned via PPO with the `trl` library [76].

Metrics. **Value F1:** the predictive quality of any monitor and comparing it to the **Privileged** monitor. **ZeroShot-**X models are asked a yes/no questions (e.g., "Is there an action that can keep the system safe in the future?") while we look at the sign of the largest Q-value for **ReGuard**. **Monitor accuracy:** true positive/negative rates of the safety assessment for each proposed action along the evaluation trajectories, compared to the true outcome of attempting to keep the system safe by using the guardrail's recovery policy allowing the candidate action. **Monitor conservativeness:** false negative rate of the safety assessment compared to whether it would have been possible for the *best* recovery policy to maintain safety after allowing the candidate action. **Success rate**: fraction of trajectories safely reaching the goal. **Failure rate**: fraction of trajectories that fail.

Results. Table 1 shows the results. The **Myopic** baselines are trained purely to be recovery policies, so we do not measure their performance on monitor metrics. **ReGuard** has the highest success rate, lowest failure rate, highest value F1 score, and lowest conservativeness compared to all non-**Privileged** baselines. **ReGuard** has the highest true positive rate on accuracy, but a relatively low true negative rate. This means that while the value function is close to the privileged policy's value, it overestimates the fallback policy's ability to recover. The performance of the **Myopic** policies depends heavily on the reward model since it serves as a *proxy* for the true downstream outcome of the multi-step interaction between the agent and the environment, which occurs in the true state space. When the reward model has perfect information about the downstream outcome from the **Privileged** $\pi^{\texttt{AI}}_{\textbf{0}}$, this signal is sufficient to learn a high-quality recovery policy. However, this paradigm is highly unrealistic, since even the best **ZeroShot** models are unable to properly judge the safety of actions on their own, as seen in the value F1, accuracy and conservativeness metrics in Table 1. Indeed, when the **Myopic** policy is trained without this privileged reward function, its performance drops significantly.

Figure 3: **Agentic Commerce.** One rollout of the base LLM agent vs. **ReGuard** vs. GPT-4o.

6.1.2 Agentic Commerce

Setup. We next study an agentic commerce setting where we can measure whether the agent has successfully avoided failure by keeping the user's shopping cart under the specified budget ($50).

Baselines. We compare **ReGuard** to four foundation models of varying sizes zero-shot, **ZeroShot-**X, where $X \in \{$**GPT-4o**, **Llama-3.1-8B-Instruct**, **Llama-3.1-3B-Instruct**, **Llama-3.1-1B-Instruct**$\}$. Detailed prompts are included in the Appendix.

Metrics. We measure the **Success Rate** as the percent of initial conditions where the cart total at the final timestep is under the allotted budget of $50.

Table 2: **Agentic Commerce.** Success rate across 8 initial cart conditions.

Model	Success Rate (↑)
Llama-3.1-8B-Instruct	62.5%
Llama-3.2-3B-Instruct	50%
Llama-3.2-1B-Instruct	62.5%
GPT-4o	**87.5%**
ReGuard	**87.5%**

Quantitative Results. Table 2 shows a $\sim 25\%$ improvement in the safety success rate for **ReGuard** compared to the **ZeroShot-Llama-**X baselines. GPT-4o matches the performance of our control-theoretic guardrail that is obtained after fine-tuning a significantly smaller open-sourced model for safety. This suggests that for some classes of tasks and safety constraints, particularly those well-represented in public web data used to train GPT-4o [77], near-optimal recovery policies may emerge zero-shot.

Qualitative Results. We visualize the decisions of **Llama-3.1-8B-Instruct**, **ReGuard** and **GPT-4o** from one initial condition of the cart in Figure 3. The agent is only able to remove a maximum of 5 items from the cart, so this requires removing 5 of the largest items from the cart to stay within the user's budget. The **Llama-3.1-8B-Instruct** model removes items without understanding that it will leave larger items in the cart at checkout at $t = 5$, leading to failure. **ReGuard** removes items that will allow the final state of the cart to stay under budget, and **GPT-4o** removes the same five items, but additionally prioritizes removing the highest-price items first.

6.2 Control-Theoretic Guardrails Balance Safety & Efficiency When Shielding LLM Agents

Setup. We study the **Agentic Driving** setting. We deploy all guardrails to safeguard three increasingly large open-source LLM agent models, treating them as the task-driven policy: $\pi^{\texttt{AI}}_{\boxdot} \in \{$Llama-3.2-**1B**-Instruct, Llama-3.2-**3B**-Instruct, Llama-3.1-**8B**-Instruct$\}$. All guardrails are implemented via the switching mechanism from Section 3.3 where they only activate when the safety monitor says failure is imminent; otherwise $\pi^{\texttt{AI}}_{\boxdot}$ is executed.

Baselines. We compare to a state-of-the-art LLM guardrail, **LlamaGuard** [17]. We note that **LlamaGuard** is only a monitor (Δ) and does *not* come with a recovery policy $\pi^{\texttt{AI}}_{\mathbb{U}}$; its implicit policy is $\pi^{\text{LlamaGuard}}_{\mathbb{U}} = \texttt{stop}$. Since our driving setting is out-of-distribution for the off-the-shelf **LlamaGuard** model, we train an equivalent model in our setting by generating a supervised learning dataset from rollouts of the Llama-3.1-1B-Instruct model. The label $y \in \{0, 1\}$ for the state-action pair $(z^{\texttt{AI}}, a^{\texttt{AI}})$ in physical state s is 0 if $s_+ \sim T(\cdot \mid s, f_{\mathcal{U}}(s, a^{\texttt{AI}})) \in \mathcal{F}$ and is 1 otherwise. The

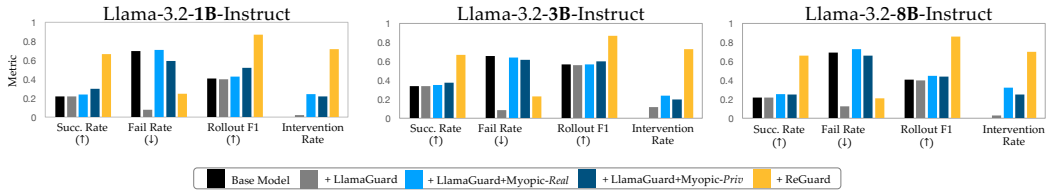

Figure 4: **Agentic Driving: Shielding.** Success rate, failure rate, rollout f1 score, intervention rates.

LlamaGuard model uses the same base model as **ReGuard**—a Llama-3.2-1B-Instruct model fine-tuned with LoRA and an MLP head to output the probability of safety. We also compare to a baseline where **LlamaGuard** is the monitor and **Myopic** from Section 6.1.1 is the recovery policy.

Metrics. **Success Rate** is the fraction of trajectories that safely reach the goal, the **Failure Rate** is the fraction of trajectories that result in failure. **Rollout F1** is the F1 score of successful trajectories compared to the Privileged policy, and the **Intervention Rate** is the fraction of timesteps where the monitor intervened. Note that a high intervention is not necessarily bad, since a strong guardrail will have to intervene frequently when shielding a very weak base policy.

Results. The results are shown in Figure 4. We note that the *same* **ReGuard** guardrail (trained with the 1B model) was deployed to safeguard *all* LLM agent models. Across all base models, **ReGuard** has the highest success rate and highest rollout F1 score. The failure rate is lowest for the **LlamaGuard** model, but this is because the model gets to "cheat" by shutting off the simulation before failure occurs (since $\pi_{\text{◐}}^{\text{LlamaGuard}} = \texttt{stop}$). Conversely, this results in the success rate being no higher than the base models, since **LlamaGuard** only refuses to generate an action when it detects danger rather that suggesting a recovery action. In reality, it is not feasible to stop the trajectory of an embodied system when imminent failure is detected—some fallback policy still needs to be executed.

We also find that the safe success rate does not substantially improve when **LlamaGuard** is paired with either the **Myopic-Realistic** or **Myopic-Privileged** recovery policies. Despite **Myopic-Privileged** reaching nearly as high a success rate as the **Privileged** policy when it is *always* executed (as seen in Section 6.1.1), it is still not useful as a fallback policy when paired with **LlamaGuard**. This is because the **LlamaGuard** monitor is trained myopically—it learns whether the immediate next action will result in failure, but not if the next action may lead to an *inevitably* unsafe state. This means it cannot reliably detect critical states early enough for recovery to be possible. In contrast, since **ReGuard** co-trains both the safety monitor Δ *and* the fallback policy $\pi_{\text{◐}}^{\texttt{AI}}$, the guardrail intervenes at times where the fallback policy has time to correct the base LLM agent away from failures.

6.3 Consequence-Aware Safety Specifications Result in Contextual Recovery Behaviors

Finally, we study an AI assistant scenario where the LLM agent needs to give textual advice on what physical actions the user should take to stay safe. Importantly, the LLM agent's actions only have *indirect* influence on the environment through the human proxy.

Setup. We focus on the **Backseat AI Driver** environment. The user is simulated via a human proxy LLM, which is a fixed Llama-3.1-8B-Instruct model. The AI's advice is a full sentence of text passed on to the human proxy. We focus on the safety guardrail's ability to generate open-vocabulary advice that can influence the human-AI system towards safe outcomes; as such, we provide it with the correct multiple choice physical action to take to stay safe (computed via the **Privileged** baseline from Section 6.1) appended to the prompt. We test two personas for the human proxy: $\pi^{\texttt{H}} \in \{\texttt{none}, \texttt{urgent}\}$, where the `urgent` persona only listens to the AI's advice if it is convinced that the advice is particularly urgent. In the first set of experiments (called **Full Backseat Driving**), we allow **ReGuard** to generate actions at every timestep. In the second set of experiments (called **Shielded Backseat Driving**), we evaluate **ReGuard** as a safety shield, only intervening when the safety monitor's Q-value of the human's intended action $a^{\texttt{H}}$ is less than a small value ϵ. Prompts are in the Appendix.

Baselines. In the **Full Backseat Driving** setting, we compare to the base model that **ReGuard** was fine-tuned on—**Llama-3.2-1B-Instruct**. In the **Shielded Backseat Driving** setting, we compare to just the human proxy acting in isolation.

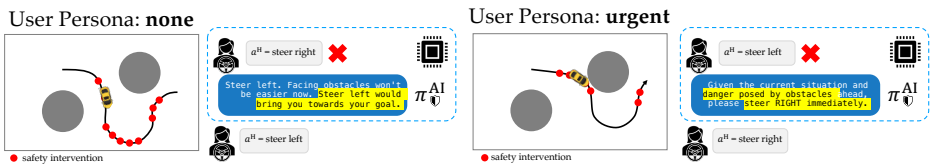

Figure 5: **Backseat Driving: Shielding.** Shielded rollouts of **ReGuard** influencing the `none` and `urgent` human proxy personas to stay safe.

Table 3: **Backseat Driving: Full.** Success rate, failure rate, rollout F1 score

Model	Persona	Success Rate (↑)	Failure Rate (↓)	Rollout F1 (↑)
Llama-3.2-1B-Instruct	`none`	82.4%	16.6%	0.97
ReGuard	`none`	**84.9%**	**15.1%**	**0.99**
Llama-3.2-1B-Instruct	`urgent`	42.9%	56.7%	0.66
ReGuard	`urgent`	**53.3%**	**46.5%**	**0.76**

Table 4: **Backseat Driving: Shielding.** Success rate, failure rate, rollout F1 score, intervention rate

Model	Persona	Succes Rate (↑)	Failure Rate (↓)	Rollout F1 (↑)	Interventions
Llama-3.1-8B-Instruct	`none`	44.9%	54.7%	0.68	N/A
+ReGuard	`none`	**64.2%**	**25.4%**	**0.84**	62.1%
Llama-3.1-8B-Instruct	`urgent`	26.4%	67.8%	0.47	N/A
+ReGuard	`urgent`	**41.6%**	**57.1%**	**0.65**	72.4%

Metrics. For **Full Backseat Driving**, we again measure the **Success Rate**, **Failure Rate** and **Rollout F1 score**. For **Shielded Backseat Driving**, we additionally report the **Intervention Rate**.

Results: Full Backseat. Table 3, shows **ReGuard** has higher success rates, lower failure rates and higher rollout F1 scores than the base model when the human has no persona and also when they have the `urgent` persona. **ReGuard** is able to learn textual recovery policies even when the physical outcomes are only *indirectly* influenced by the AI action $a^{\texttt{AI}}$, via the highly nonlinear human proxy.

Results: Shielded Backseat. Table 4 shows **ReGuard** improves the success rate, failure rate and rollout F1 score without needing to intervene at every timestep. In Figure 5, we see that the safety shield activates close to obstacles and the goal, but the advice generated by $\pi^{\texttt{AI}}_{\shield}$ is adapted to the context. With no persona, **ReGuard** convinces the human to steer left to bring them closer to their goal, whereas it convinces the `urgent` persona human to avoid the obstacle by highlighting the danger of the obstacle and capitalizing its suggested safe action.

7 Conclusion & Limitations

In this work, we propose that generative AI guardrails can be computed as solutions to a specific class of sequential decision problems. We formalize *predictive guardrails* as control-theoretic safety filters operating in learned latent representation spaces. Our scalable training framework based on safety-centric reinforcement learning enables the synthesis of AI agent guardrails that can filter arbitrary base agent models without retraining. In experiments across simulated autonomous driving, e-commerce, and AI assistants, we demonstrate that our control-theoretic guardrails trained on real outcome data reliably detect and correct unsafe LLM agent behaviors, surpassing output-centric flag-and-block guardrail baselines on both safety enforcement and task performance.

Limitations. Our approach assumes access to simulators that provide reliable safety outcome signals during training, which remains an open challenge for many agentic domains. Additionally, our current guardrails are computed for a fixed safety specification; future work should investigate how to generalize guardrails for test-time safety specifications that meet different stakeholder needs.

Acknowledgments and Disclosure of Funding

This work was supported in part by the U.S. National Science Foundation (NSF) Graduate Research Fellowship Program under Grant No. DGE-2444107 and by NSF Awards 2246447 and 2340851. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] S. Zhou, F. F. Xu, H. Zhu, et al. "WebArena: A Realistic Web Environment for Building Autonomous Agents". *International Conference on Learning Representations*. 2024.

[2] J. Y. Koh, R. Lo, L. Jang, et al. "Visualwebarena: Evaluating multimodal agents on realistic visual web tasks". *arXiv preprint arXiv:2401.13649* (2024).

[3] OpenAI. "Buy it in ChatGPT: Instant Checkout and the Agentic Commerce Protocol". https://openai.com/index/buy-it-in-chatgpt/. 2025.

[4] C. E. Jimenez, J. Yang, A. Wettig, et al. "Swe-bench: Can language models resolve real-world github issues?" *arXiv preprint arXiv:2310.06770* (2023).

[5] A. B. Soni, B. Li, X. Wang, et al. "Coding Agents with Multimodal Browsing are Generalist Problem Solvers". *arXiv preprint arXiv:2506.03011* (2025).

[6] B. Li, Y. Wang, J. Mao, et al. "Driving Everywhere with Large Language Model Policy Adaptation". 2024.

[7] Y.-C. Hsu, J. DeCastro, A. Silva, and G. Rosman. "Timing the Message: Language-Based Notifications for Time-Critical Assistive Settings". *arXiv preprint arXiv:2509.07438* (2025).

[8] M. J. Kim, K. Pertsch, S. Karamcheti, et al. "Openvla: An open-source vision-language-action model". *arXiv preprint arXiv:2406.09246* (2024).

[9] Y. Ma, Y. Cao, J. Sun, et al. "Dolphins: Multimodal Language Model for Driving". 2023. arXiv: 2312.00438 [cs.CV].

[10] T. Gemini Robotics, S. Abeyruwan, J. Ainslie, et al. "Gemini robotics: Bringing ai into the physical world". *arXiv preprint arXiv:2503.20020* (2025).

[11] B. Larsen, C. Li, S. Teeuwen, et al. "Navigating the ai frontier: A primer on the evolution and impact of ai agents". Technical report, World Economic Forum. 2024.

[12] Y. Bengio, T. Maharaj, L. Ong, et al. "The singapore consensus on global ai safety research priorities". *arXiv preprint arXiv:2506.20702* (2025).

[13] R. Shah, A. Irpan, A. M. Turner, et al. "An approach to technical agi safety and security". *arXiv preprint arXiv:2504.01849* (2025).

[14] T. Rebedea, R. Dinu, M. Sreedhar, et al. "Nemo guardrails: A toolkit for controllable and safe llm applications with programmable rails". *arXiv preprint arXiv:2310.10501* (2023).

[15] S. G. Ayyamperumal and L. Ge. "Current state of LLM Risks and AI Guardrails". *arXiv preprint arXiv:2406.12934* (2024).

[16] Y. Dong, R. Mu, G. Jin, et al. "Building guardrails for large language models". *ICML* (2024).

[17] H. Inan, K. Upasani, J. Chi, et al. "Llama guard: Llm-based input-output safeguard for human-ai conversations". *arXiv preprint arXiv:2312.06674* (2023).

[18] S. Karnik and S. Bansal. "Preemptive Detection and Steering of LLM Misalignment via Latent Reachability". *arXiv preprint arXiv:2509.21528* (2025).

[19] 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* 7 (2024).

[20] K. P. Wabersich, A. J. Taylor, J. J. Choi, et al. "Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems". *IEEE Control Systems Magazine* 43.5 (2023), pp. 137–177.

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

[22] A. D. Ames, S. Coogan, M. Egerstedt, et al. "Control barrier functions: Theory and applications". *European control conference (ECC)*. Ieee. 2019, pp. 3420–3431.

[23] O. Bastani. “Safe reinforcement learning with nonlinear dynamics via model predictive shielding”. *American control conference (ACC)*. IEEE. 2021, pp. 3488–3494.

[24] M. K. Rad, H. Nghiem, A. Luo, et al. “Refining input guardrails: Enhancing llm-as-a-judge efficiency through chain-of-thought fine-tuning and alignment”. *arXiv preprint arXiv:2501.13080* (2025).

[25] P. Kapoor, A. Ganlath, C. Liu, et al. “Constrained Decoding for Robotics Foundation Models”. *arXiv preprint arXiv:2509.01728* (2025).

[26] G. Alon and M. Kamfonas. “Detecting language model attacks with perplexity”. *arXiv preprint arXiv:2308.14132* (2023).

[27] L. Dixon, J. Li, J. Sorensen, et al. “Measuring and mitigating unintended bias in text classification”. *AAAI/ACM Conference on AI, Ethics, and Society*. 2018, pp. 67–73.

[28] Z. Lin, Z. Wang, Y. Tong, et al. “Toxicchat: Unveiling hidden challenges of toxicity detection in real-world user-ai conversation”. *arXiv preprint arXiv:2310.17389* (2023).

[29] E. Perez, S. Huang, F. Song, et al. “Red teaming language models with language models”. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing,* (2022).

[30] Y. Miyaoka and M. Inoue. “Cbf-llm: Safe control for llm alignment”. *arXiv preprint arXiv:2408.15625* (2024).

[31] H. Hu, A. Robey, and C. Liu. “Steering Dialogue Dynamics for Robustness against Multi-turn Jailbreaking Attacks”. *ICML 2025 Workshop on Reliable and Responsible Foundation Models*. 2025.

[32] L. Kong, H. Wang, W. Mu, et al. “Aligning large language models with representation editing: A control perspective”. *Advances in Neural Information Processing Systems* 37 (2024), pp. 37356–37384.

[33] X. Chen, Y. As, and A. Krause. “Learning Safety Constraints for Large Language Models”. *International Conference on Machine Learning* (2025).

[34] Z. Xiang, L. Zheng, Y. Li, et al. “Guardagent: Safeguard llm agents by a guard agent via knowledge-enabled reasoning”. *arXiv preprint arXiv:2406.09187* (2024).

[35] Z. Chen, M. Kang, and B. Li. “ShieldAgent: Shielding Agents via Verifiable Safety Policy Reasoning”. *Forty-second International Conference on Machine Learning*. 2025.

[36] B. Zheng, Z. Liao, S. Salisbury, et al. “WebGuard: Building a Generalizable Guardrail for Web Agents”. *arXiv preprint arXiv:2507.14293* (2025).

[37] 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* 3.1 (2020), pp. 269–296.

[38] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin. “A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games”. *IEEE Transactions on automatic control* 50.7 (2005), pp. 947–957.

[39] K. Margellos and J. Lygeros. “Hamilton–Jacobi formulation for reach–avoid differential games”. *IEEE Transactions on automatic control* 56.8 (2011), pp. 1849–1861.

[40] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Sastry. “Reach-avoid problems with time-varying dynamics, targets and constraints”. *international conference on hybrid systems: computation and control*. 2015, pp. 11–20.

[41] S. Singh, A. Majumdar, J.-J. Slotine, and M. Pavone. “Robust online motion planning via contraction theory and convex optimization”. *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 5883–5890.

[42] Z. Qin, K. Zhang, Y. Chen, et al. “Learning safe multi-agent control with decentralized neural barrier certificates”. *arXiv preprint arXiv:2101.05436* (2021).

[43] 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* 5.1 (2020), pp. 127–132.

[44] J. Li, Q. Liu, W. Jin, et al. “Robust safe learning and control in an unknown environment: An uncertainty-separated control barrier function approach”. *IEEE Robotics and Automation Letters* 8.10 (2023), pp. 6539–6546.

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

[46] T. Mannucci, E.-J. Van Kampen, C. De Visser, and Q. Chu. "Safe exploration algorithms for reinforcement learning controllers". *IEEE transactions on neural networks and learning systems* 29.4 (2017), pp. 1069–1081.

[47] I. M. Mitchell. "The flexible, extensible and efficient toolbox of level set methods". *Journal of Scientific Computing* 35.2 (2008), pp. 300–329.

[48] C. Liu and M. Tomizuka. "Control in a safe set: Addressing safety in human-robot interactions". *Dynamic Systems and Control Conference*. Vol. 46209. American Society of Mechanical Engineers. 2014, V003T42A003.

[49] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada. "Control barrier function based quadratic programs for safety critical systems". *IEEE Transactions on Automatic Control* 62.8 (2016), pp. 3861–3876.

[50] K. P. Wabersich and M. N. Zeilinger. "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems". *Automatica* 129.C (2021).

[51] J. F. Fisac, N. F. Lugovoy, V. Rubies-Royo, et al. "Bridging Hamilton–Jacobi safety analysis and reinforcement learning". *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 8550–8556.

[52] K.-C. Hsu, V. Rubies-Royo, C. J. Tomlin, and J. F. Fisac. "Safety and Liveness Guarantees through Reach-Avoid Reinforcement Learning". *Robotics: Science and Systems (RSS)*. 2021.

[53] K.-C. Hsu, D. P. Nguyen, and J. F. Fisac. "ISAACS: Iterative soft adversarial actor-critic for safety". *Learning for Dynamics and Control Conference*. PMLR. 2023, pp. 90–103.

[54] S. Bansal and C. Tomlin. "DeepReach: A Deep Learning Approach to High-Dimensional Reachability". *ICRA* (2020).

[55] A. Lin, S. Peng, and S. Bansal. "One filter to deploy them all: Robust safety for quadrupedal navigation in unknown environments". *arXiv preprint arXiv:2412.09989* (2024).

[56] T. He, C. Zhang, W. Xiao, et al. "Agile but safe: Learning collision-free high-speed legged locomotion". *Robotics: Science and Systems* (2024).

[57] K. Nakamura, L. Peters, and A. Bajcsy. "Generalizing safety beyond collision-avoidance via latent-space reachability analysis". *Robotics: Science and Systems* (2025).

[58] J. Seo, K. Nakamura, and A. Bajcsy. "Uncertainty-aware Latent Safety Filters for Avoiding Out-of-Distribution Failures". *Conference on Robot Learning* (2025).

[59] L. Ouyang, J. Wu, X. Jiang, et al. "Training language models to follow instructions with human feedback". *Advances in neural information processing systems* 35 (2022), pp. 27730–27744.

[60] Y. Bai, A. Jones, K. Ndousse, et al. "Training a helpful and harmless assistant with reinforcement learning from human feedback". *arXiv preprint arXiv:2204.05862* (2022).

[61] Y. Zhou, A. Zanette, J. Pan, et al. "Archer: Training language model agents via hierarchical multi-turn rl". *arXiv preprint arXiv:2402.19446* (2024).

[62] Z. Qi, X. Liu, I. L. Iong, et al. "Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning". *arXiv preprint arXiv:2411.02337* (2024).

[63] Z. Xi, Y. Ding, W. Chen, et al. "Agentgym: Evolving large language model-based agents across diverse environments". *arXiv preprint arXiv:2406.04151* (2024).

[64] Z. Xi, J. Huang, C. Liao, et al. "Agentgym-rl: Training llm agents for long-horizon decision making through multi-turn reinforcement learning". *arXiv preprint arXiv:2509.08755* (2025).

[65] Z. Wang, K. Wang, Q. Wang, et al. "Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning". *arXiv preprint arXiv:2504.20073* (2025).

[66] R. Isaacs. "Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization". Revised edition. Mineola, N.Y: Dover Publications, 1965.

[67] E. Barron and H. Ishii. "The Bellman equation for minimizing the maximum cost." *NONLINEAR ANAL. THEORY METHODS APPLIC.* 13.9 (1989), pp. 1067–1090.

[68] C. Tomlin, J. Lygeros, and S. Shankar Sastry. "A Game Theoretic Approach to Controller Design for Hybrid Systems". *Proceedings of the IEEE* 88.7 (2000), pp. 949–970.

[69] J. Lygeros. "On reachability and minimum cost optimal control". *Automatica* 40.6 (2004), pp. 917–927.

[70] H. Van Hasselt, A. Guez, and D. Silver. "Deep reinforcement learning with double q-learning". *AAAI conference on artificial intelligence*. Vol. 30. 2016.

[71] A. Dubey, A. Jauhri, A. Pandey, et al. "The llama 3 herd of models". *arXiv e-prints* (2024), arXiv–2407.

[72] E. J. Hu, Y. Shen, P. Wallis, et al. "Lora: Low-rank adaptation of large language models. arXiv 2021". *arXiv preprint arXiv:2106.09685* 10 (2021).

[73] A. Z. Ren, A. Dixit, A. Bodrova, et al. "Robots That Ask For Help: Uncertainty Alignment for Large Language Model Planners". *Conference on Robot Learning*. PMLR. 2023, pp. 661–682.

[74] X. Sun, Y. Zhang, X. Tang, et al. "Trustnavgpt: Modeling uncertainty to improve trustworthiness of audio-guided llm-based robot navigation". *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2024, pp. 8794–8801.

[75] L. Zheng, W.-L. Chiang, Y. Sheng, et al. "Judging llm-as-a-judge with mt-bench and chatbot arena". *Advances in neural information processing systems* 36 (2023), pp. 46595–46623.

[76] L. von Werra, Y. Belkada, L. Tunstall, et al. "TRL: Transformer Reinforcement Learning". `https://github.com/huggingface/trl`. 2020.

[77] OpenAI. "GPT-4o System Card". 2024.