Generalizing Safety Beyond Collision-Avoidance via Latent-Space Reachability Analysis

Kensuke Nakamura
Carnegie Mellon University

Lasse Peters
Delft University of Technology

Andrea Bajcsy
Carnegie Mellon University

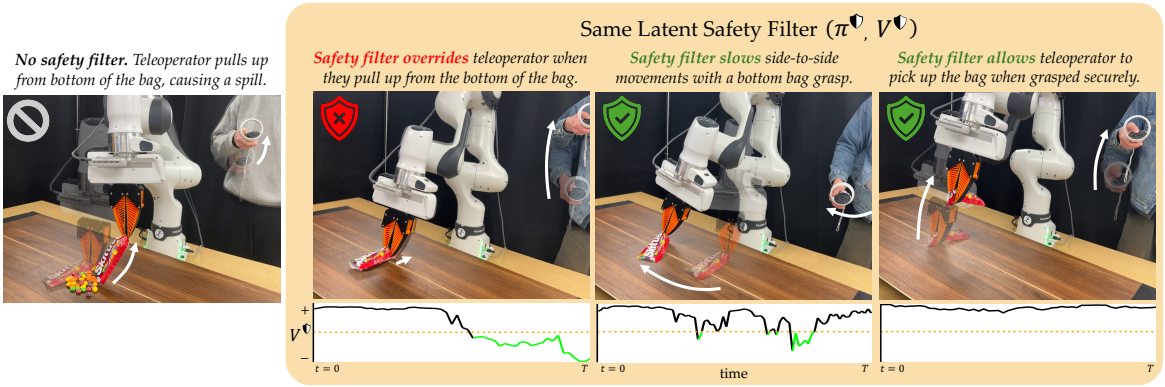

Fig. 1: *Far Left*: Without a safety filter, a teleoperator lifts the closed end of the bag too quickly and spills the Skittles. Our Latent Safety Filter overrides unsafe actions (*Middle Left*) and allows safe actions that do not violate constraints (*Right*).

***Abstract*—Hamilton-Jacobi (HJ) reachability is a rigorous mathematical framework that enables robots to simultaneously detect unsafe states and generate actions that prevent future failures. While in theory, HJ reachability can synthesize safe controllers for nonlinear systems and nonconvex constraints, in practice, it has been limited to hand-engineered collision-avoidance constraints modeled via low-dimensional state-space representations and first-principles dynamics. In this work, our goal is to *generalize* safe robot controllers to prevent failures that are hard—if not impossible—to write down by hand, but can be intuitively identified from high-dimensional observations: for example, spilling the contents of a bag. We propose *Latent Safety Filters*, a latent-space generalization of HJ reachability that tractably operates directly on raw observation data (e.g., RGB images) to automatically computes safety-preserving actions without explicit recovery demonstrations by performing safety analysis in the latent embedding space of a generative world model. Our method leverages diverse robot observation-action data of varying quality to learn a world model. Constraint specification is then transformed into a classification problem in the latent space of the learned world model. In hardware experiments, we use Latent Safety Filters to safeguard arbitrary policies (from imitation-learned policies to direct teleoperation) from complex safety hazards, like preventing a Franka Research 3 manipulator from spilling the contents of a bag.**

I. Introduction

Imagine that a robot manipulator is deployed in your home, like shown in Figure 1. What safety constraints should the robot reason about? It is common to equate robot safety with "collision avoidance", but in unstructured open world environments, a robot's representation of safety should be much more nuanced. For example, the household manipulator should understand that pouring coffee too fast will cause the liquid to overflow; pulling a mug too quickly from a cupboard will cause other dishes to fall; or, in Figure 1, aggressively pulling up from the bottom of an open bag will cause the contents to spill. Safe control frameworks, such as Hamilton-Jacobi (HJ) reachability analysis [13, 15] mathematically model safety constraints as arbitrary sets in a state space and automatically identify states that will inevitably lead to robot failures by solving an optimal control problem. However, the question remains how to practically instantiate this theoretical framework to safeguard against more nuanced failures—*beyond collision-avoidance*—in robotics.

Our key insight is that the latent representations learned by generative world models [7, 20] enable safe control for constraints not traditionally expressable in handcrafted state-space representations. While world models require diverse coverage (success, play, and/or failure data) to accurately predict the dynamical consequences of robot actions, this formulation makes constraint specification as easy as learning a classifier in the latent space [18], and HJ reachability can safely evaluate possible outcomes of different actions within the "imagination" of the world model without additional unsafe environment interactions.

We evaluate our approach in vision-based safe-control tasks using hardware experiments with a Franka Research 3 arm, which picks up an open bag of Skittles without spilling. Our quantitative results show that, without assuming access to ground-truth dynamics or hand-designed failure specifications, *Latent Safety Filters* can safeguard an unsafe imitation learned policy [2] to reduce safety violations by 63.6% while allowing performant policies to operate freely. In qualitative experiments, we also find that *Latent Safety Filters* enable safe teleoperation and can generalize to out-of-distribution Skittles

bag colors and background changes.

II. A Brief Background on HJ Reachability

Traditionally, reachability assumes access to a privileged state space $s \in \mathcal{S}$ and a corresponding bounde nonlinear dynamics model $s_{t+1} = f(s_t, a_t)$. A domain expert will first specify what safety means in this state space by imposing a constraint, referred to as the *failure set*, $\mathcal{F} \subset \mathcal{S}$. Given the failure set, HJ reachability will automatically compute two entities: (i) a safety monitor, $V : \mathcal{S} \to \mathbb{R}$, which quantifies if the robot is doomed to enter $\mathcal{F}$ from its current state s despite the robot's best efforts, and (ii) a best-effort safety-preserving policy, $\pi^{\text{\textbf{O}}} : \mathcal{S} \to \mathcal{A}$. These two entities are co-optimized via the solution to an optimal control problem that satisfies the fixed-point safety Bellman equation [4]:

$$V(s) = \min\Big\{\ell(s),\ \max_{a\in\mathcal{A}} V(f(s,a))\Big\}, \tag{1}$$

where $\ell : \mathcal{S} \to \mathbb{R}$ is a bounded margin function that encodes the safety constraint $\mathcal{F}$ via its zero-sublevel set $\mathcal{F} = \{s \mid \ell(s) < 0\}$, typically modeled as a signed-distance function. The maximally safety-preserving policy can be obtained via $\pi^{\text{\textbf{O}}}(s) := \arg\max_{a\in\mathcal{A}} V(f(s,a))$. Finally, the *unsafe set*, $\mathcal{U} \subset \mathcal{S}$, which models the set of states from which the robot is doomed to enter $\mathcal{F}$, can be recovered from the zero-sublevel set of the value function: $\mathcal{U} := \{s : V(s) < 0\}$.

At deployment time, the safety monitor and safety policy can be utilized together to perform *safety filtering*: detecting an unsafe action generated by any base policy, π^{task}, and minimally modifying it to ensure safety. While there are a myriad of safety filtering schemes (see surveys [9, 17] for details), a common minimally-invasive approach switches between the nominal and the safety policy when the robot is on the verge of being doomed to fail: $a^{\text{exec}} = \mathbb{1}_{\{V(s)>0\}} \cdot \pi^{\text{task}} + \mathbb{1}_{\{V(s)\le 0\}} \cdot \pi^{\text{\textbf{O}}}$.

III. Latent Safety Filters

To tackle both detecting and mitigating hard-to-model failures, we present a latent-space generalization of HJ reachability (from Section II) that tractably operates on raw observation data (e.g., RGB images) by performing safety analysis in the latent embedding space of a generative world model. This also transforms nuanced constraint specification into a classification problem in latent space and enables reasoning about dynamical consequences that are hard to simulate.

Setup: Environment and Latent World Models. We model the robot as operating in an environment $\mathbf{E} \in \mathbb{E}$ that is partially observable. While we never have direct access to the true state, we leverage a world model that jointly infers a lower-dimensional latent state which correspond to high-dimensional observations (e.g., RGB images) and associated dynamics,

A world model consists of an encoder that maps observations o_t (e.g., images, proprioception, etc.) and latent state $\hat{z}_t$ into a posterior latent z_t, and a transition function that predicts the future latent state conditioned on an action. This can be mathematically described as:

$$\text{Encoder: } z_t \sim \mathcal{E}_\psi(z_t \mid \hat{z}_t, o_t)$$

$$\text{Transition Model: } \hat{z}_{t+1} \sim p_\phi(\hat{z}_{t+1} \mid z_t, a_t).$$

This formulation describes a wide range of world models [5, 6, 7, 8, 20], and our latent safety filter is not tied to a particular world model architecture. We focus on world models that are trained via self-supervised learning (observation reconstruction, teacher forcing, etc.) and do not require access to a privileged state. Specifically, we use DINO-WM [20], which is trained via teacher-forcing.

Safety Specification: Failure Classifier on Latent State. A common approach for representing $\mathcal{F}$ is to encode it as the zero-sublevel set of a function $\ell(s)$ (as in Eq. 1). This "margin function" is typically a signed distance function to the failure set, which easily expresses constraints like collision-avoidance. However, other types of constraints, such as spills, are much more difficult to directly express with this class of functions and traditional state spaces. We instead chose to learn $\ell_\mu(z)$ from data by modeling it as a classifier over latent states $z \in \mathcal{Z}$, with learnable parameters μ.

We train our classifier on labelled datasets of observations corresponding to safe and unsafe states, $o^+ \in \mathcal{D}_{\text{safe}}$ and $o^- \in \mathcal{D}_{\text{unsafe}}$, and optimize a loss function inspired by [19]:

$$\begin{aligned}\mathcal{L}(\mu) =& \frac{1}{N_{\text{safe}}} \sum_{o^+\in\mathcal{D}_{\text{safe}}} \text{ReLU}\big(\delta - \ell_\mu(\mathcal{E}_\psi(o^+))\big) \\ &+ \frac{1}{N_{\text{fail}}} \sum_{o^-\in\mathcal{D}_{\text{fail}}} \text{ReLU}\big(\delta + \ell_\mu(\mathcal{E}_\psi(o^-))\big),\end{aligned} \tag{2}$$

where the loss function is parameterized by $\delta \in \mathbb{R}^+$ to prevent degenerate solutions where all latent states are labeled as zero by the classifier. Intuitively, this loss penalizes latent states corresponding to observations in the failure set from being labeled positive and vice versa. The learned classifier represents the failure set $\mathcal{F}_{\text{latent}}$ in the latent space of the world model via: $\mathcal{F}_{\text{latent}} = \{z \mid \ell_\mu(z) < 0\}$. Our failure classifier can be co-trained or trained after world model learning.

Latent-Space Reachability in Imagination. Traditionally, reachability analysis requires either an analytic model of the robot and environment dynamics [1, 14] or a high-fidelity simulator [4, 10] to solve the fixed-point Bellman equation, both of which are currently inadequate for complex system dynamics underlying nuanced safety problems (e.g., liquid interaction). Instead, we propose using the latent imagination of a pretrained world model as our environment model, capturing hard-to-design and hard-to-simulate interaction dynamics. We introduce the latent fixed-point Bellman equation:

$$V_{\text{latent}}(z) = \min\Big\{\ell_\mu(z),\ \max_{a\in\mathcal{A}} \mathbb{E}_{\hat{z}'\sim p_\phi(\cdot \mid z,a)}\Big[V_{\text{latent}}(\hat{z}')\Big]\Big\}. \tag{3}$$

Note that in contrast to Equation 1, this backup operates on the latent state z and, for full generality, includes an expectation over transitions to account for world models with stochastic transitions (e.g., RSSMs). For world models with deterministic transitions (e.g., DINO-WM), the expectation can be removed.

While the world model allows us to compress high-dimensional observations into a compact informative latent

state, computing an exact solution to the latent reachability problem is still intractable due to the dimensionality of the latent embedding. This motivates the use of a learning-based approximation to the value function in Equation 3. We follow [4] and induce a contraction mapping for the Bellman backup by adding a time discounting factor $\gamma \in [0, 1)$:

$$V_{\text{latent}}(z) = (1-\gamma)\ell_\mu(z) + \gamma \min\left\{\ell_\mu(z), \max_{a\in\mathcal{A}} \mathbb{E}_{\hat{z}'\sim p_\phi(\cdot \mid z,a)}[V_{\text{latent}}(\hat{z}')]\right\} \quad (4)$$

In theory, if solved to optimality, this latent value function would offer a safety assurance only with respect to the data used to train the world model and the failure classifier. Intuitively, this implies that the robot can only provide an assurance that it will try its hardest to avoid failure *in its representation of the world*. In the following section, we study our overall latent safety framework in a high-dimensional manipulation example on hardware.

IV. Hardware Results: Preventing Hard-to-Model Robot Failures

We design a set of experiments in hardware to see if our Latent Safety Filter can be applied in the real world (shown in Figure 1). We use a Franka Research 3 manipulator equipped with a 3D printed gripper from [3]. The robot is tasked with interacting with an opened bag of Skittles on the table. The safety constraint is not to spill any Skittles. We test the efficacy of our approach by deploying the *same* Latent Safety Filter to safeguard a human teleoperator (Section IV-A) and a strong and weak Diffusion Policy [2] from spilling (Section IV-B), as well as stress-testing our safety filter to out-of-distribution candy bags and environment backgrounds (Section IV-C).

Safety Specification. Our safety specification is to prevent the contents of the Skittles bag from falling out of the bag. Given only image observations and proprioception, this problem is clearly partially observed since the robot cannot directly recover the position of the Skittles in the bag. Even if privileged state information were available, designing a function to characterize failure states or a dynamics model for interactions between all relevant objects would be difficult.

Latent Safety Filter Setup. We use DINO-WM [20], a Vision Transformer-based world model that uses DinoV2 as an encoder [16]. The manipulator uses a 3rd person camera and a wrist-mounted camera and records $3 \times 256 \times 256$ RGB images at 15 Hz. For world model training, we collected a dataset $\mathcal{D}_{\text{WM}}$ of 1,300 offline trajectories: 1,000 of the trajectories are generated sampling random actions drawn from a Gaussian distribution at each time step, 150 trajectories are demonstrations where the bag is grasped without spilling any Skittles, and 150 demonstrations pick up the bag while spilling candy on the table. We manually labeled the observations in the trajectory dataset for apparent failures.

Our world model is trained by first preprocessing and encoding the two camera view using DINOv2 to obtain a set of dense patch tokens for each image. We use the DINOv2 ViT-S, the smallest DINOv2 model with 14M parameters, resulting in latent states z of size 256×384 corresponding to 256 image patches each with embedding dimension 384. The transition function is implemented as a vision transformer, which takes as input the past $H = 3$ patch tokens, proprioception, and actions to predict the latent. The transformer employs frame-level causal attention to ensure that predictions can only depend on previous observations. The model is trained via teacher-forcing minimizing mean-squared error between the ground-truth DINO embeddings of observations and proprioception information from $\mathcal{D}_{\text{WM}}$ and the embeddings and proprioception predicted by the model. After world model training, we separately train the failure classifier (implemented as a 2-layer MLP with a hidden dimension of 788 and ReLU activations) on the DINO patch tokens corresponding to the manually labeled constraint-violating observations. For approximating the HJ value function, we use DDPG [11, 12].

Setup. The safety filter operates according to the following control law:

$$a_t^{\text{exec}} = \begin{cases} \pi^{\text{task}}(z_t), & \text{if } V(\hat{z}_{t+1}) > \epsilon \\ \pi_{\text{latent}}^{\text{safe}}(z_t), & \text{otherwise} \end{cases} \quad (5)$$

where $\hat{z}_{t+1} \sim p_\phi(\hat{z}_{t+1} \mid z_t, \pi^{\text{task}}(z_t))$ is a one-step rollout of the world model using the action proposed by an unshielded teleoperator or task policy. In hardware experiments, we set $\epsilon = 0.3$. The system was controlled at 15 Hz.

A. Shielding Human Teleoperators

To emphasize the policy-agnostic nature of our latent safety filters, we demonstrate filtering a teleoperator in Figure 1.

Results: Shielding Unsafe Grasps and Dynamic Motions. We visualize our qualitative results in Figure 1. Un-shielded by our safety filter, the teleoperator can grab the opened bag of Skittles by the base and pull up sharply, spilling its contents on the table (left, Figure 1). By using **LatentSafe**, the same behavior gets automatically overridden by the safety filter, preventing the teleoperators "pull up" motion from being executed and keeping the Skittles inside (center, Figure 1). At the same time, the latent safety filter is not overly pessimistic (right-most images in Figure 1). When the teleoperator moves the Skittles bag side-to-side while grasping the *bottom* of the opened bag, the safety filter accurately accounts for these dynamics and minimally modifies the teleoperator to slow them down, preventing any Skittles from falling out while still allowing the general motion to be executed. When the teleoperator chooses a safe grasp—grabbing the bag by the top, open side—the safety filter does not activate and allows the person to complete the task safely and autonomously.

B. Shielding Autonomous Imitation-Learned Policies

Next we study how well the same Latent Safety Filter from Section IV-A can shield autonomous imitation-learned (IL) policies. Specifically, we test whether the latent safety filter does *not* impede a strong IL policy (i.e., our filter is not overly conservative) and *improves* the safety of a suboptimal

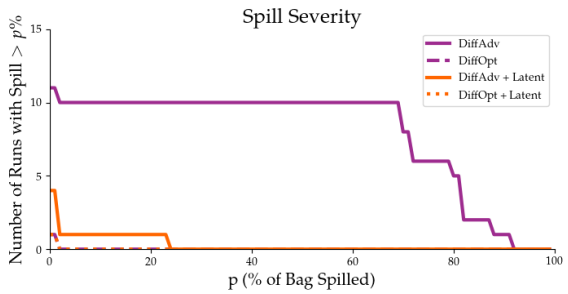

Fig. 2: **Shielding IL Policies.** Percent of bag spilled (p) vs number of runs that spilled at least $p\%$ of the bag.

IL policy (i.e., our filter shields effectively), while removing teleoperator bias that may be present in our prior experiments.

Methods. For our base task policy, $\pi^{\text{task}}(o)$, we use a generative imitation-learned (IL) policy trained with a diffusion objective [2] and which takes as input RGB images and end effector pose as observations $o \in \mathcal{O}$. We train two diffusion policies—**DiffusionAdv** and **DiffusionOpt**—which represent relevant extremes of a base policy's capabilities. **DiffusionOpt** represents the "upper bound" of a strong base policy that uses carefully curated demos of the task. We use this baseline to study whether our safety filter is not overly conservative when shielding a strong base policy. We train it with 100 safe teleoperated demonstrations. We also train **DiffusionAdv**, which represents a "lower bound" of a base policy trained with demonstrations that could lead to unsafe outcomes. This policy is trained with 100 potentially unsafe teleoperated demonstrations. This results in a base policy that has an incomplete understanding of how to interact safely with the Skittles bag, allowing us to test our safety filter's ability to prevent failures in a controlled and repeatable manner.

Metrics. We compare the performance of the $\pi^{\text{task}} \in \{$**DiffusionAdv**, **DiffusionOpt**$\}$ with and without using **LatentSafe** (yielding four methods in total). We use exactly the same *Latent Safety Filter* as we used to shield the human teleoperator in Section IV-A. For each method, we record 15 rollouts where the policy successfully grasped the bag (ignoring missed grasps) 15 times in hardware. We measure the frequency of constraint violations (if even one Skittle falls out during an episode) and spill severity (percentage of the Skittles spilled) in each of these trajectories.

Results: Shielding IL Policies. We report in Figure 2 how often each method spilled more than $p\%$ of the bag. While **DiffusionAdv** frequently spills a large percentage of the bag ($\sim 85\%$), **DiffusionAdv** + **LatentSafe** spills less than 5% of the bag in all but one of the constraint-violating rollouts. **DiffusionOpt** with and without **LatentSafe** spills only 1 skittle in across all 15 rollouts. Overall, **LatentSafe** reduces both the failure rate by 63.6% and failure severity for the base **DiffusionAdv** policy that can cause difficult-to-model failures. For the safe and performant **DiffusionOpt**, our safety filter enables a strong base policy to operate without unnecessary and over-conservative overrides. We also note that practically, since the same Latent Safety Filter was used for both the weak and the strong base IL policy, this provides a promising avenue for safely improving a base task-driven policy without the need to also change the safety representation and fallback controller

C. Testing Out-of-Distribution Generalization of Latent Safety

Finally, we stress-test the performance of the same Latent Safety Filter on out-of-distribution (OOD) bag colors and background changes by replaying a known unsafe demonstration in open-loop as our task "policy". We reset the bag to the same initial condition and shield this replayed demonstration with **LatentSafe** filter for all OOD conditions. Results are shown in Figures 3 and 4.

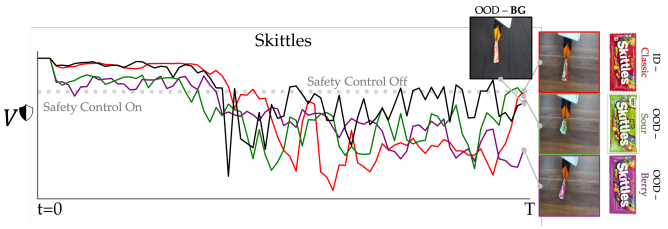

Fig. 3: **OOD Generalization: Skittles.** Our latent safety filter is trained only on a red Skittles bag. It is deployed to shield an open-loop known unsafe trajectory for two OOD skittles bag colors and an OOD background. **LatentSafe** generalizes—maintaining the same performance of preventing spills—to OOD Skittles bag colors and OOD background change.

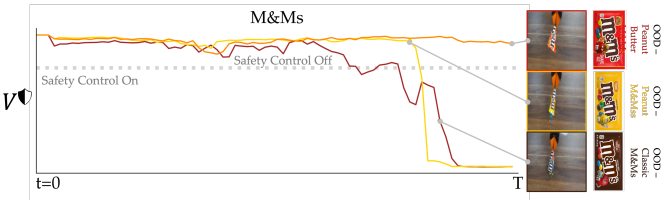

Fig. 4: **OOD Generalization: M&Ms.** Our latent safety filter is trained only on a red Skittles bag. It is deployed to shield an open-loop known unsafe trajectory. **LatentSafe** is deployed with 3 M&M bags with different colors and dynamics. Our filter does *not* prevent the manipulator from lifting these OOD bags. For the brown bag, even though the filter begins to override the recorded trajectory, it does not manage to prevent the spill, potentially due to differing dynamics.

V. Conclusion

In this work, our goal was to generalize robot safety beyond collision-avoidance, accounting for hard-to-model failures like spills, items breaking, or items toppling. We introduced *Latent Safety Filters*, a generalization of the safety filtering paradigm that operates in the learned representation of a generative world model. We instantiated our method on hardware, demonstrating that our latent reachability formulation protects against extremely hard-to-specify failures, such as spills, in the real world for both generative IL policies and human teleoperation.

References

[1] Somil Bansal and Claire J Tomlin. Deepreach: A deep learning approach to high-dimensional reachability. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1817–1824. IEEE, 2021.

[2] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2024.

[3] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[4] Jaime F Fisac, Neil F Lugovoy, Vicenç Rubies-Royo, Shromona Ghosh, and Claire J Tomlin. Bridging hamilton-jacobi safety analysis and reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8550–8556. IEEE, 2019.

[5] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31, 2018.

[6] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019.

[7] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *International Conference on Learning Representations*, 2021.

[8] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models, 2024. URL https://arxiv.org/abs/2301.04104.

[9] Kai-Chieh Hsu, Haimin Hu, and Jaime F Fisac. The safety filter: A unified view of safety-critical control in autonomous systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 7, 2023.

[10] Kai-Chieh Hsu, Duy Phuong Nguyen, and Jaime Fernàndez Fisac. Isaacs: Iterative soft adversarial actor-critic for safety. In *Learning for Dynamics and Control Conference*, pages 90–103. PMLR, 2023.

[11] Jingqi Li, Donggun Lee, Jaewon Lee, Kris Shengjun Dong, Somayeh Sojoudi, and Claire Tomlin. Certifiable deep learning for reachability using a new lipschitz continuous value function, 2025. URL https://arxiv.org/abs/2408.07866.

[12] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning, 2019. URL https://arxiv.org/abs/1509.02971.

[13] John Lygeros. On reachability and minimum cost optimal control. *Automatica*, 40(6):917–927, 2004.

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

[15] Ian Mitchell, Alex Bayen, and Claire J. Tomlin. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control (TAC)*, 50(7):947–957, 2005.

[16] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[17] Kim P Wabersich, Andrew J Taylor, Jason J Choi, Koushil Sreenath, Claire J Tomlin, Aaron D Ames, and Melanie N Zeilinger. Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems. *IEEE Control Systems Magazine*, 43(5):137–177, 2023.

[18] Albert Wilcox, Ashwin Balakrishna, Brijen Thananjeyan, Joseph E Gonzalez, and Ken Goldberg. Ls3: Latent space safe sets for long-horizon visuomotor control of sparse reward iterative tasks. In *Conference on Robot Learning*, pages 959–969. PMLR, 2022.

[19] H Yu, C Hirayama, C Yu, S Herbert, and S Gao. Sequential neural barriers for scalable dynamic obstacle avoidance. in 2023 ieee. In *RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11241–11248.

[20] Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.