

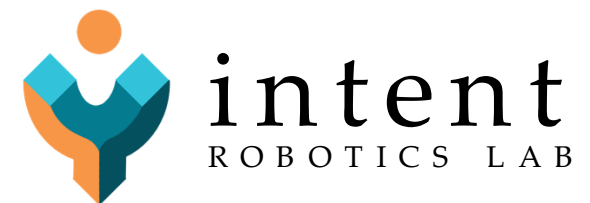
*Some slides courtesy of Somil Bansal*

Lecture 7: Embodied AI Safety (16-886)

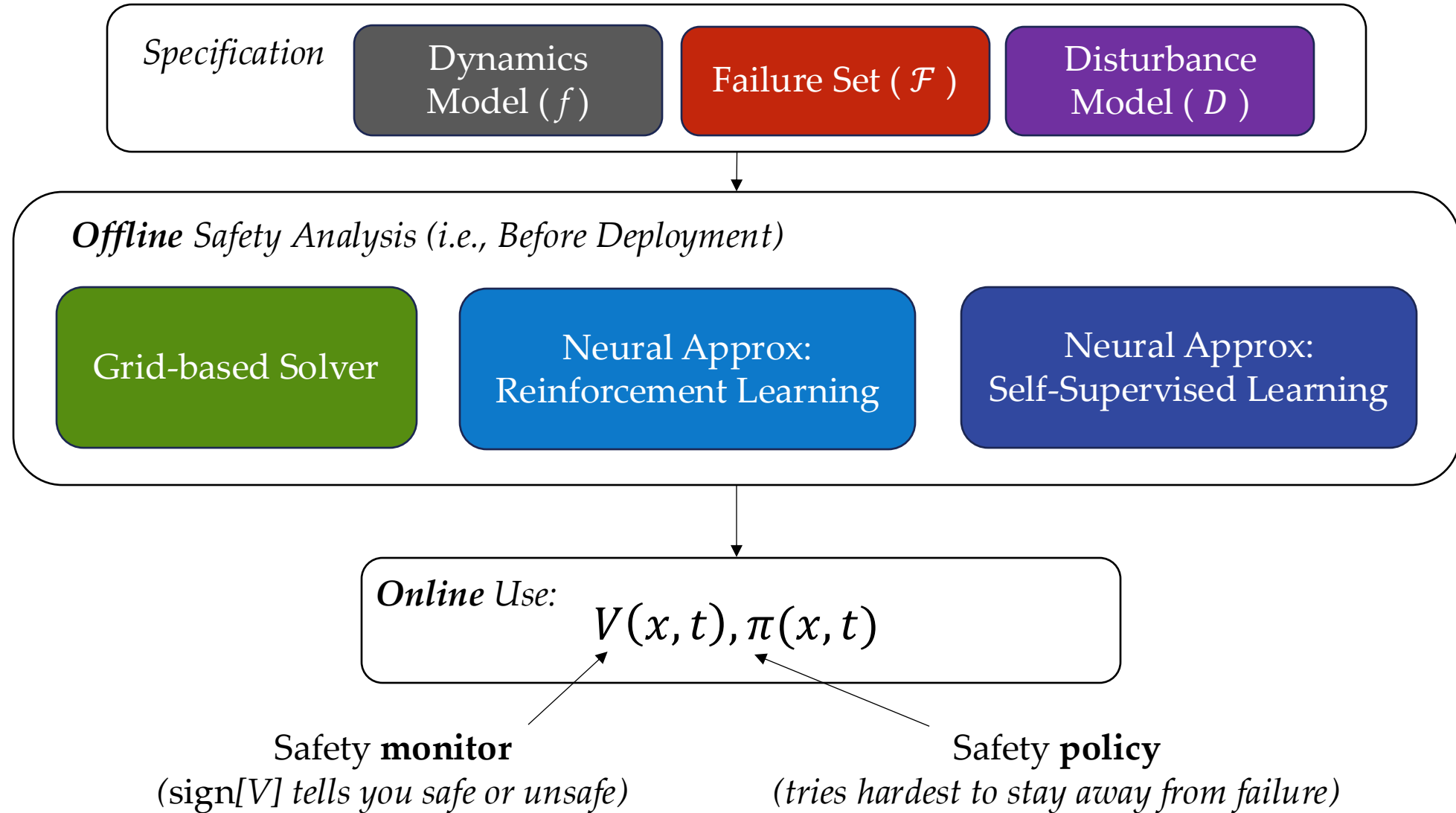
# Updating Safety Online

Instructor: Andrea Bajcsy

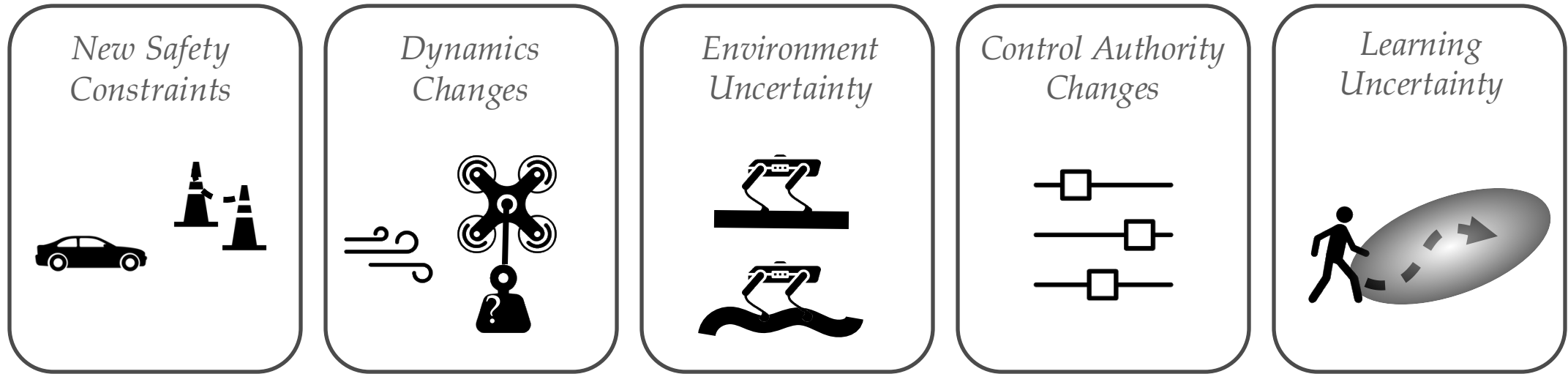
**Carnegie  
Mellon  
University**



# So far, have studied **offline safety (pre-)computations**

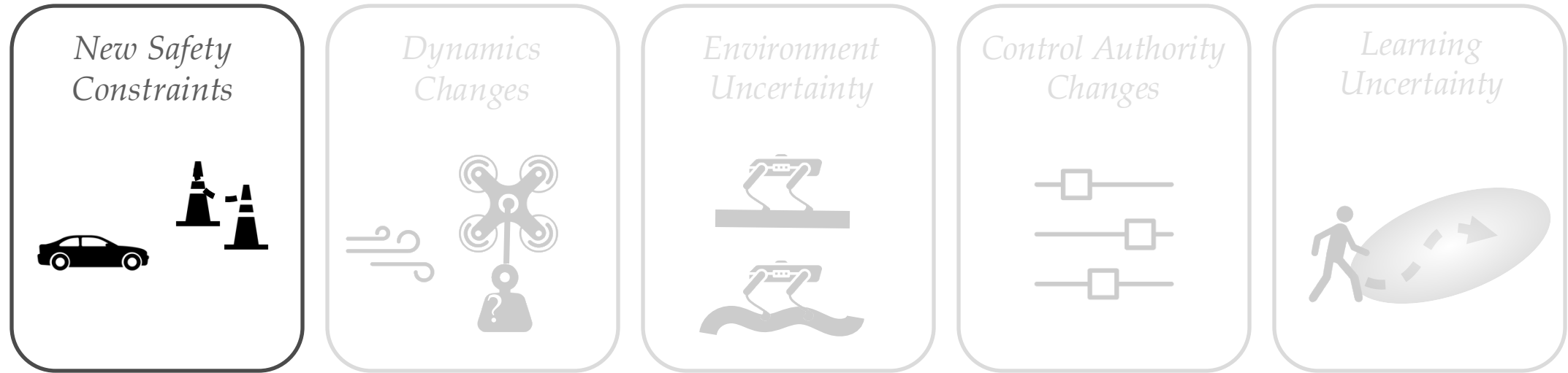


But at **deployment time** the robot may experience new situations



Requires adaptation of reachable sets & safety controller online!

But at **deployment time** the robot may experience new situations



Requires adaptation of reachable sets & safety controller online!

# An Efficient Reachability-Based Framework for Provably Safe Autonomous Navigation in Unknown Environments

Andrea Bajcsy\*, Somil Bansal\*, Eli Bronstein, Varun Tolani, Claire J. Tomlin

**Abstract**—Real-world autonomous vehicles often operate in *a priori* unknown environments. Since most of these systems are safety-critical, it is important to ensure they operate safely in the face of environment uncertainty, such as unseen obstacles. Current safety analysis tools enable autonomous systems to reason about safety given full information about the state of the environment *a priori*. However, these tools do not scale well to scenarios where the environment is being sensed in real time, such as during navigation tasks. In this work, we propose a novel, real-time safety analysis method based on Hamilton-Jacobi reachability that provides strong safety guarantees despite environment uncertainty. Our safety method is planner-agnostic and provides guarantees for a variety of mapping sensors. We demonstrate our approach in simulation and in hardware to provide safety guarantees around a state-of-the-art vision-based, learning-based planner. Videos of our approach and experiments are available on the project website<sup>1</sup>.

## I. INTRODUCTION

Autonomous vehicles operating in the real world must navigate through *a priori* unknown environments using on-board, limited-range sensors. As a vehicle makes progress towards a goal and receives new sensor information about the environment, rigorous safety analysis is critical to ensure that the system’s behavior does not lead to dangerous collisions. In order to provide such safety guarantees for real vehicles, any analysis should take into account multiple sources of uncertainty, such as modelling error, external disturbances, and unknown parts of the environment.

A variety of mechanisms have been proposed to ensure robustness to modeling error and external disturbances [24], [16], [34]. Additionally, safety guarantees for systems using limited range sensors in unknown environments have been

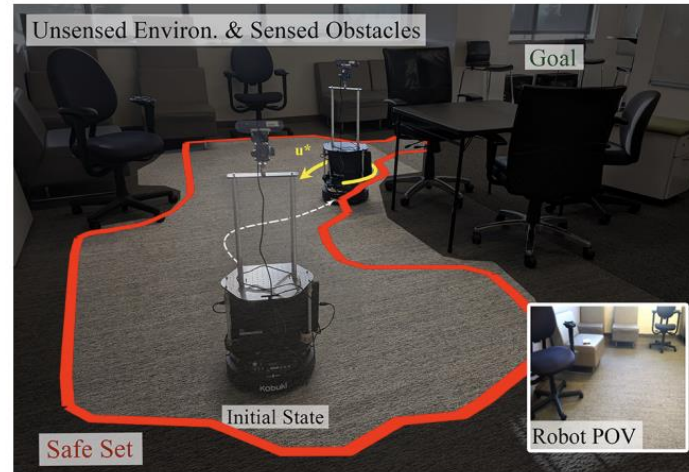


Fig. 1. **Overview:** We consider the problem of safe navigation from an initial state to a goal state in an *a priori* unknown environment. Our approach treats the unsensed environment as an obstacle, and uses a HJ reachability framework to compute a safe controller for the vehicle, which is updated in real-time as the vehicle explores the environment. We show an application of our approach on a Turtlebot using a vision-based planner. When the robot is at risk of colliding, the safe controller ( $u^*$ ) keep the system safe.

external disturbances while minimally interfering with goal-driven behavior. Second, real-time safety assurances need to be provided as new environment information is acquired, which requires approximations that are both computationally efficient and not overly conservative. Moreover, this safety analysis should be applicable to a wide variety of real-world sensors, planners, and vehicles.

In this paper, we propose a safety framework that can overcome these challenges for autonomous vehicles operating in



Third-Person POV



Top-Down



Robot POV



FOV

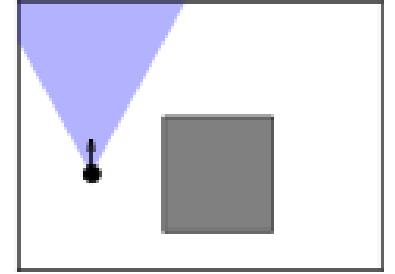
# Assumptions

1. Static environments\*
2. Occupancy perception is perfect within FOV

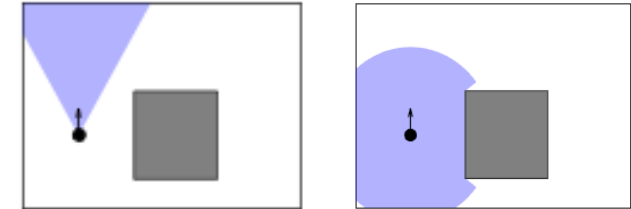
*\* For theoretical guarantees*

# Safety Challenges in Unknown Environments

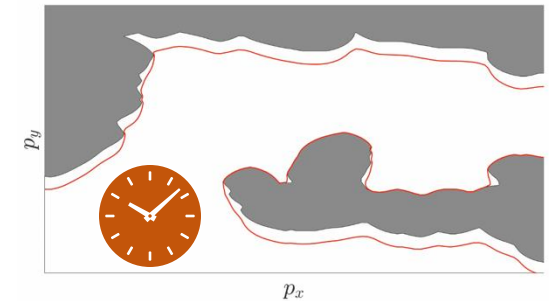
Computing a safe set despite unseen obstacles



Computing a safe set for arbitrary environment exposures

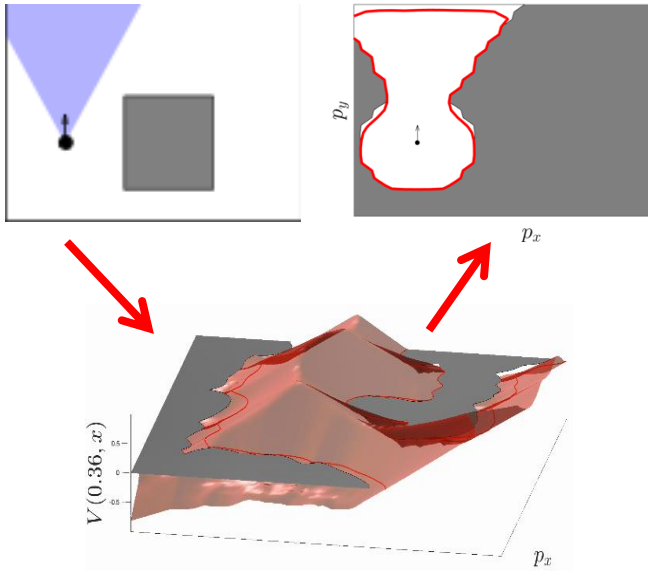


Quickly updating the safe set based on new observations

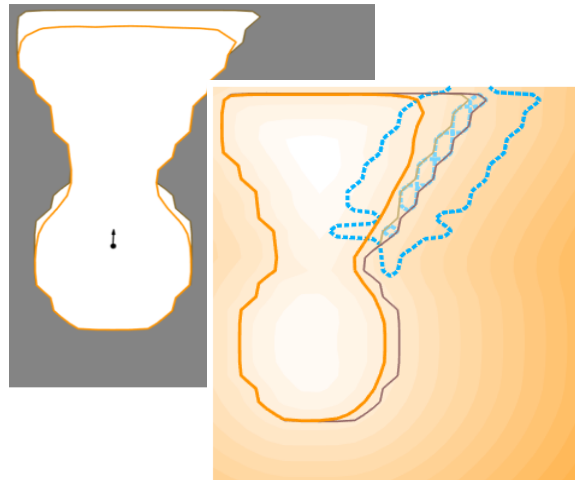




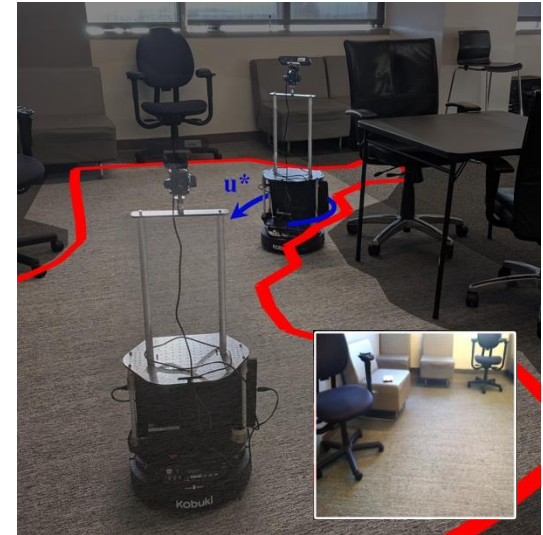
## Setup & Warm Starting



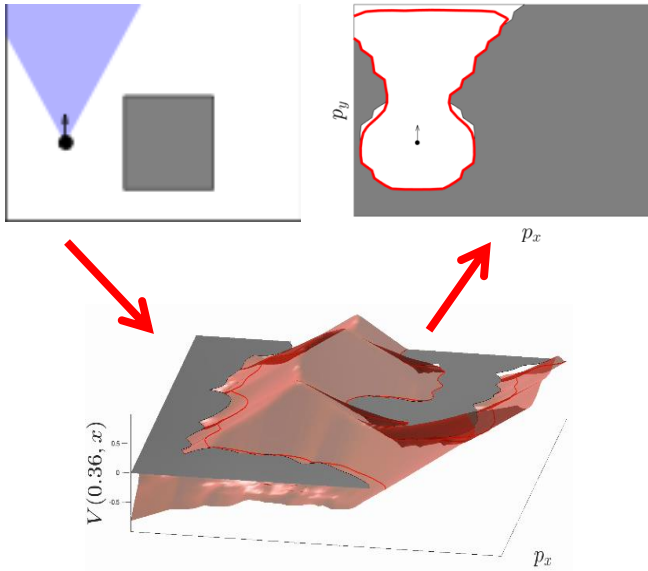
## Local Updates



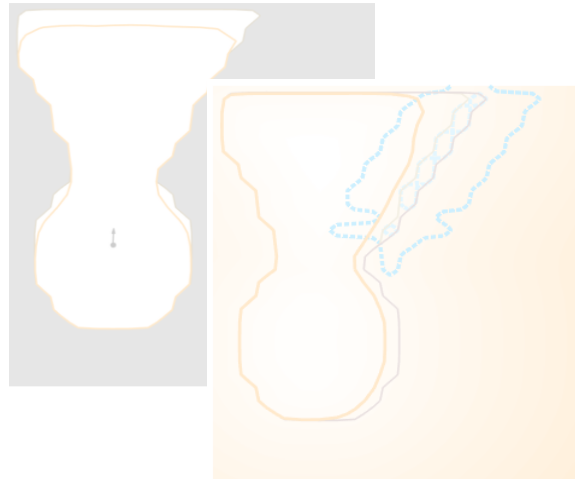
## Safety Filtering



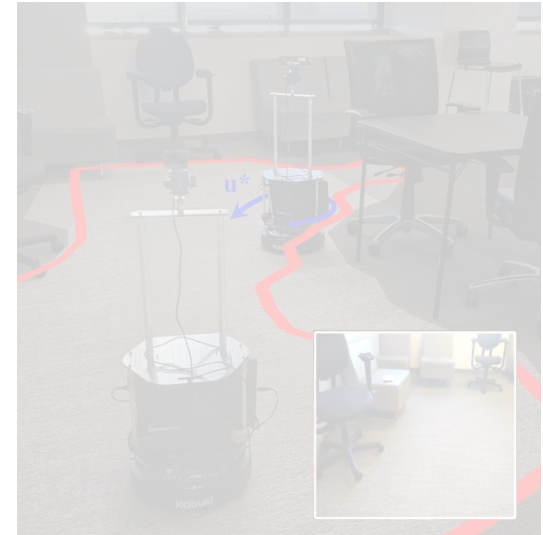
## Setup & Warm Starting

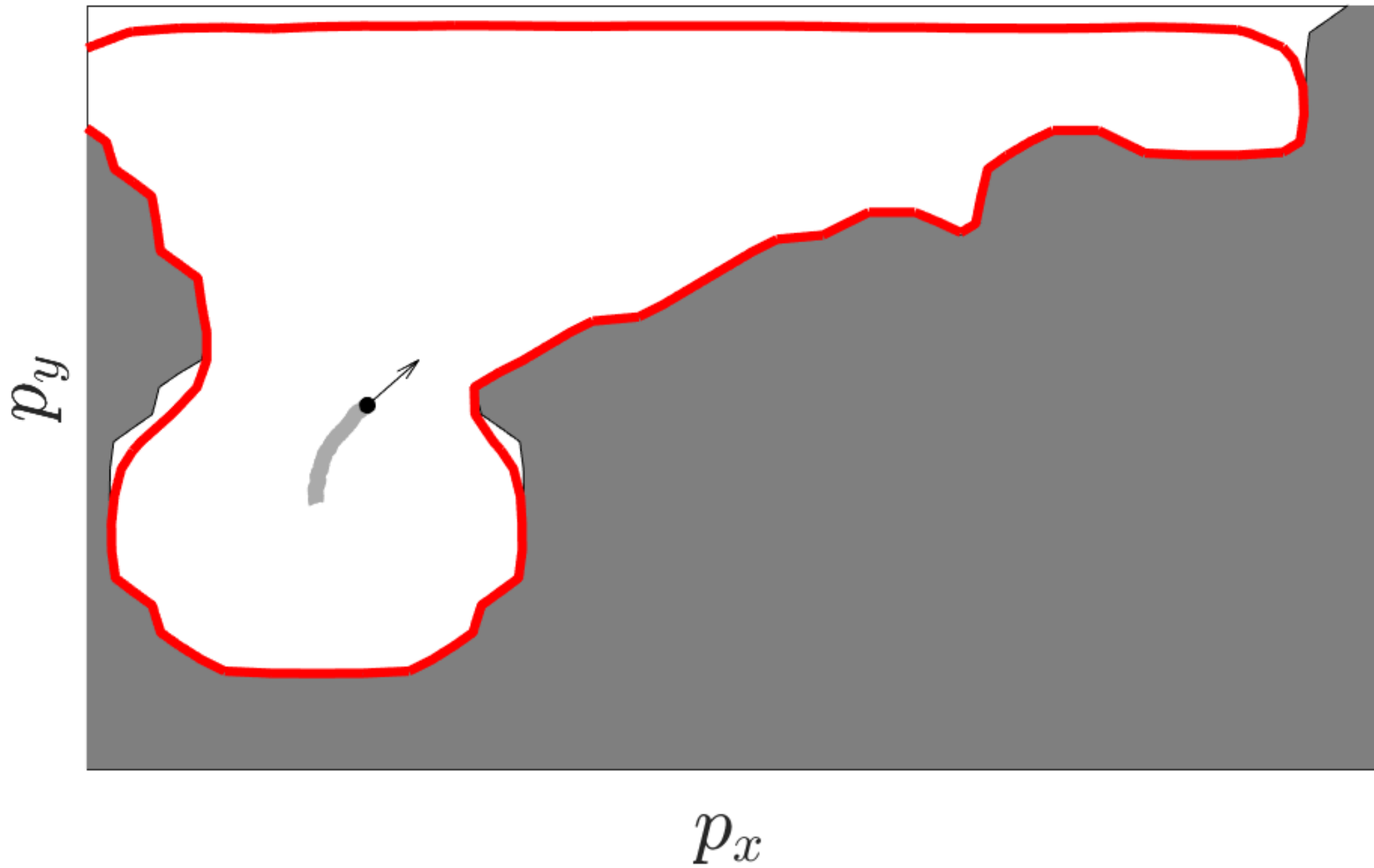


## Local Updates

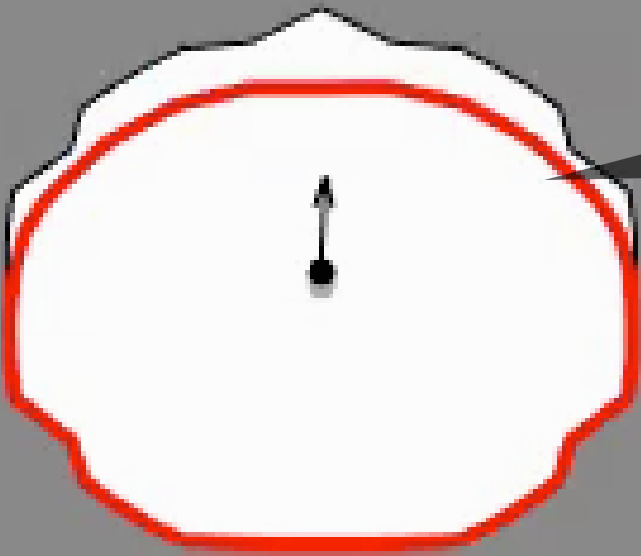


## Safety Filtering



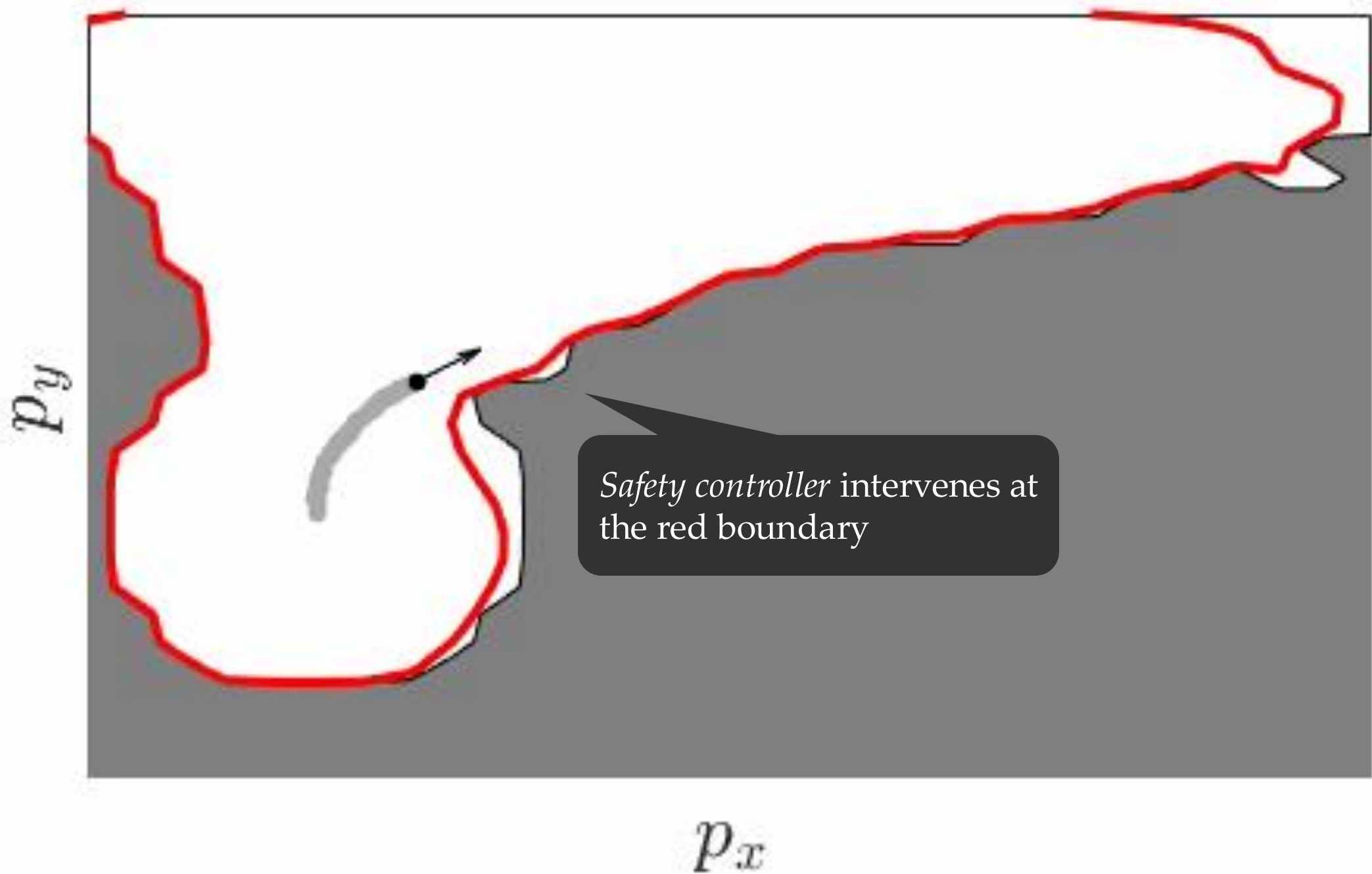


$\mathcal{P}_y$



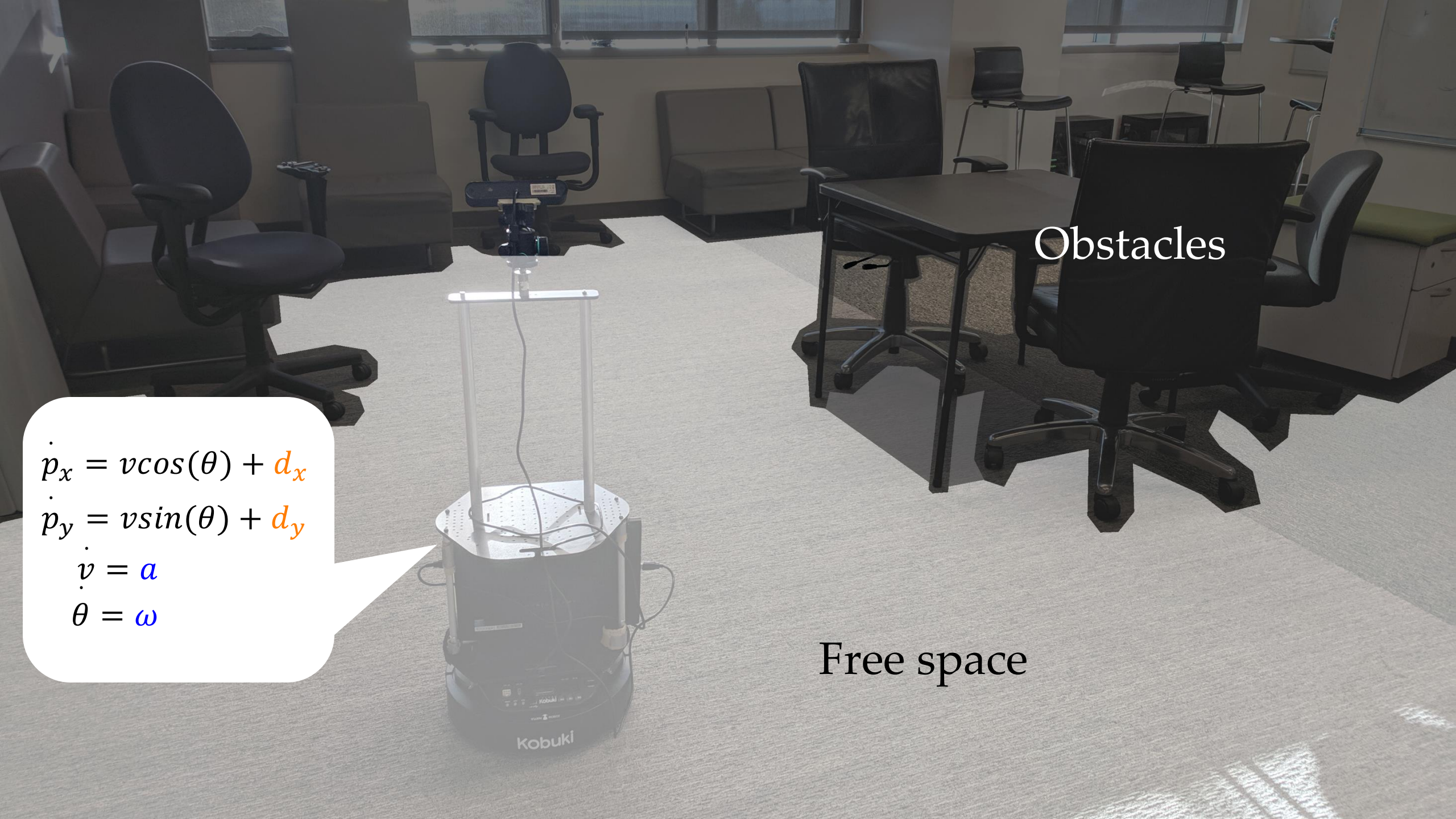
*Any planner can be used within the red boundary*

$\mathcal{P}_x$





$$\begin{aligned}\dot{p}_x &= v \cos(\theta) + d_x \\ \dot{p}_y &= v \sin(\theta) + d_y \\ \dot{v} &= a \\ \dot{\theta} &= \omega\end{aligned}$$



Obstacles

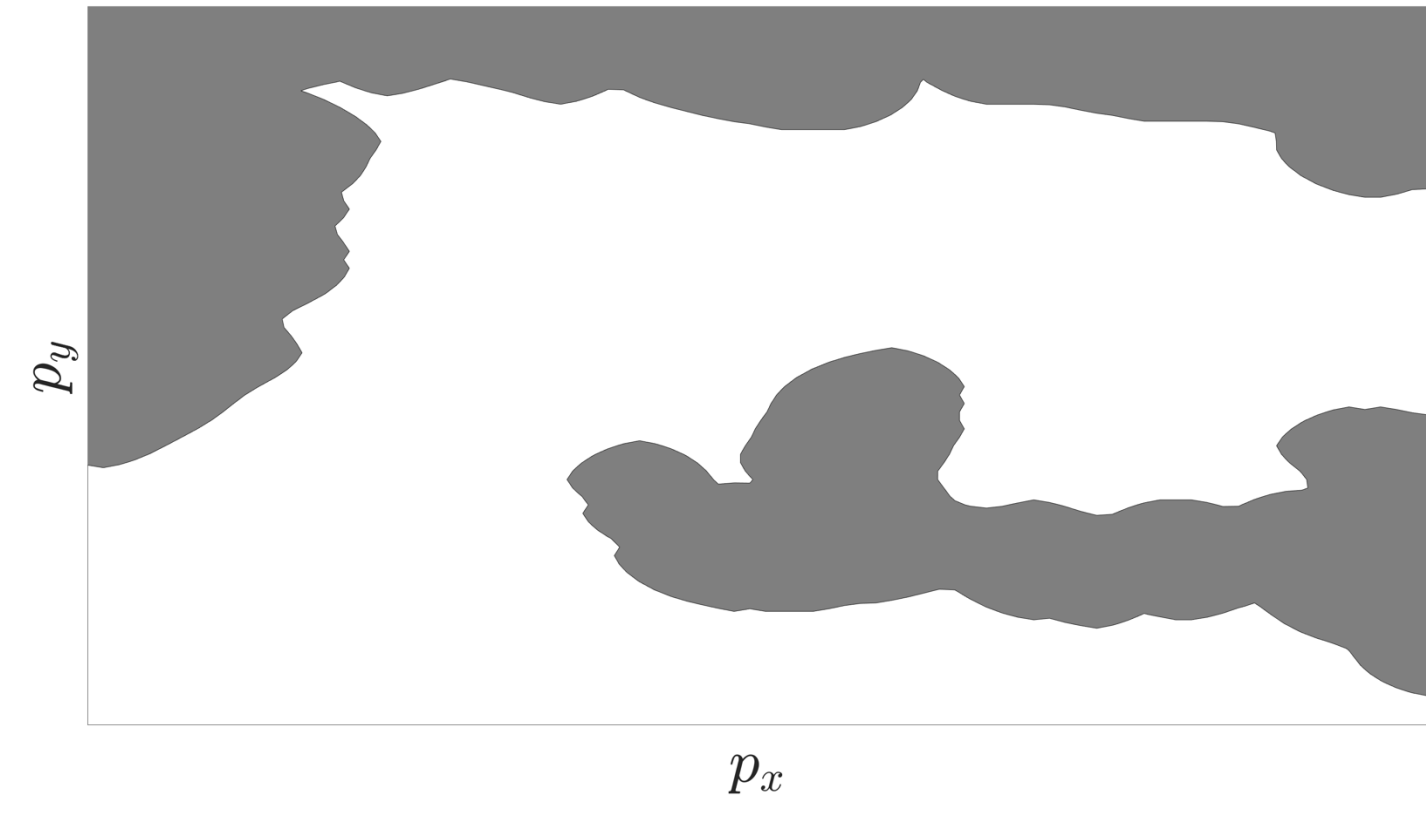
Free space

$$\dot{p}_x = v \cos(\theta) + d_x$$

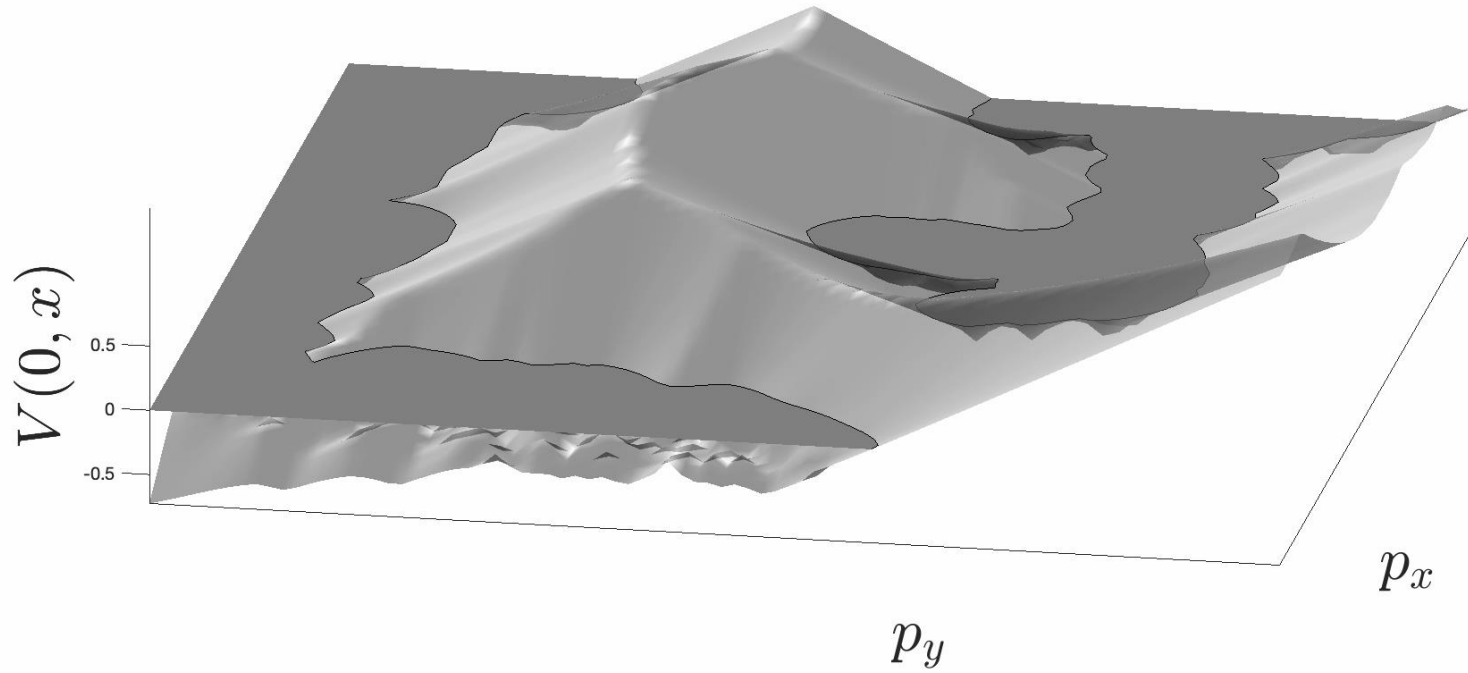
$$\dot{p}_y = v \sin(\theta) + d_y$$

$$\dot{v} = a$$

$$\dot{\theta} = \omega$$





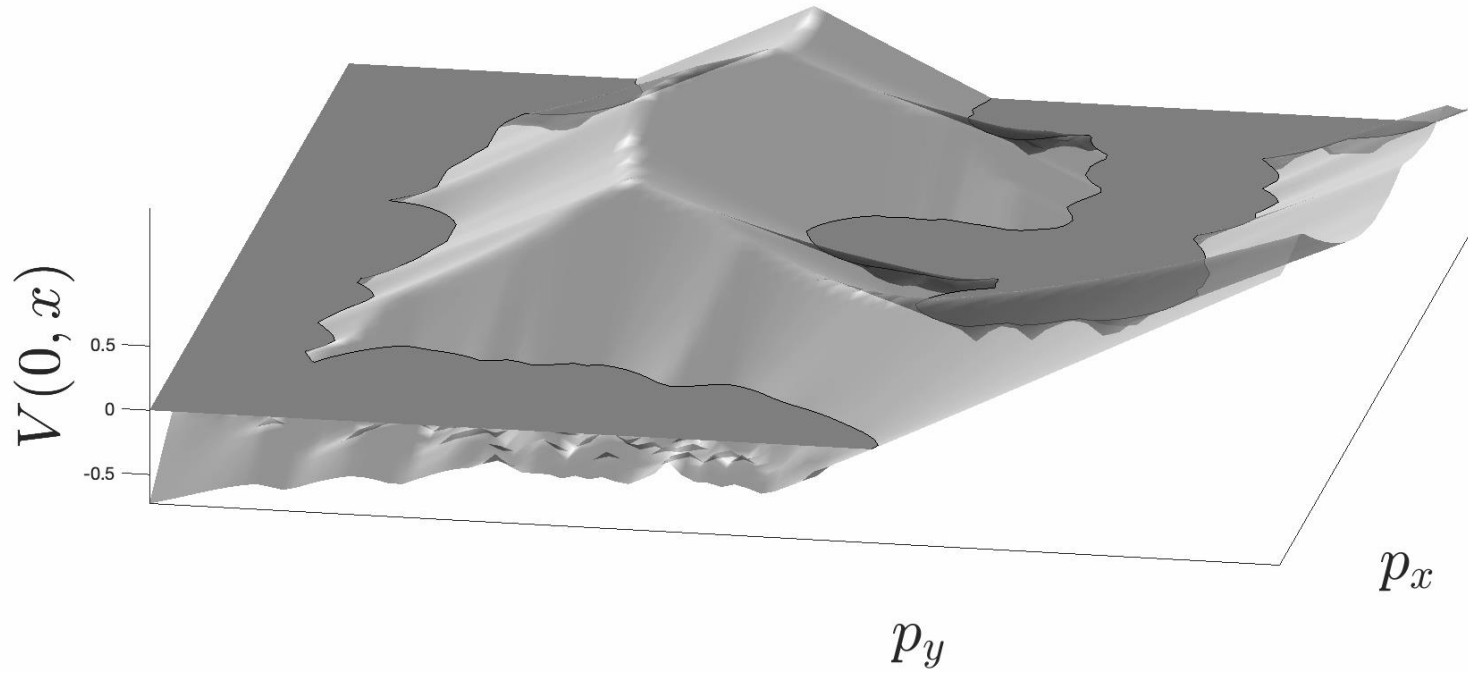


Failure States

$$\mathcal{L} = \{x: l(x) \leq 0\}$$

Value Function

$$V(T, x) = \max_{\pi_u} \min_{\pi_d} \min_{t \in [0, T]} l(\mathbf{x}_{x,t}^{\mathbf{u}, \mathbf{d}}(t))$$

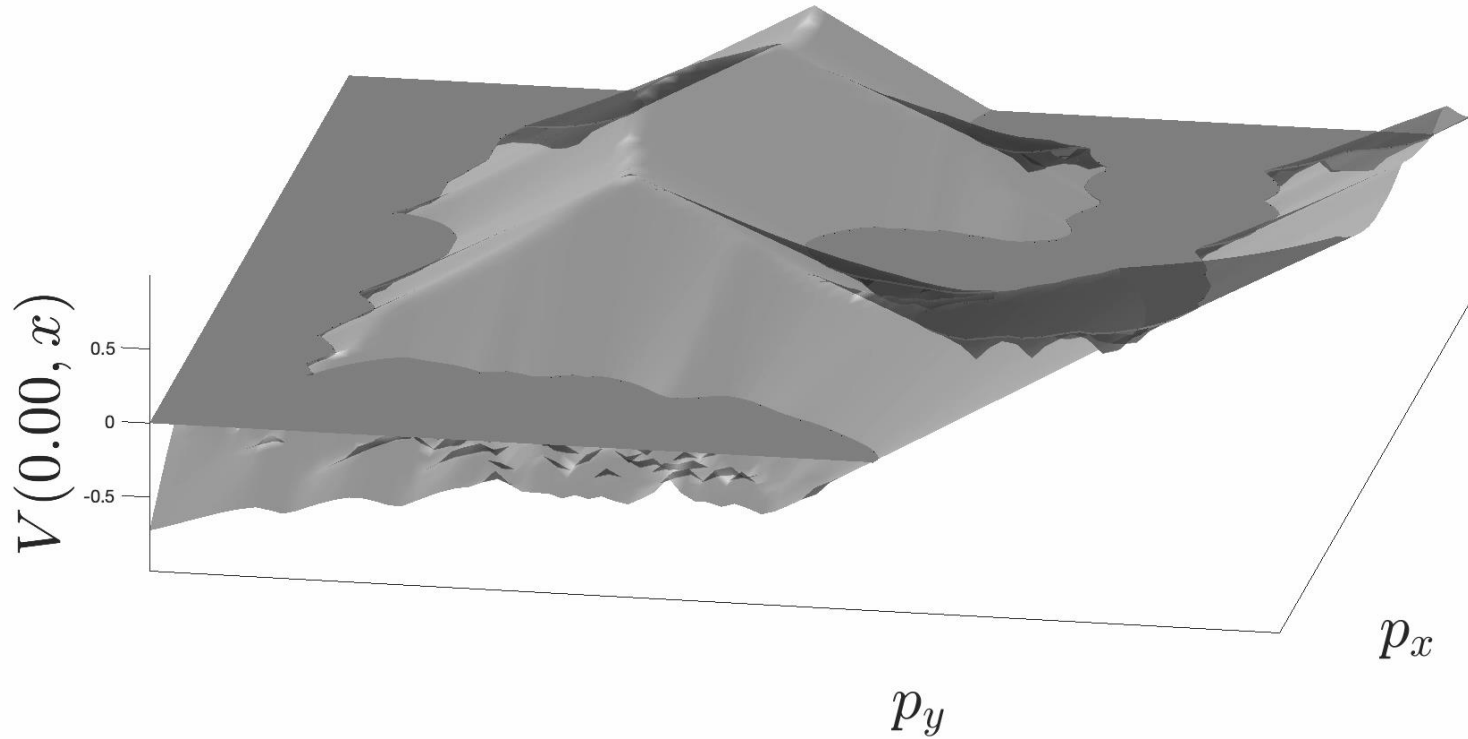


Failure States

$$\mathcal{L} = \{x: l(x) \leq 0\}$$

Value Function

$$V(T, x) = \max_{\pi_u} \min_{\pi_d} \min_{t \in [0, T]} l(\mathbf{x}_{x,t}^{\mathbf{u}, \mathbf{d}}(t))$$



Value Function

$$V(T, x) = \max_{\pi_u} \min_{\pi_d} \min_{t \in [0, T]} l(\mathbf{x}_{x,t}^{u,d}(t))$$

Dynamic Programming

HJI-VI

$$\left\{ \begin{array}{l} \min \left\{ \frac{\partial V}{\partial t} + \mathbf{H}(x, \nabla V), l(x) - V(t, x) \right\} = 0 \\ V(0, x) = l(x) \end{array} \right.$$

Backward Reachable Tube

$$BRT = \{x: V(T, x) \leq 0\}$$

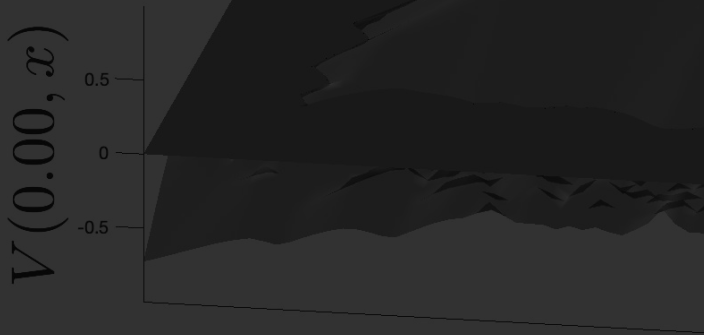
Optimal Control

$$\mathbf{u}^*(x, t) = \operatorname{argmax}_u \min_d \nabla V(t, x)^\top f(x, u, d)$$

Hamiltonian

$$\mathbf{H}(x, \nabla V) = \max_u \min_d \nabla V(t, x)^\top f(x, u, d)$$

$V(0.00, x)$



Computation Time for BRS (seconds)

Full HJ Reachability

51.7

Value Function

$$V(0, x) = \max_{\pi_u} \min_{\pi_d} \min_{t \in [0, T]} l(\mathbf{x}_{x,t}^{\mathbf{u}, \mathbf{d}}(t))$$

Dynamic Programming

$$\left. \begin{aligned} &+ H(x, \nabla V), l(x) - V(t, x) \end{aligned} \right\} = 0$$

$$V(0, x) = l(x)$$

Backward Reachable Tube

$$BRT = \{x: V(T, x) \leq 0\}$$

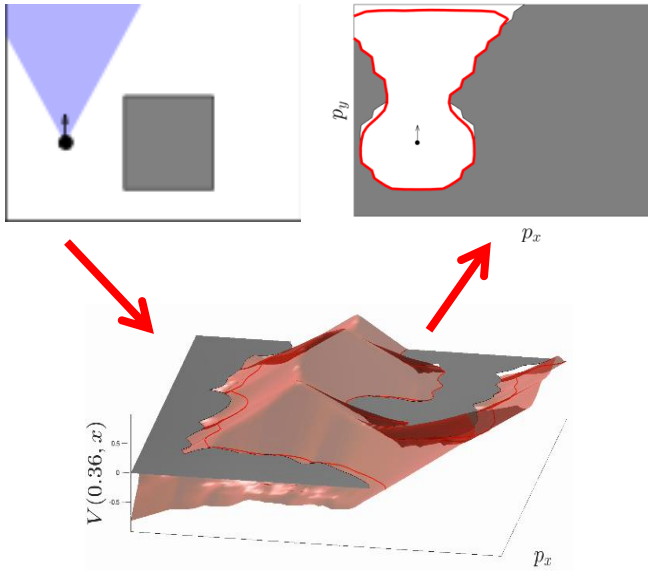
Optimal Control

$$\mathbf{u}^*(x, t) = \operatorname{argmax}_u \min_d \nabla V(t, x)^\top f(x, \mathbf{u}, \mathbf{d})$$

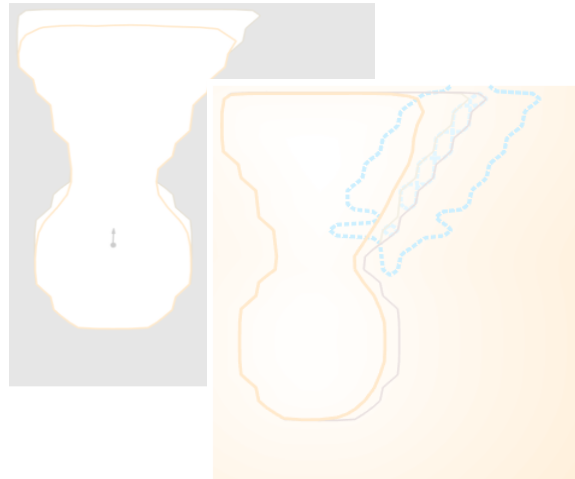
Hamiltonian

$$H(x, \nabla V) = \max_u \min_d \nabla V(t, x)^\top f(x, \mathbf{u}, \mathbf{d})$$

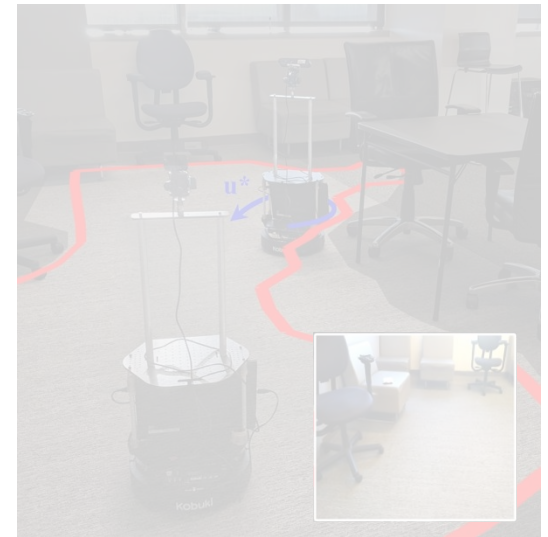
## Setup & Warm Starting



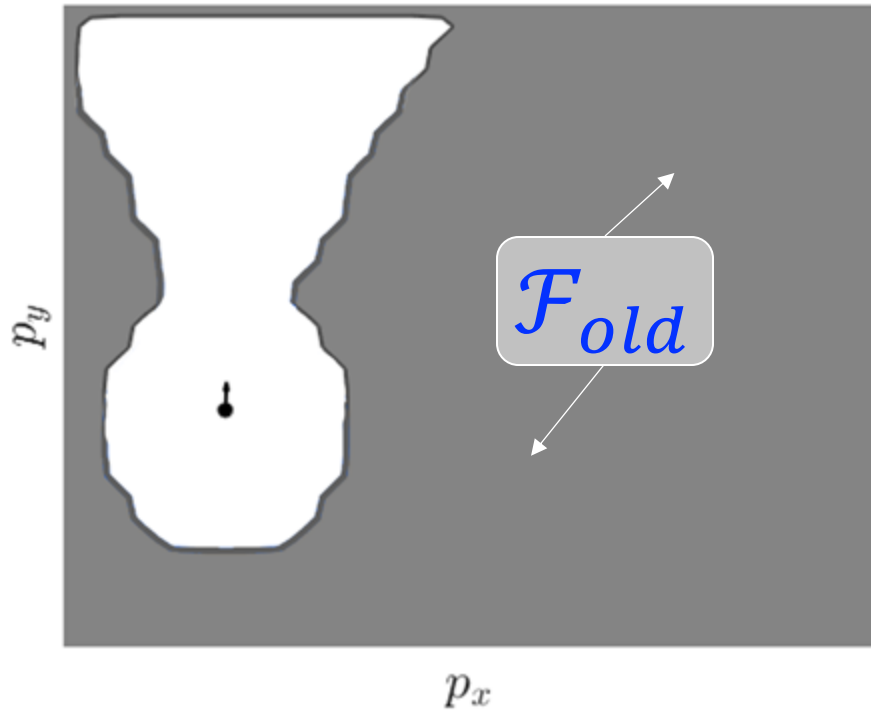
## Local Updates



## Safety Filtering



# Warm Starting Reachability Computation



Initial Failure Set:  $\mathcal{F}_{old} := \{x: l_{old}(x) \leq 0\}$

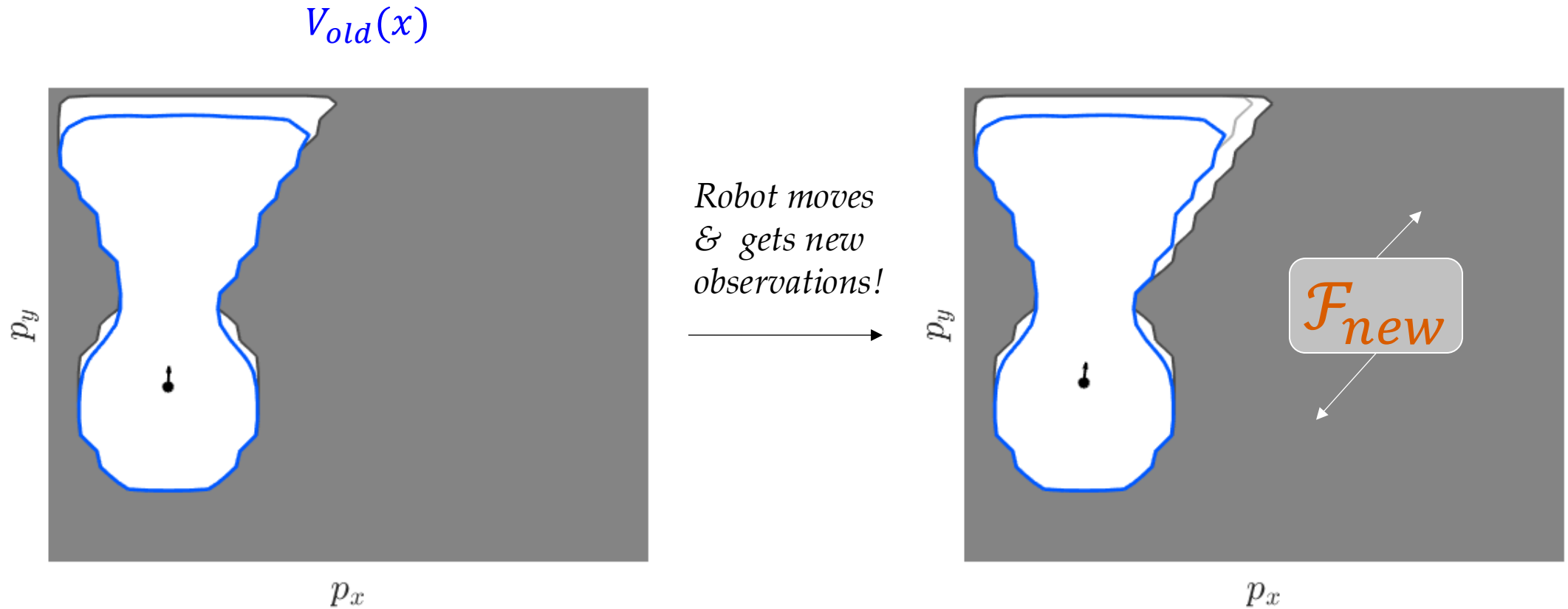
Initial Safety Computation:

$$\min \left\{ \frac{\partial V}{\partial t} + H(x, \nabla V), l_{old}(x) - V(t, x) \right\} = 0$$

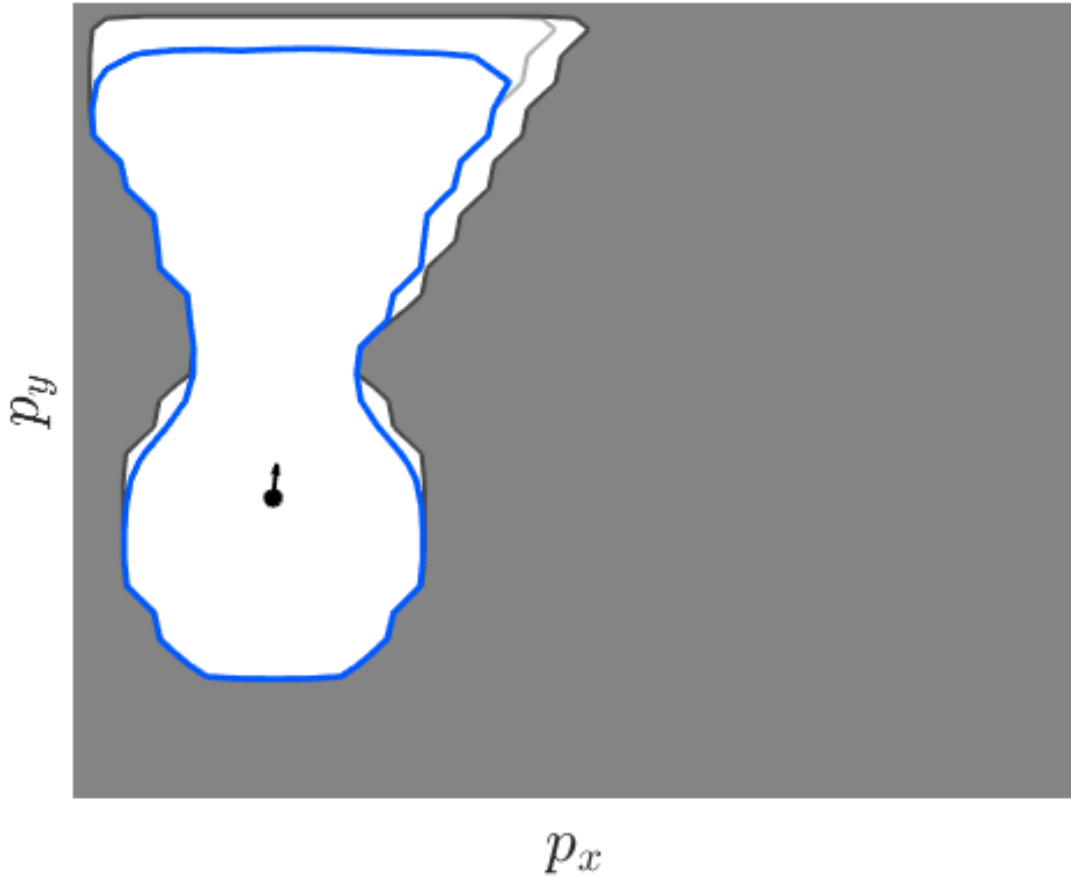
$$V(0, x) = l_{old}(x)$$

$$\begin{array}{c} \downarrow t \rightarrow \infty \\ V_{old}(x) \end{array}$$

# Warm Starting Reachability Computation



# Warm Starting Reachability Computation



New Failure Set:  $\mathcal{F}_{new} := \{x: l_{new}(x) \leq 0\}$

Safety Computation:

$$\min \left\{ \frac{\partial V}{\partial t} + H(x, \nabla V), l_{new}(x) - V(t, x) \right\} = 0$$

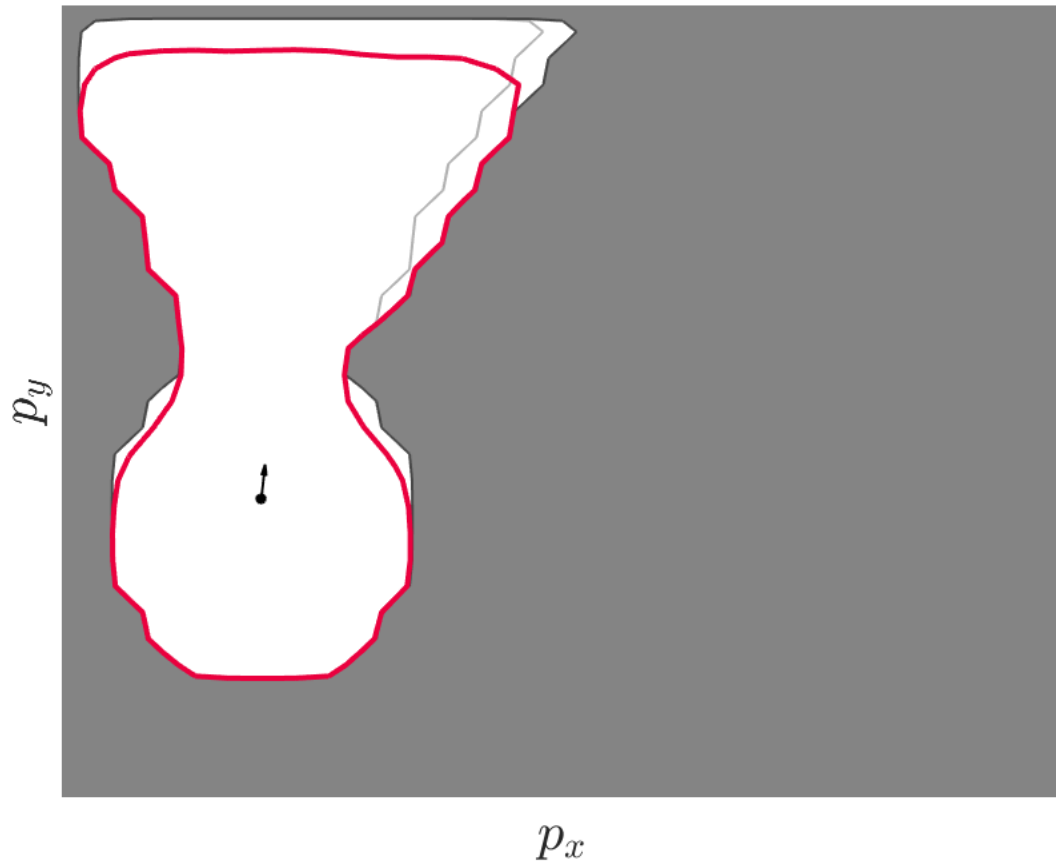
$$V(0, x) = V_{old}(x)$$

$$\downarrow t \rightarrow \infty$$

$$V_{new}(x)$$



# Warm Starting Reachability Computation



$$\min \left\{ \frac{\partial V}{\partial t} + H(x, \nabla V), l_{new}(x) - V(t, x) \right\} = 0$$

$$V(0, x) = V_{old}(x)$$

$$\downarrow t \rightarrow \infty$$

$$V_{new}(x)$$

Computation time for BRT (s)	
Full HJ Reachability	Warm-started Reachability
51.7	12.5

# An Efficient Reachability-Based Framework for Provably Safe Autonomous Navigation in Unknown Environments

Andrea Bajcsy\*, Somil Bansal\*, Eli Bronstein, Varun Tolani, Claire J. Tomlin

**Abstract**—Real-world autonomous vehicles often operate in *a priori* unknown environments. Since most of these systems are safety-critical, it is important to ensure they operate safely in the face of environment uncertainty, such as unseen obstacles. Current safety analysis tools enable autonomous systems to reason about safety given full information about the state of the environment *a priori*. However, these tools do not scale well to scenarios where the environment is being sensed in real time, such as during navigation tasks. In this work, we propose a novel, real-time safety analysis method based on Hamilton-Jacobi reachability that provides strong safety guarantees despite environment uncertainty. Our safety method is planner-agnostic and provides guarantees for a variety of mapping sensors. We demonstrate our approach in simulation and in hardware to provide safety guarantees around a state-of-the-art vision-based, learning-based planner. Videos of our approach and experiments are available on the project website<sup>1</sup>.

## I. INTRODUCTION

Autonomous vehicles operating in the real world must navigate through *a priori* unknown environments using on-board, limited-range sensors. As a vehicle makes progress towards a goal and receives new sensor information about the environment, rigorous safety analysis is critical to ensure that the system's behavior does not lead to dangerous collisions. In order to provide such safety guarantees for real vehicles, any analysis should take into account multiple sources of uncertainty, such as modelling error, external disturbances, and unknown parts of the environment.

A variety of mechanisms have been proposed to ensure robustness to modeling error and external disturbances [24], [16], [34]. Additionally, safety guarantees for systems using

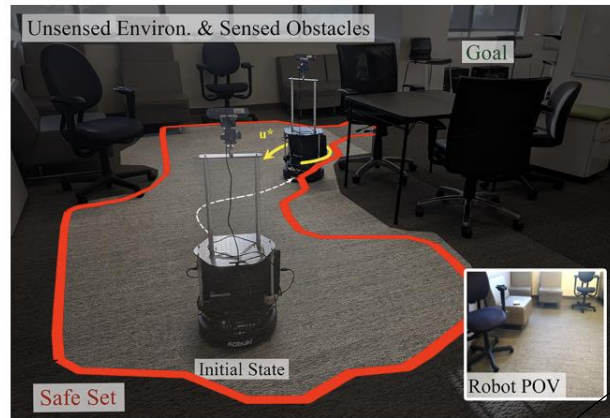


Fig. 1. **Overview:** We consider the problem of safe navigation from an initial state to a goal state in an *a priori* unknown environment. Our approach treats the unsensed environment as an obstacle, and uses a HJ reachability framework to compute a safe controller for the vehicle, which is updated in real-time as the vehicle explores the environment. We show an application of our approach on a Turtlebot using a vision-based planner. When the robot is at risk of colliding, the safe controller ( $u^*$ ) keep the system safe.

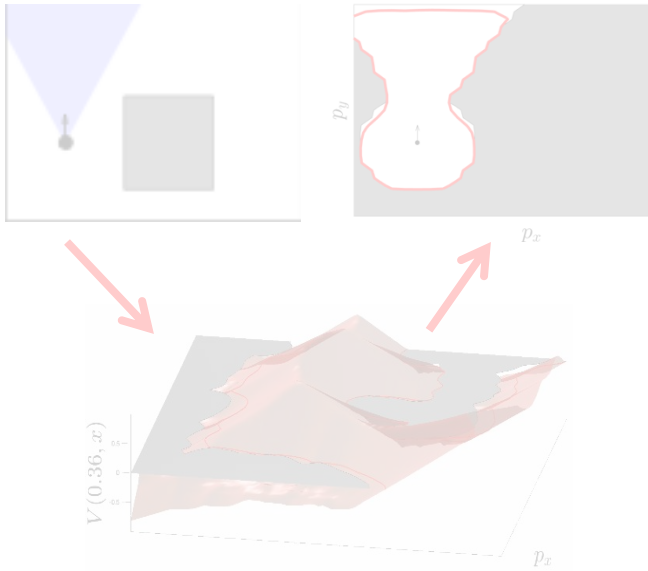
external disturbances while minimally interfering with goal-driven behavior. Second, real-time safety assurances need to be provided as new environment information is acquired, which requires approximations that are both computationally efficient and not overly conservative. Moreover, this safety analysis should be applicable to a wide variety of real-world sensors, planners, and vehicles.

In this paper, we propose a safety framework that can overcome these challenges for autonomous vehicles operating in

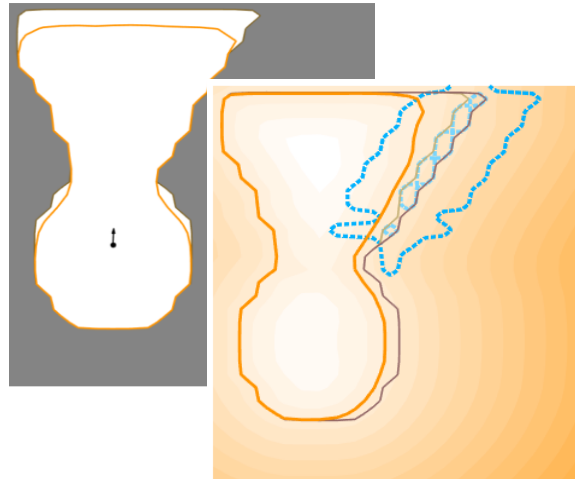
**Lemma (Informal):** The safe set obtained by warm-starting is an *under-approximation* of the true safe set obtained by solving full HJI-VI.

! We can use warm-starting to ensure safety for the vehicle while being computationally efficient!

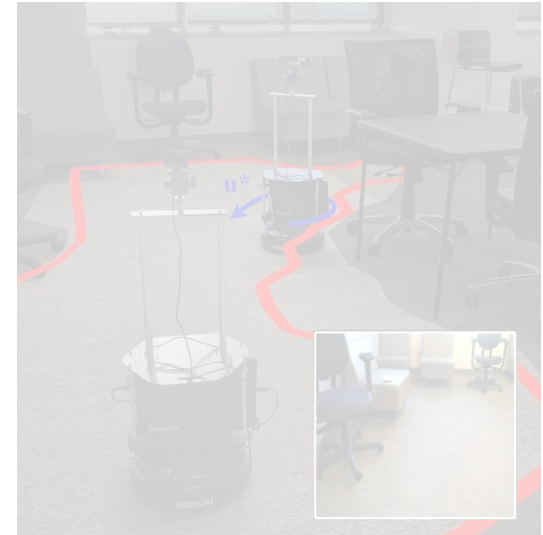
## Setup & Warm Starting



## Local Updates



## Safety Filtering



# Local Update of the BRT

$$V_{old} \leftarrow V_t(0, Q)$$

$Q \leftarrow$  new free states and neighbors

while  $Q$  is not empty do:

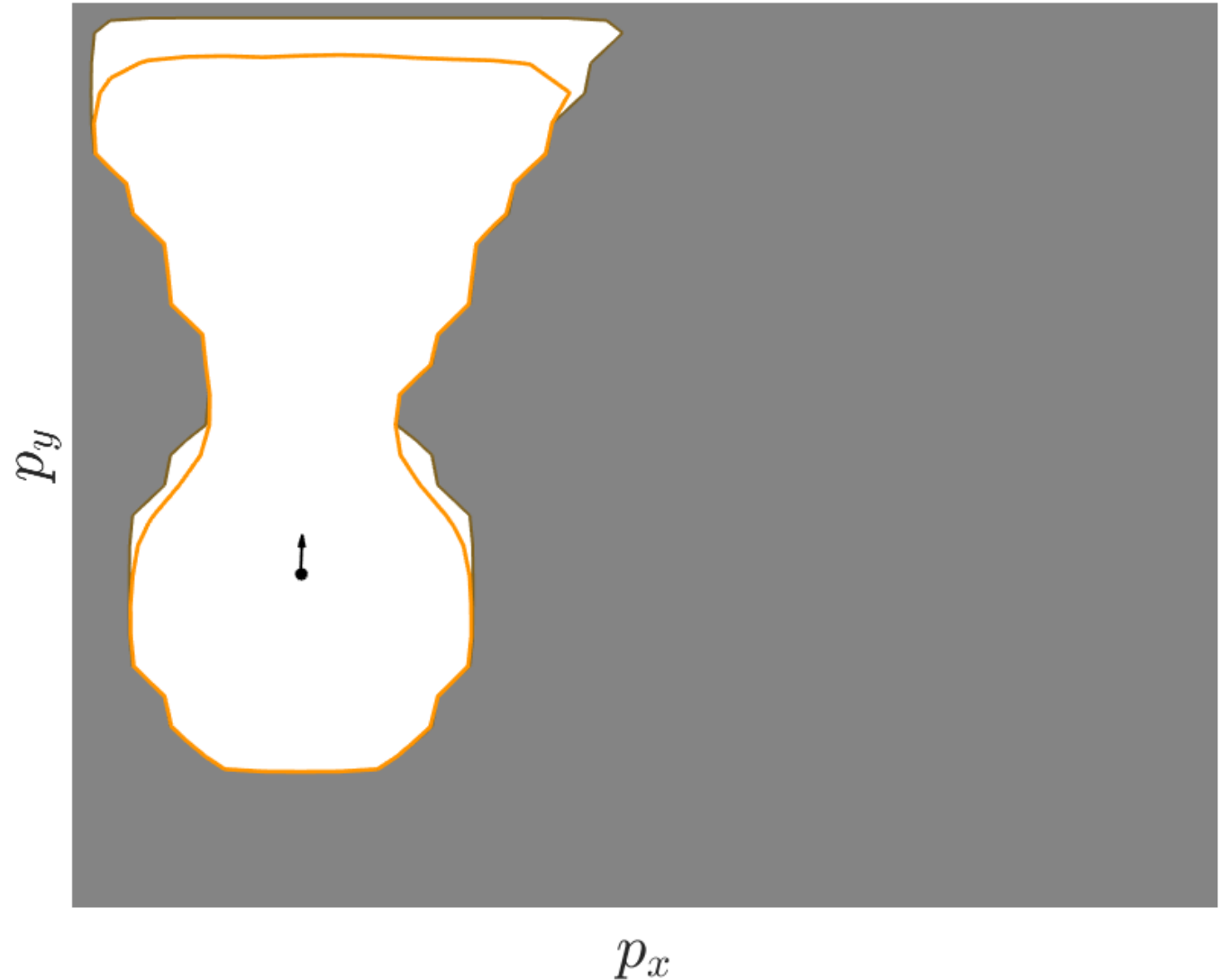
$V_{update} \leftarrow$  update  $V_{old}$  for  $\Delta T$

$$\Delta V = ||V_{update} - V_{old}||$$

$Q \leftarrow$  remove states with  $\Delta V = 0$

$Q \leftarrow$  add neighbors

$$V_{old} \leftarrow V_{update}$$



# Local Update of the BRT

$$V_{old} \leftarrow V_t(0, Q)$$

$Q \leftarrow$  new free states and neighbors

while  $Q$  is not empty do:

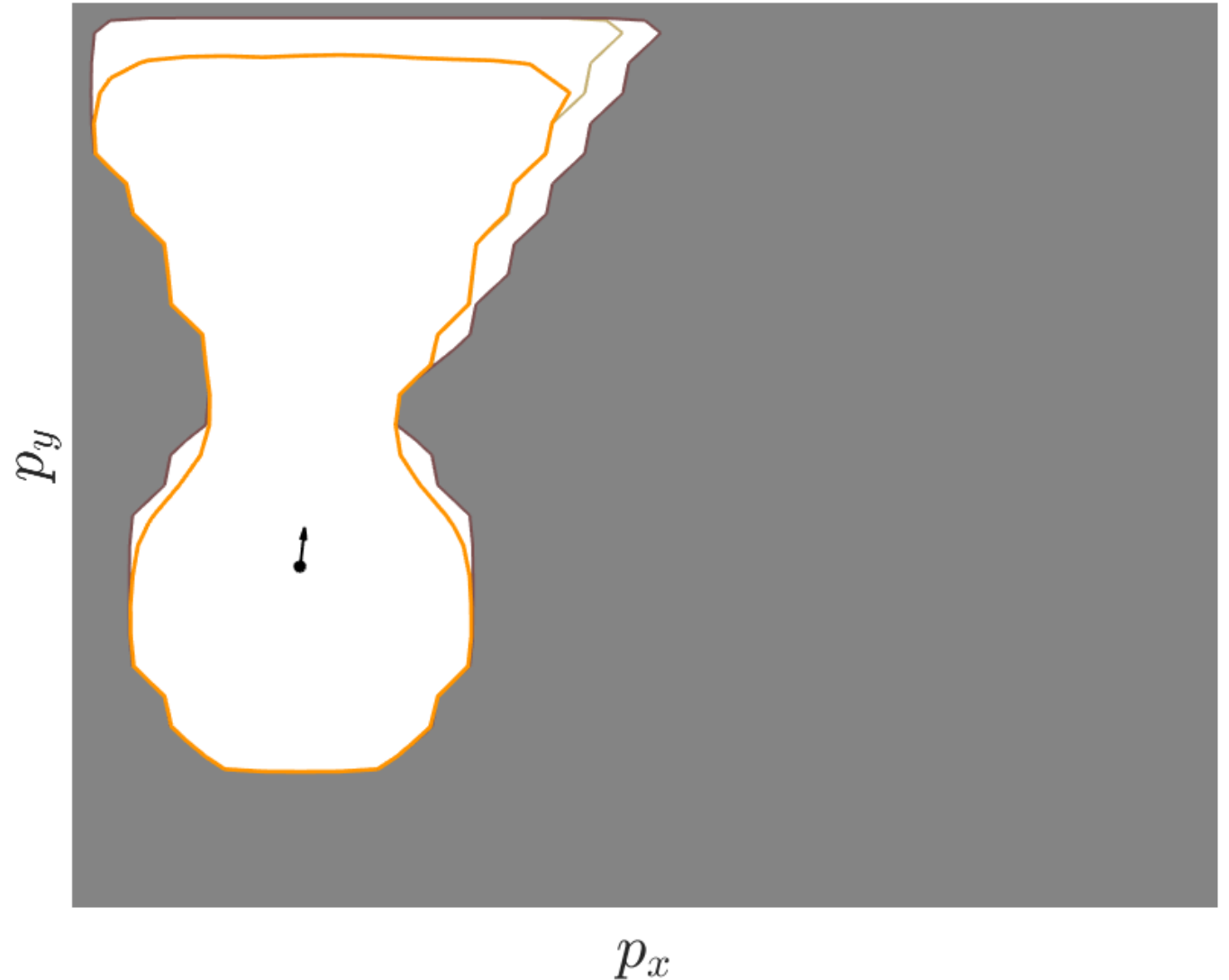
$V_{update} \leftarrow$  update  $V_{old}$  for  $\Delta T$

$$\Delta V = ||V_{update} - V_{old}||$$

$Q \leftarrow$  remove states with  $\Delta V = 0$

$Q \leftarrow$  add neighbors

$$V_{old} \leftarrow V_{update}$$



# Local Update of the BRT

$$V_{old} \leftarrow V_t(0, Q)$$

$Q \leftarrow$  new free states and neighbors

while  $Q$  is not empty do:

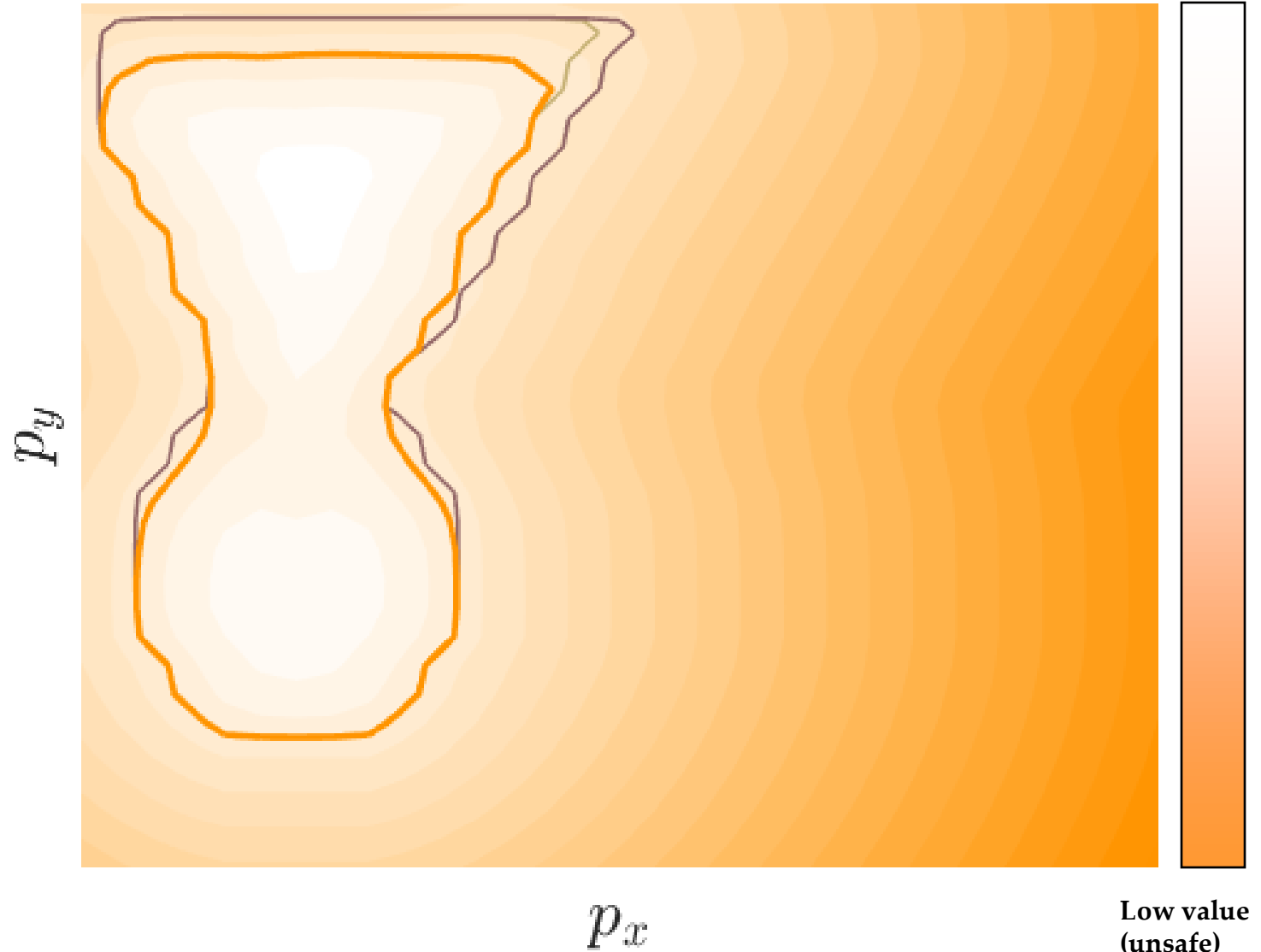
$$V_{update} \leftarrow \text{update } V_{old} \text{ for } \Delta T$$

$$\Delta V = ||V_{update} - V_{old}||$$

$Q \leftarrow$  remove states with  $\Delta V = 0$

$Q \leftarrow$  add neighbors

$$V_{old} \leftarrow V_{update}$$



# Local Update of the BRT

$$V_{old} \leftarrow V_t(0, Q)$$

$Q \leftarrow$  new *free states* and neighbors

while  $Q$  is not empty do:

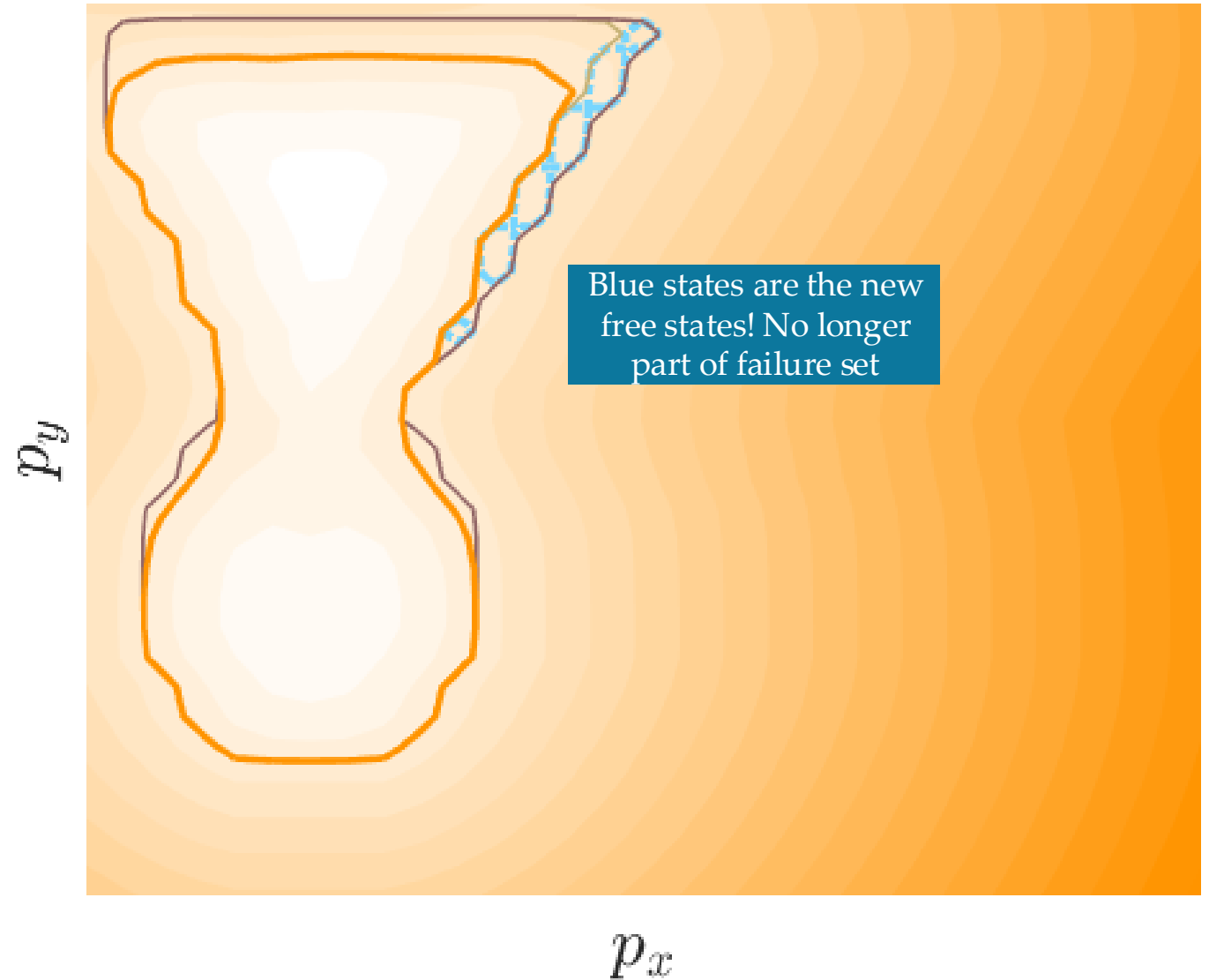
$$V_{update} \leftarrow \text{update } V_{old} \text{ for } \Delta T$$

$$\Delta V = ||V_{update} - V_{old}||$$

$Q \leftarrow$  remove states with  $\Delta V = 0$

$Q \leftarrow$  add neighbors

$$V_{old} \leftarrow V_{update}$$



# Local Update of the BRT

$$V_{old} \leftarrow V_t(0, Q)$$

$Q \leftarrow$  new free states and *neighbors*

while  $Q$  is not empty do:

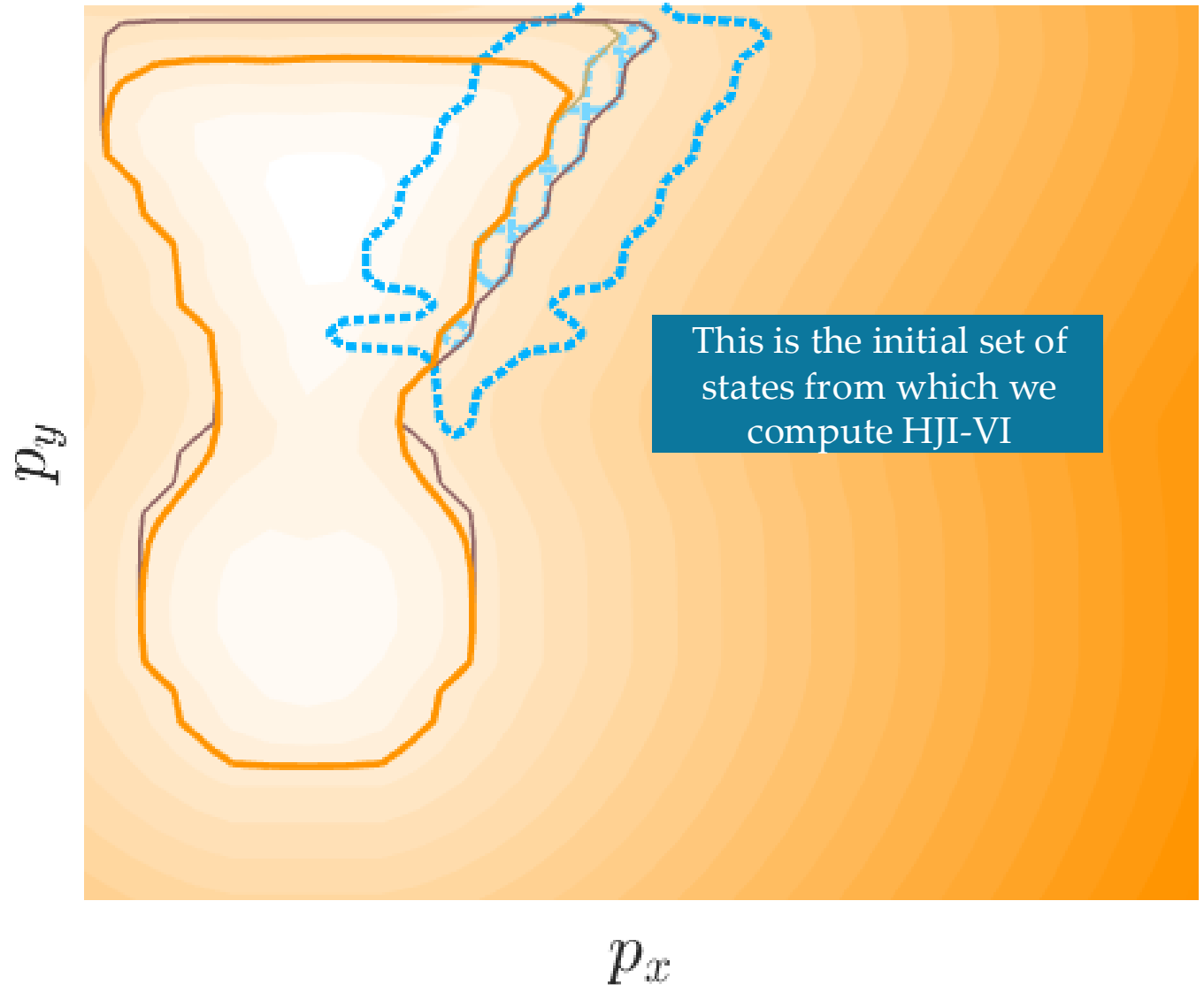
$$V_{update} \leftarrow \text{update } V_{old} \text{ for } \Delta T$$

$$\Delta V = ||V_{update} - V_{old}||$$

$Q \leftarrow$  remove states with  $\Delta V = 0$

$Q \leftarrow$  add neighbors

$$V_{old} \leftarrow V_{update}$$





# Local Update of the BRT

$$V_{old} \leftarrow V_t(0, Q)$$

$Q \leftarrow$  new free states and *neighbors*

while  $Q$  is not empty do:

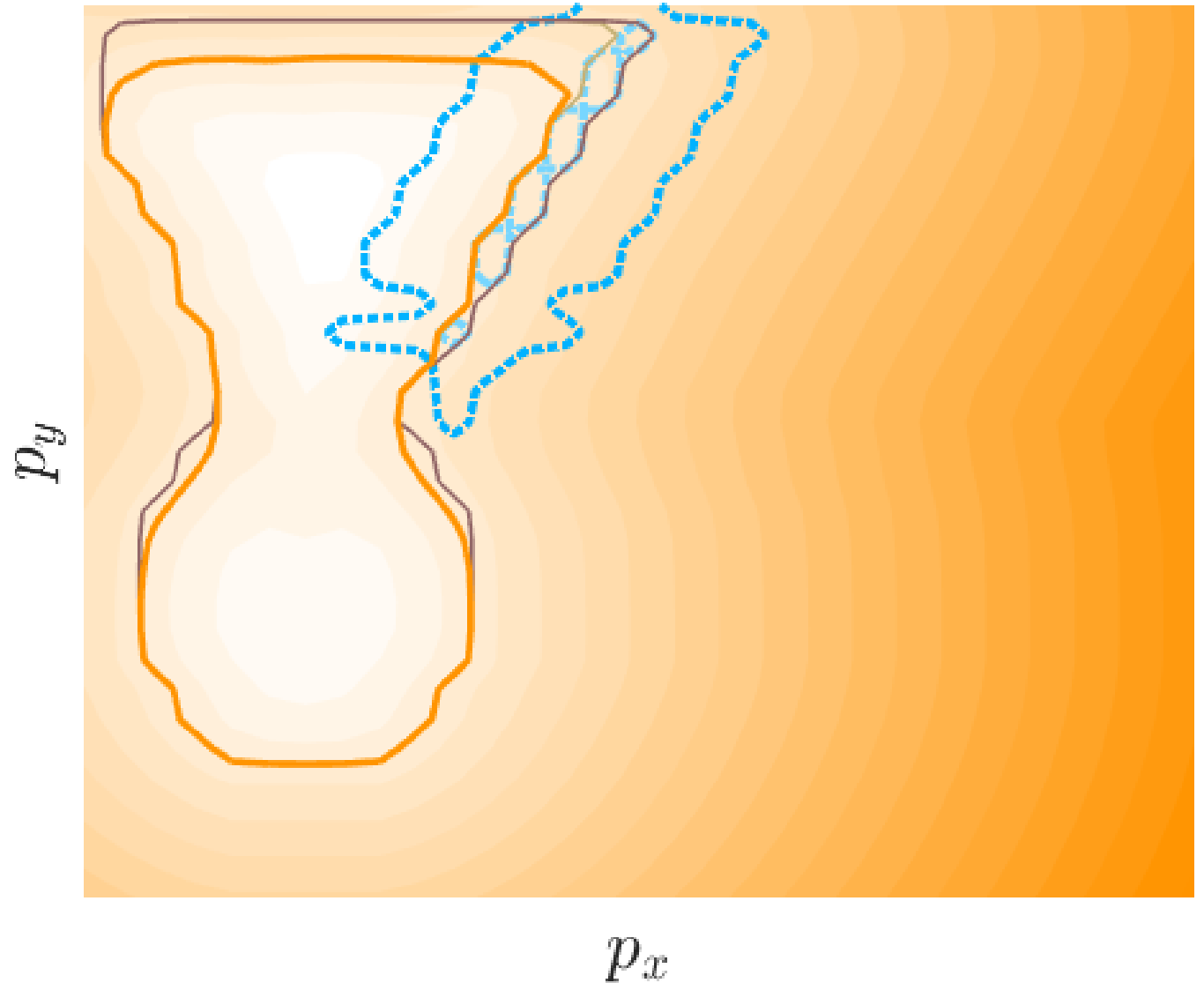
$$V_{update} \leftarrow \text{update } V_{old} \text{ for } \Delta T$$

$$\Delta V = ||V_{update} - V_{old}||$$

$Q \leftarrow$  remove states with  $\Delta V = 0$

$Q \leftarrow$  add neighbors

$$V_{old} \leftarrow V_{update}$$



# Local Update of the BRT

$$V_{old} \leftarrow V_t(0, Q)$$

$Q \leftarrow$  new free states and *neighbors*

while  $Q$  is not empty do:

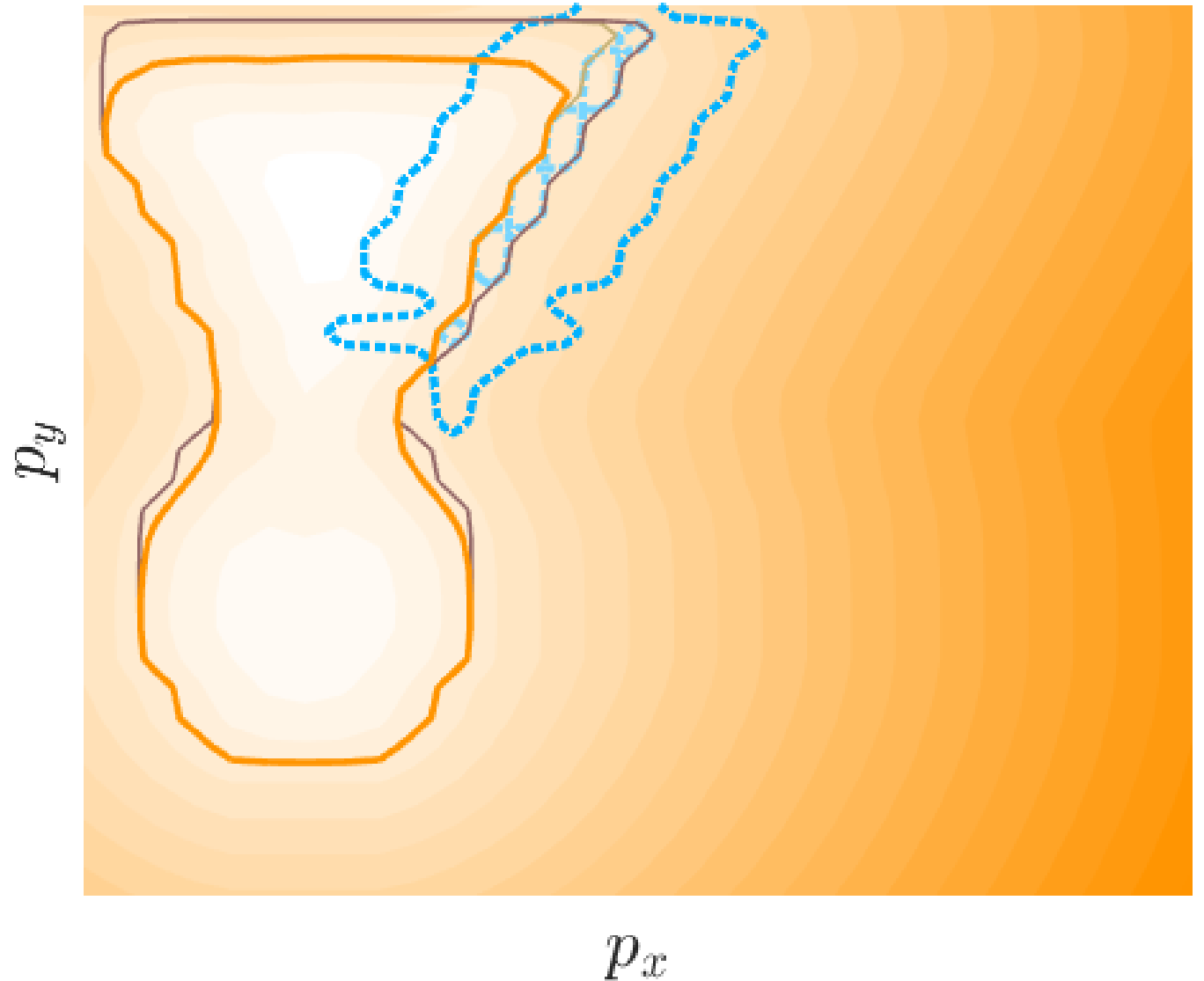
$$V_{update} \leftarrow \text{update } V_{old} \text{ for } \Delta T$$

$$\Delta V = ||V_{update} - V_{old}||$$

$Q \leftarrow$  remove states with  $\Delta V = 0$

$Q \leftarrow$  add neighbors

$$V_{old} \leftarrow V_{update}$$



# Local Update Value Propagation

$$V_{old} \leftarrow V_t(0, Q)$$

$Q \leftarrow$  new free states and neighbors

while  $Q$  is not empty do:

$$V_{update} \leftarrow \text{update } V_{old} \text{ for } \Delta T$$

$$\Delta V = ||V_{update} - V_{old}||$$

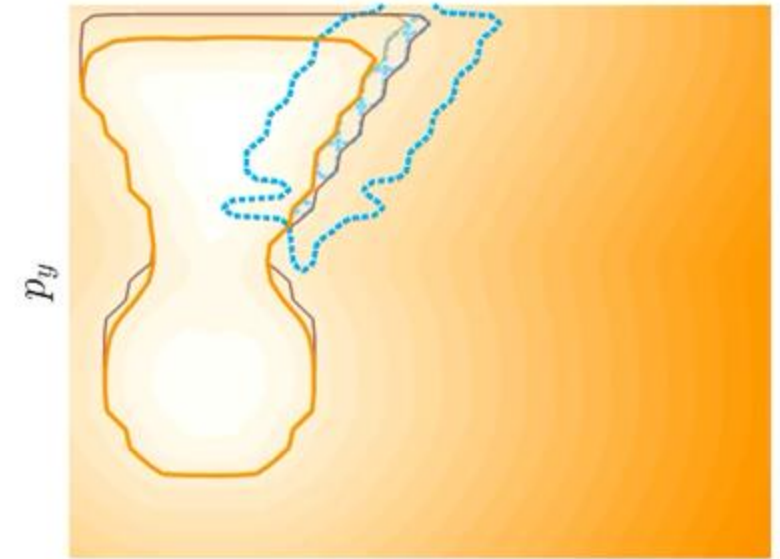
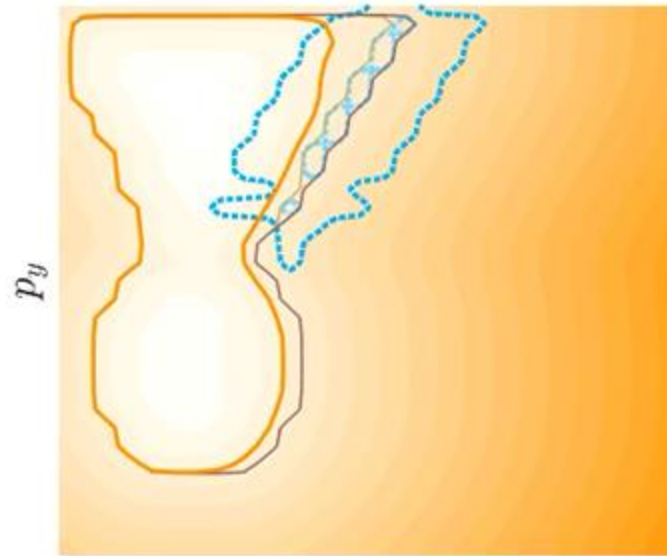
$Q \leftarrow$  remove states with  $\Delta V = 0$

$Q \leftarrow$  add neighbors

$$V_{old} \leftarrow V_{update}$$

● → Slice:  $\theta = 0$

● ↑ Slice:  $\theta = \frac{\pi}{2}$



$p_x$

$p_x$

$p_y$

$p_y$

# Local Update Value Propagation

$V_{old} \leftarrow V_t(0, Q)$

●→ *Slice:  $\theta = 0$*

●↑ *Slice:  $\theta = \frac{\pi}{2}$*

$Q \leftarrow \text{new fr}$

while  $Q$  is not

$V_{update}$

$\Delta V = ||$

$Q \leftarrow \text{re}$

$Q \leftarrow \text{add neighbors}$

$V_{old} \leftarrow V_{update}$

Computation Time for BRS (seconds)

Full HJ Reachability

Local Update Method

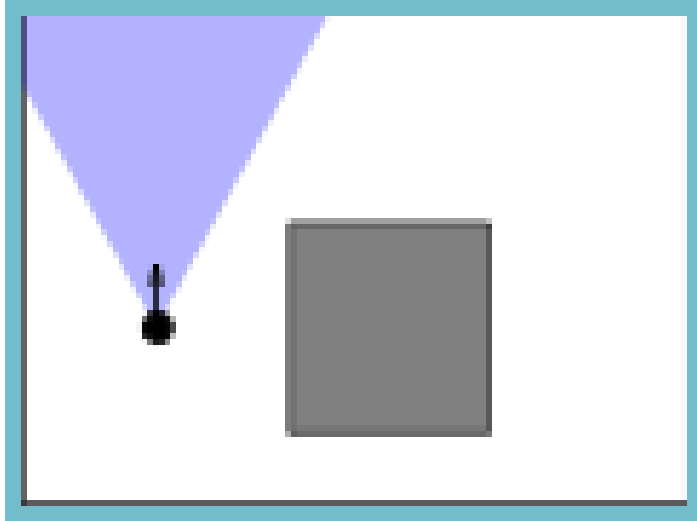
51.7

**0.9**

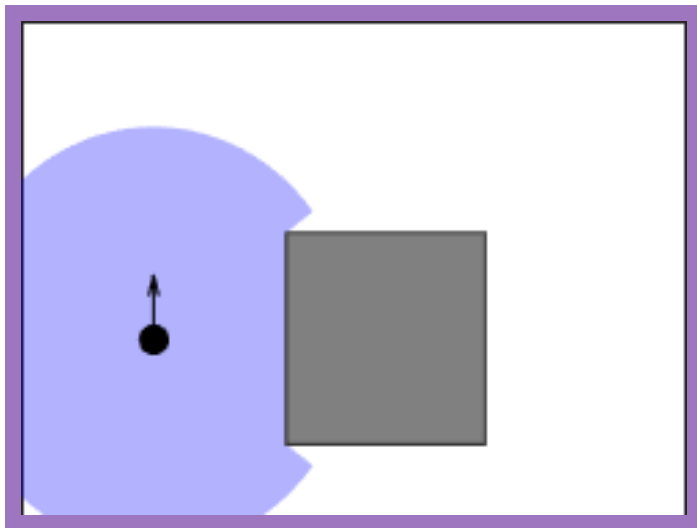
$p_x$

$p_x$

# Simulation Results

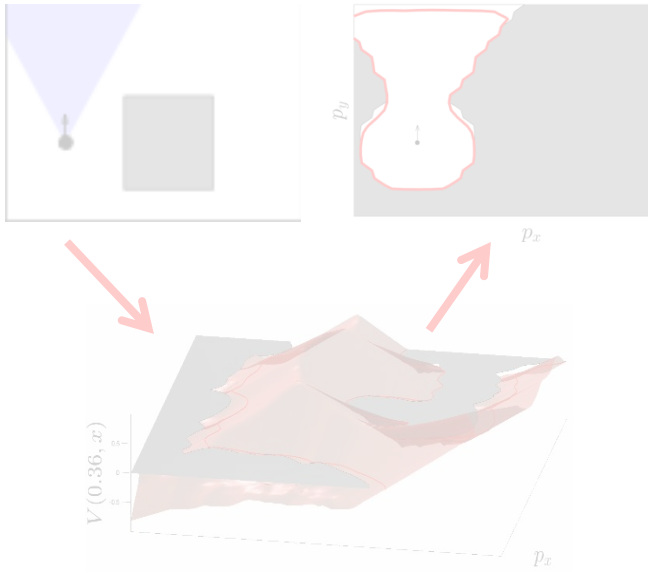


Simulated Camera Results				
Metric	Planner	HJI-VI	Warm	Local
Average Compute Time (s)	RRT	45.688	26.290	0.596
	Spline	51.723	12.489	0.898
% Over-conservative States	RRT	0.0	1.112	0.517
	Spline	0.0	0.474	0.506

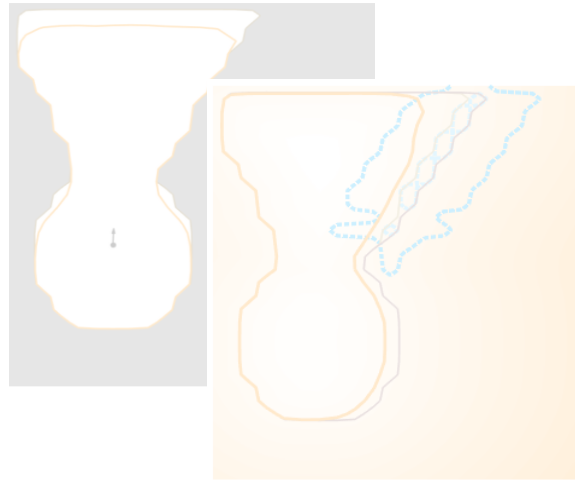


Simulated LiDAR Results				
Metric	Planner	HJI-VI	Warm	Local
Average Compute Time (s)	RRT	21.145	6.075	1.108
	Spline	25.318	3.789	1.158
% Over-conservative States	RRT	0.0	0.032	0.290
	Spline	0.0	0.024	0.240

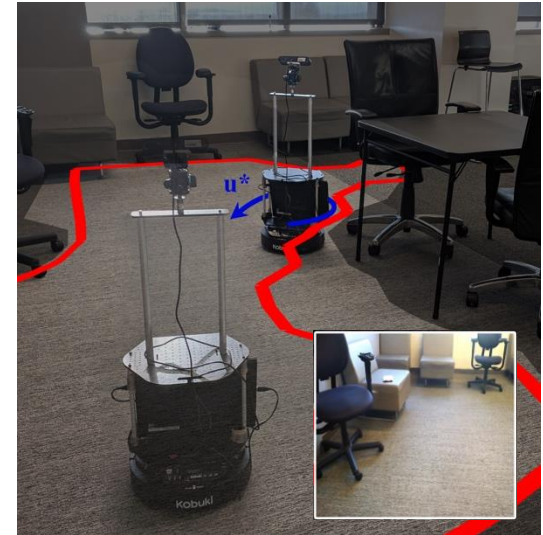
## Setup & Warm Starting

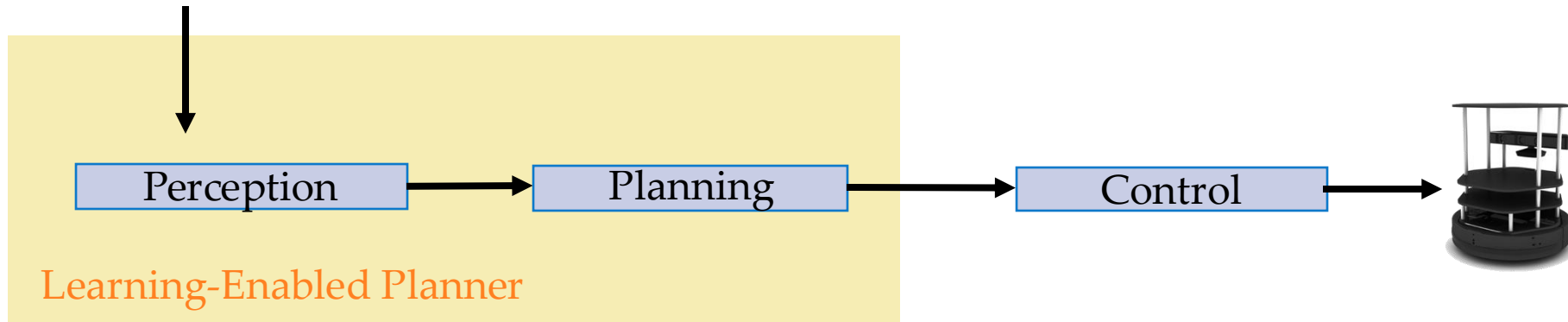
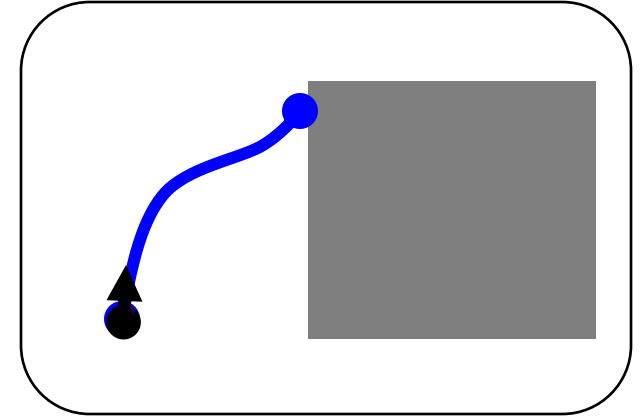


## Local Updates



## Safety Filtering





# LB-WayPtNav Autonomy Stack [1]

No Safety Controller



Third-Person POV

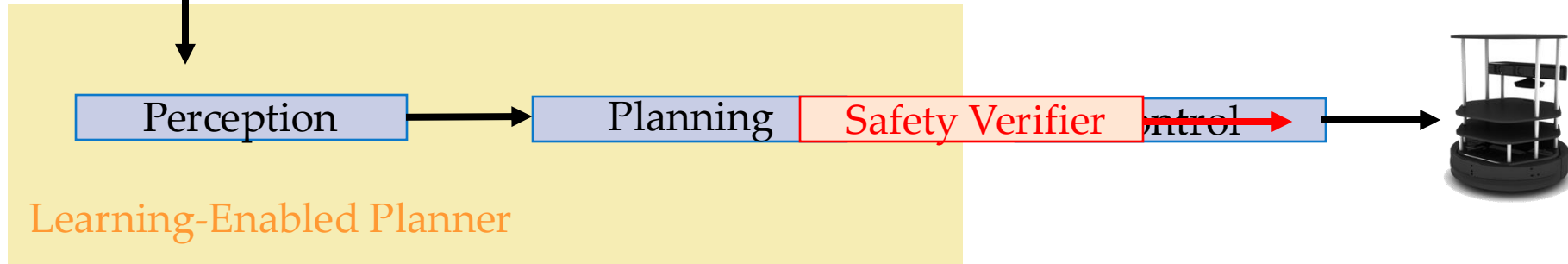
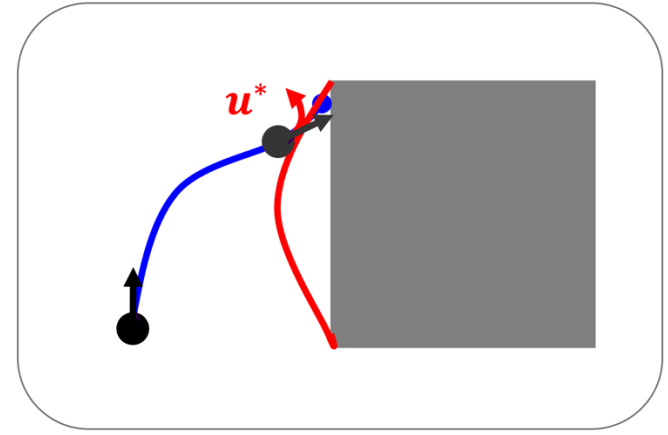


Top-Down

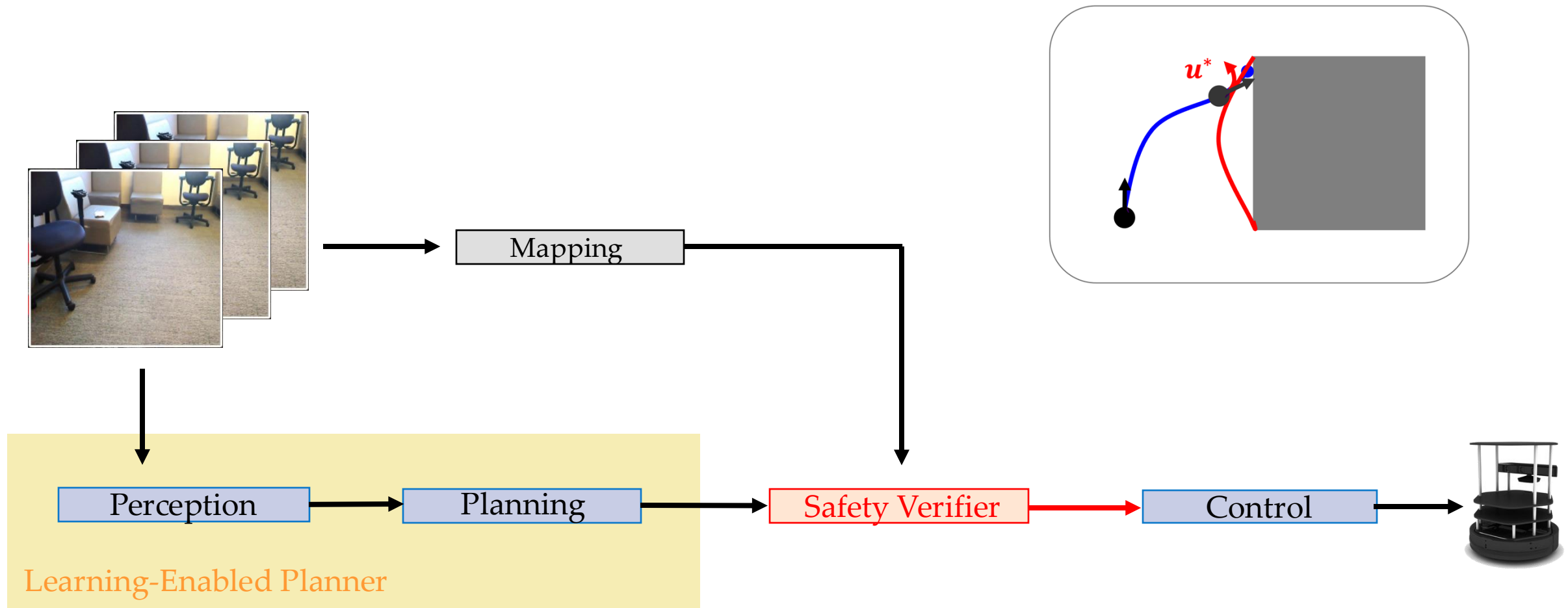


Robot POV





# LB-WayPtN Safe Autonomy Stack



# LB-WayPtNav **Safe** Autonomy Stack

With Safety Controller



Third-Person POV

Goal

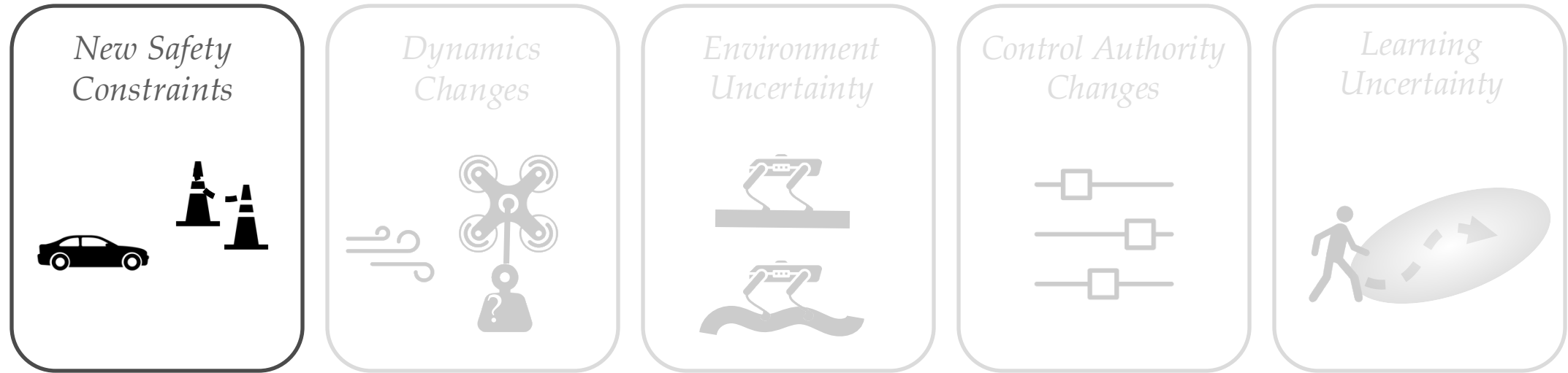


Top-Down



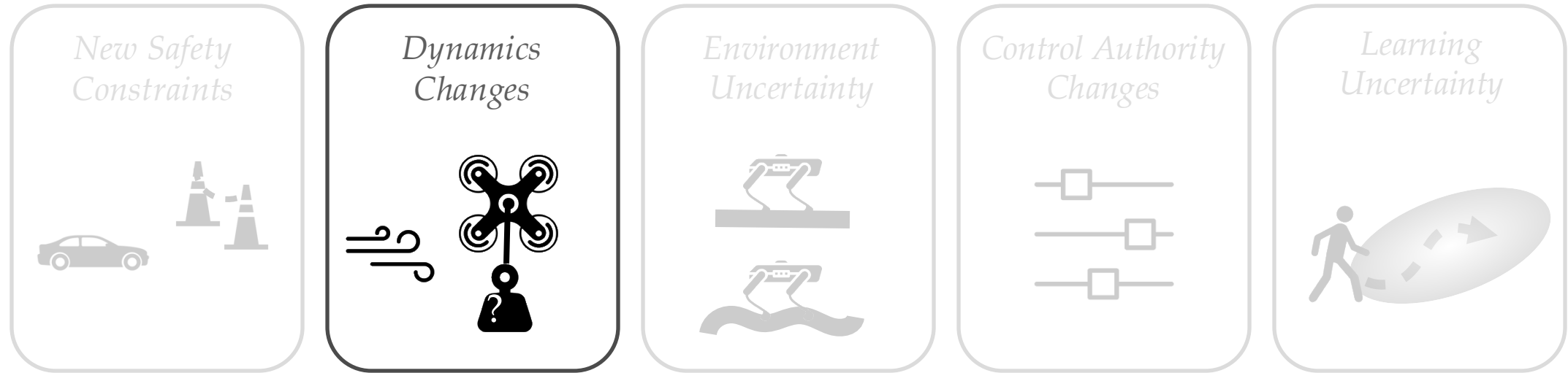
Robot POV

But at **deployment time** the robot may experience new situations



Requires adaptation of reachable sets & safety controller online!

But at **deployment time** the robot may experience new situations



Requires adaptation of reachable sets & safety controller online!

# Parameter-Conditioned Reachable Sets for Updating Safety Assurances Online

Javier Borquez<sup>1</sup>, Kensuke Nakamura<sup>2</sup>, Somil Bansal<sup>1</sup>

**Abstract**—Hamilton-Jacobi (HJ) reachability analysis is a powerful tool for analyzing the safety of autonomous systems. However, the provided safety assurances are often predicated on the assumption that once deployed, the system or its environment does not evolve. Online, however, an autonomous system might experience changes in system dynamics, control authority, external disturbances, and/or the surrounding environment, requiring updated safety assurances. Rather than restarting the safety analysis from scratch, which can be time-consuming and often intractable to perform online, we propose to compute *parameter-conditioned* reachable sets. Assuming expected system and environment changes can be parameterized, we treat these parameters as virtual states in the system and leverage recent advances in high-dimensional reachability analysis to solve the corresponding reachability problem offline. This results in a family of reachable sets that is parameterized by the environment and system factors. Online, as these factors change, the system can simply query the corresponding safety function from this family to ensure system safety, enabling a real-time update of the safety assurances. Through various simulation studies, we demonstrate the capability of our approach in maintaining system safety despite the system and environment evolution.

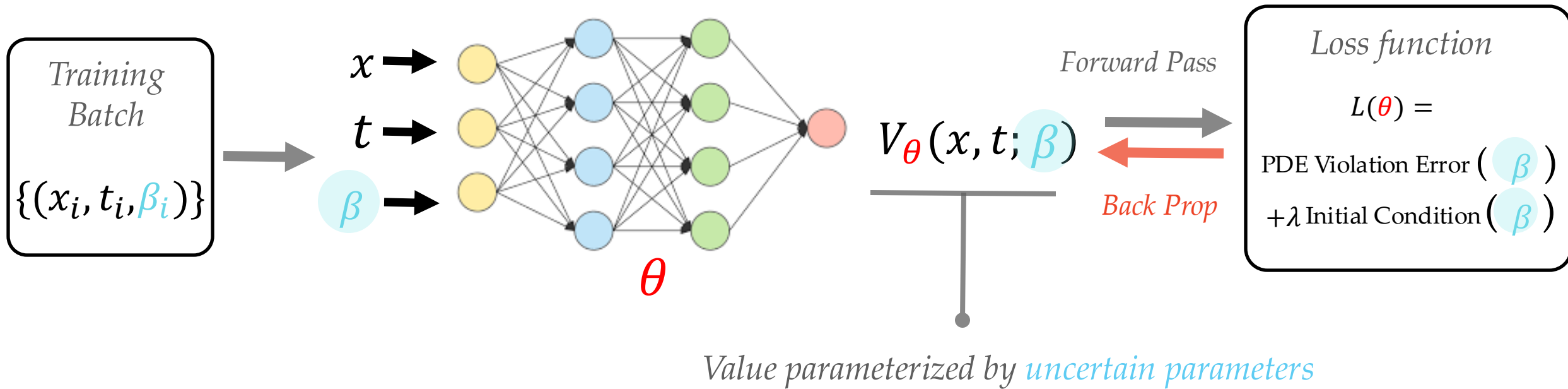
## I. INTRODUCTION

Ensuring the safe operation of autonomous systems is crucial for their successful deployment in safety-critical domains such as self-driving vehicles, unmanned aerial vehicle mobility, and human-robot interaction. These applications often require autonomous systems to operate in situations where environmental factors might change online. For example, a UAV might experience stronger wind during its

However, safety assurances are typically provided for *given* environment conditions and system dynamics. Safe motion planning methods [13]–[18] combine the above safety assurance methods with online trajectory planning to ensure safety in *a priori* unknown environments. However, these methods typically impose restrictive assumptions on the system dynamics or the environment to ensure safety. Furthermore, they often do not consider changes in system dynamics, such as changes in control authority or disturbance bounds, and require a motion planning algorithm that can operate in real-time, which itself is challenging to obtain for nonlinear systems.

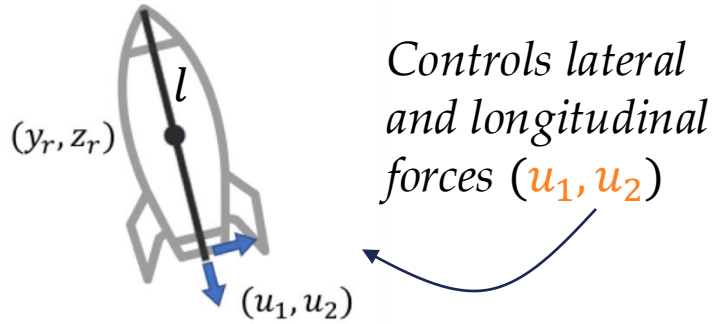
Another approach for providing safety assurances for dynamical systems is via Hamilton-Jacobi (HJ) Reachability analysis [19], [20]. Its advantages include compatibility with general nonlinear system dynamics, formal treatment of bounded disturbances, and the ability to deal with state and input constraints [10]. In reachability analysis, the system safety is characterized by *Backward Reachable Tube (BRT)*. BRT is the set of states such that the system trajectories that start from this set will eventually reach the given target set despite the worst-case disturbance (or an exogenous, adversarial input more generally). If the target set consists of those states that are known to be unsafe, then the BRT contains states which are potentially unsafe and should therefore be avoided. Along with the BRT, the reachability analysis also provides a safety controller for the system to stay outside the BRT. Given the utility of reachability analysis, several methods have been proposed to update the

# Parameter-Conditioned Safety Value Function



Parameter-conditioned safe sets can be used to adapt safety online corresponding to new conditions (with a simple query).

# Example: Rocket Landing on Floating Pad



Controls lateral and longitudinal forces  $(u_1, u_2)$

**Dynamics**  
(6D system)

$$\ddot{y} = \cos(\theta) u_1 - \sin(\theta) u_2 + d_y$$

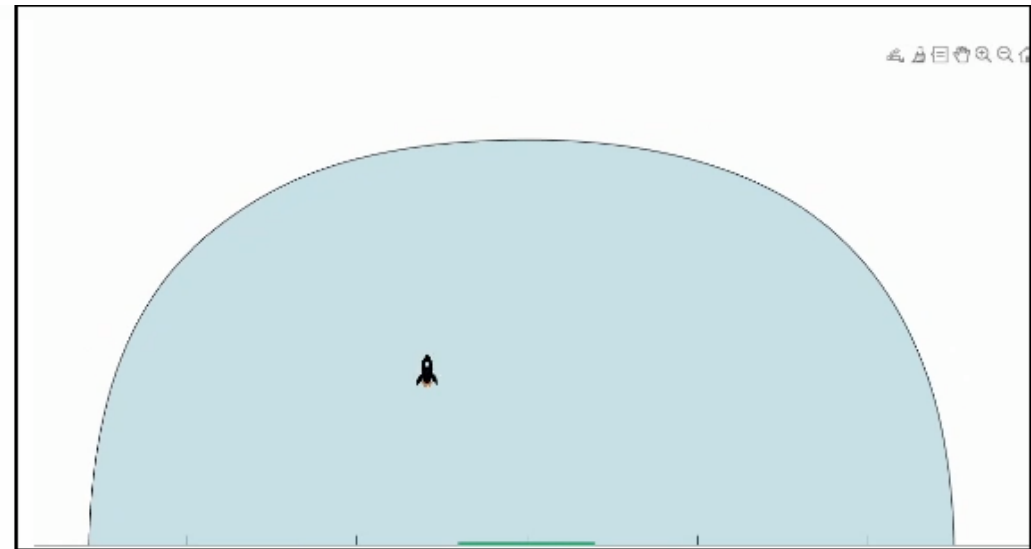
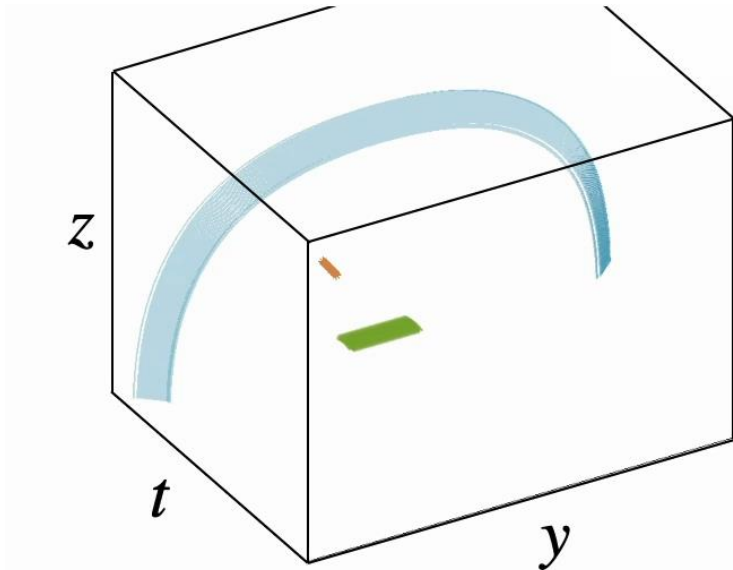
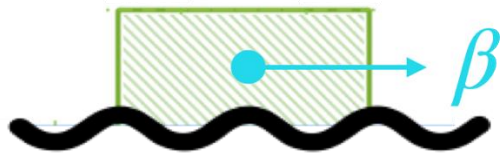
$$\ddot{z} = \sin(\theta) u_1 - \cos(\theta) u_2 - g$$

$$\ddot{\theta} = \alpha u_1 + d_\theta$$

**Parameterized**

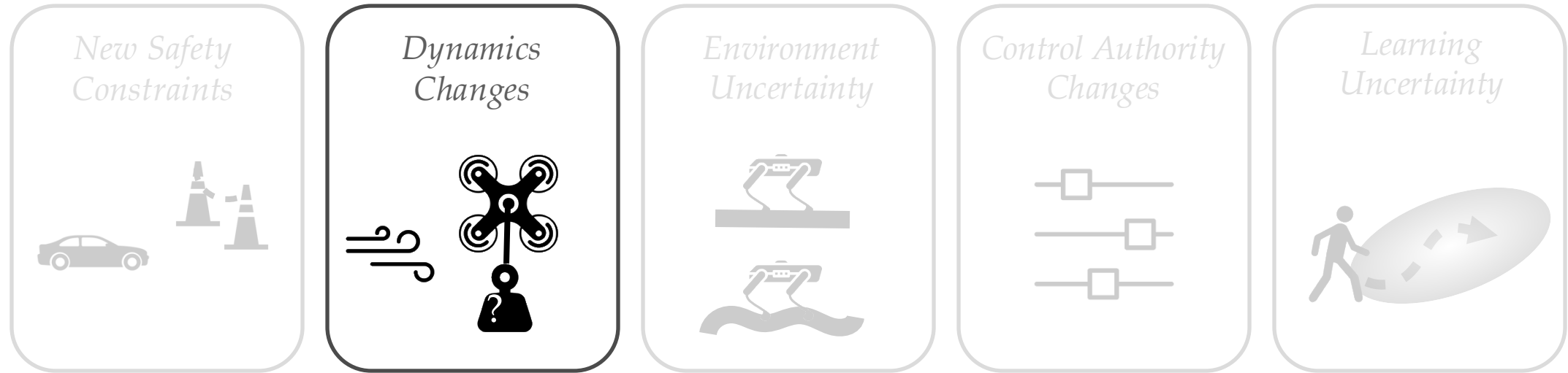
**Target Set**

$$\mathcal{L}(\beta) = \{(y, z) : |y - \beta| \leq 2l, 0 \leq z \leq 2l\}$$



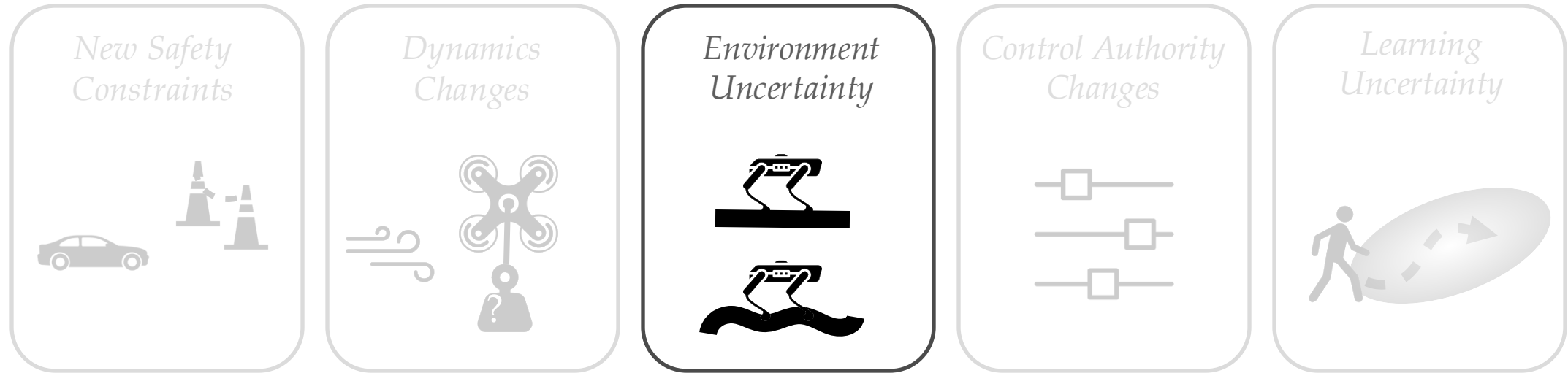


But at **deployment time** the robot may experience new situations



Requires adaptation of reachable sets & safety controller online!

But at **deployment time** the robot may experience new situations



Requires adaptation of reachable sets & safety controller online!

# One Filter to Deploy Them All: Robust Safety for Quadrupedal Navigation in Unknown Environments

Albert Lin<sup>1,2</sup>, Shuang Peng<sup>1</sup>, and Somil Bansal<sup>2</sup>

<sup>1</sup>University of Southern California <sup>2</sup>Stanford University

Project Website: [https://sia-lab-git.github.io/One\\_Filter\\_to\\_Deploy\\_Them\\_All](https://sia-lab-git.github.io/One_Filter_to_Deploy_Them_All)

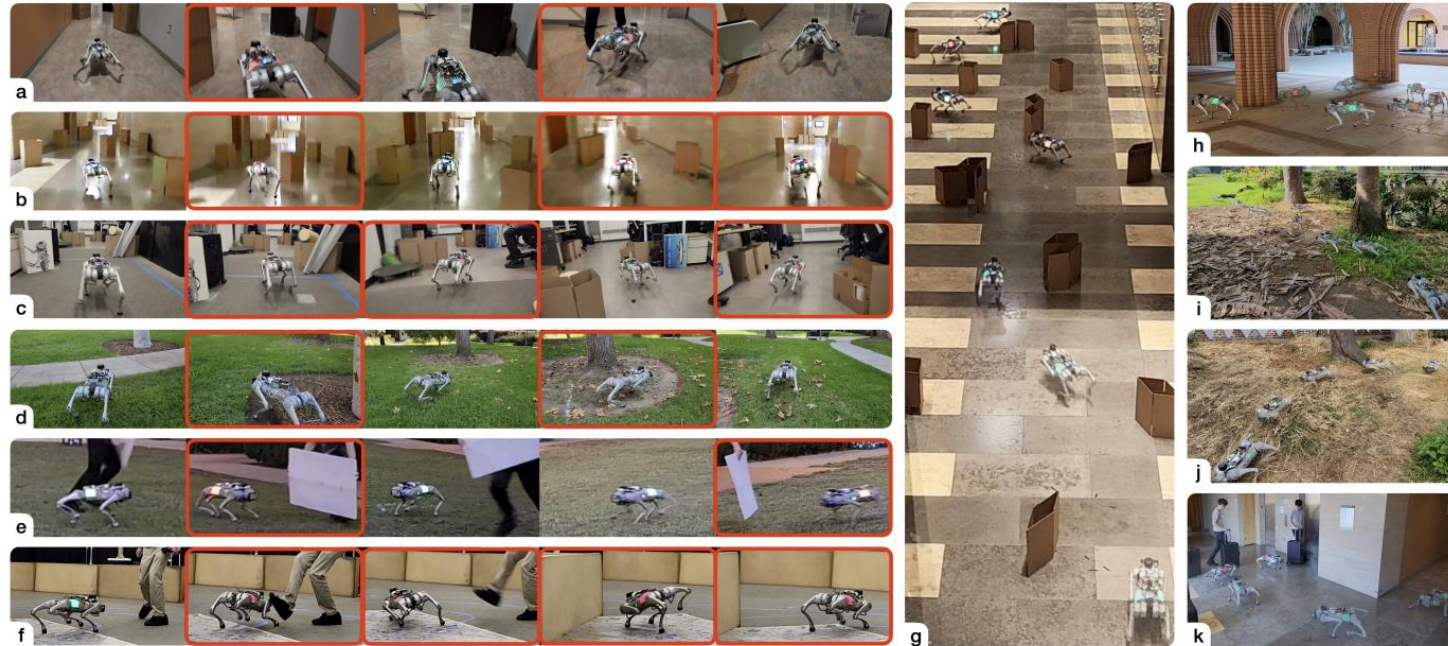
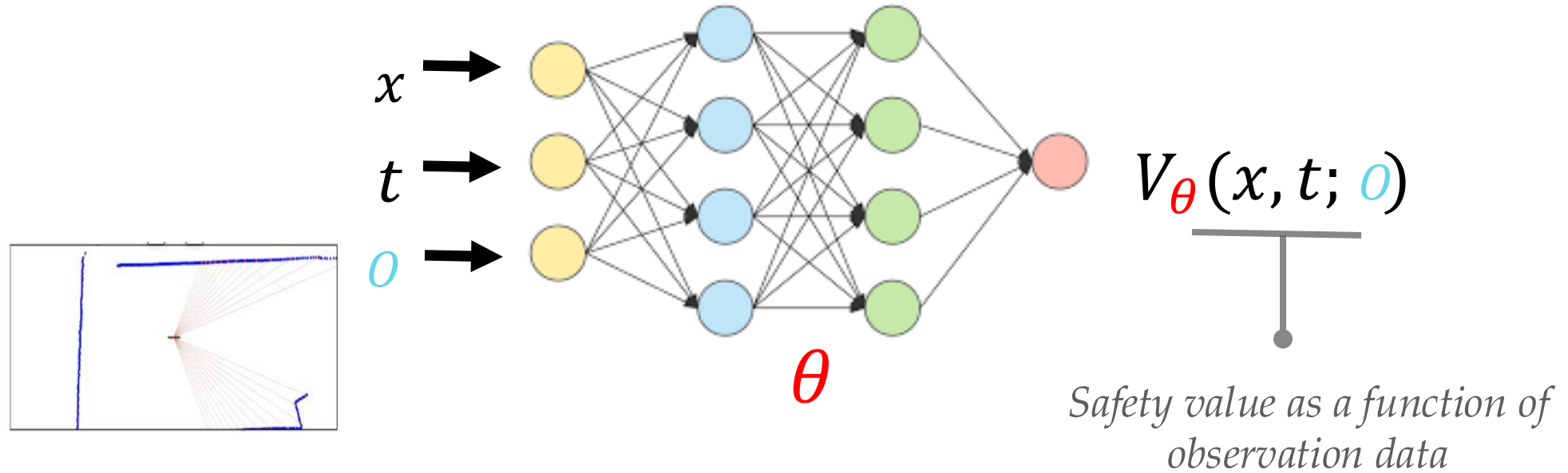


Fig. 1: Our proposed observation-conditioned reachability-based (OCR) safety-filter framework automatically safeguards different controllers in diverse settings *without a priori access to the controllers or environments*. A trained OCR value network governs the switch between nominal and **filtered** control using an onboard LiDAR sensor. The framework successfully safeguards a variety of high-level planners, including (a) learning-based, (c, f, k) model-based, (b, d, g, h, i, j) human teleoperated, and (e) naive planners, on top of different low-level locomotion policies, including (a, f, i, j, k) learning-based and (b, c, d, e, g, h) model-based policies. Safety is maintained despite (a, b, c) narrow corridors, (d, i, j) rough terrains, (e, k) dynamic obstacles, (f) external disturbances, and (h) collision-seeking human teleoperation.

**Abstract**—As learning-based methods for legged robots rapidly grow in popularity, it is important that we can provide safety assurances efficiently across different controllers and environments. Existing works either rely on *a priori* knowledge of the environ-

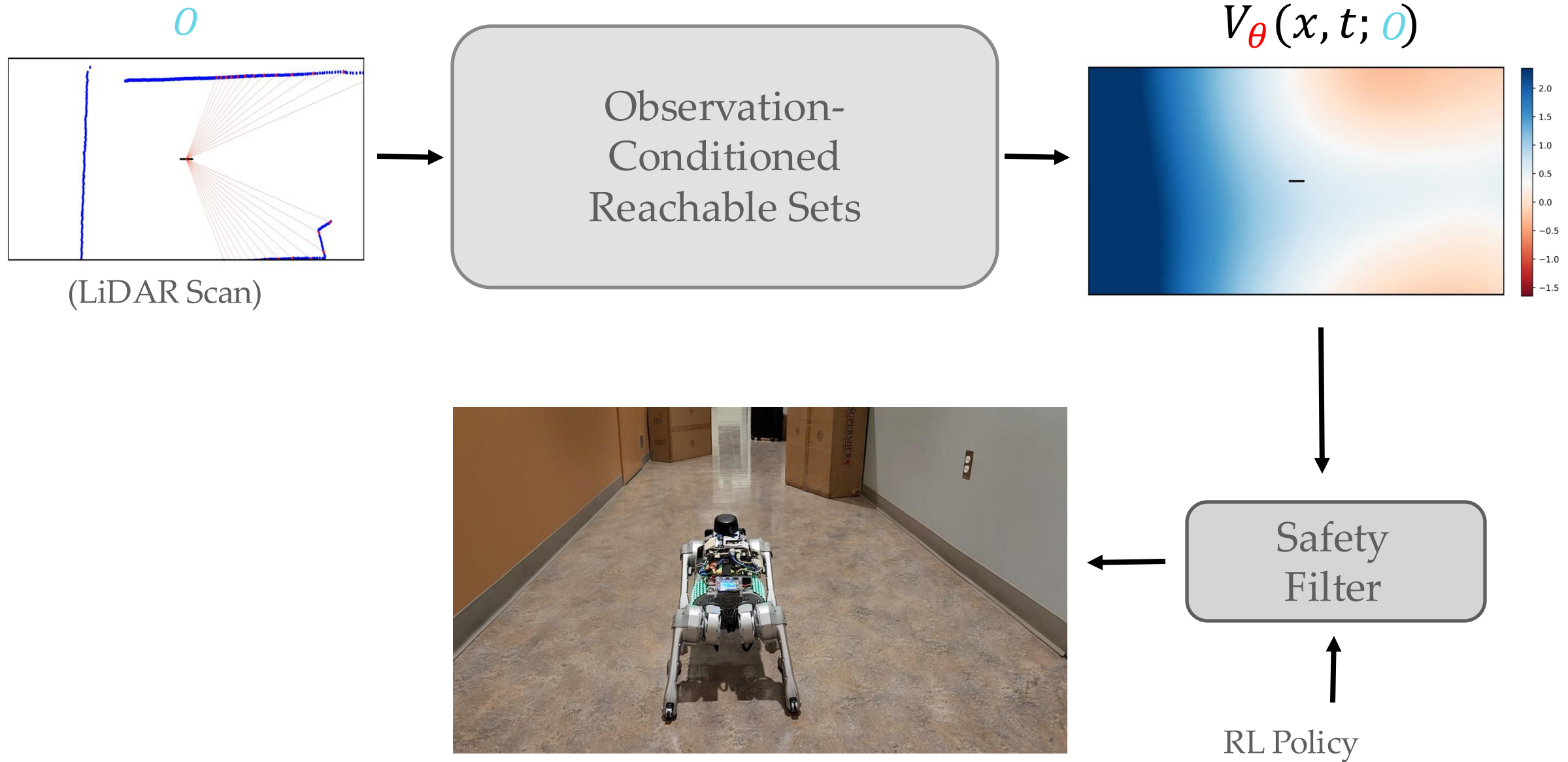
**Index Terms**—Hamilton-Jacobi reachability analysis, safety filtering, adaptive safety, robust verification, safe legged locomotion.

# Observation-Conditioned Safety Value Function

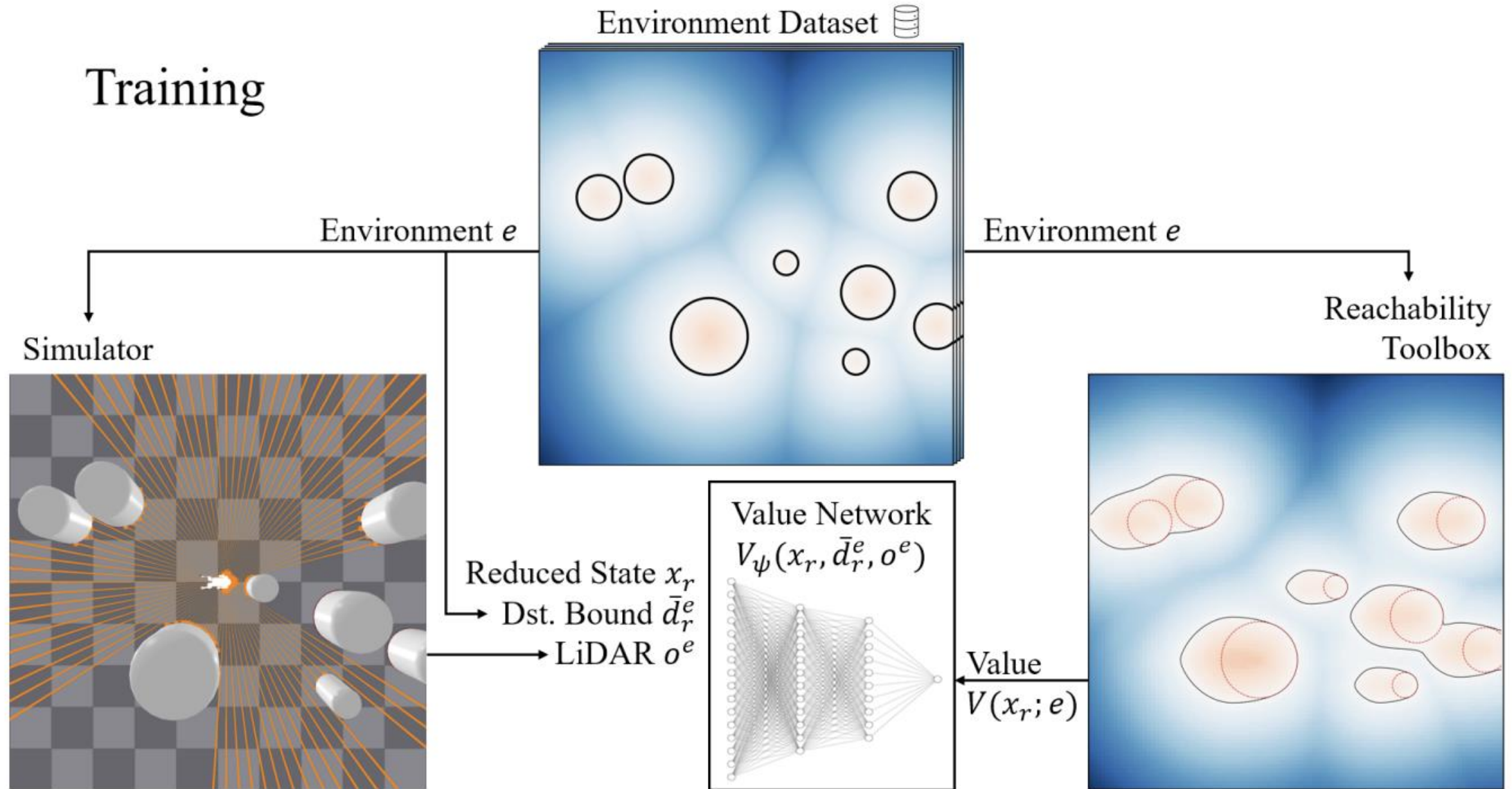


Parameterization can be more than low-D dynamics; can also be **observations!**

# Observation-Conditioned Reachability for Legged Locomotion

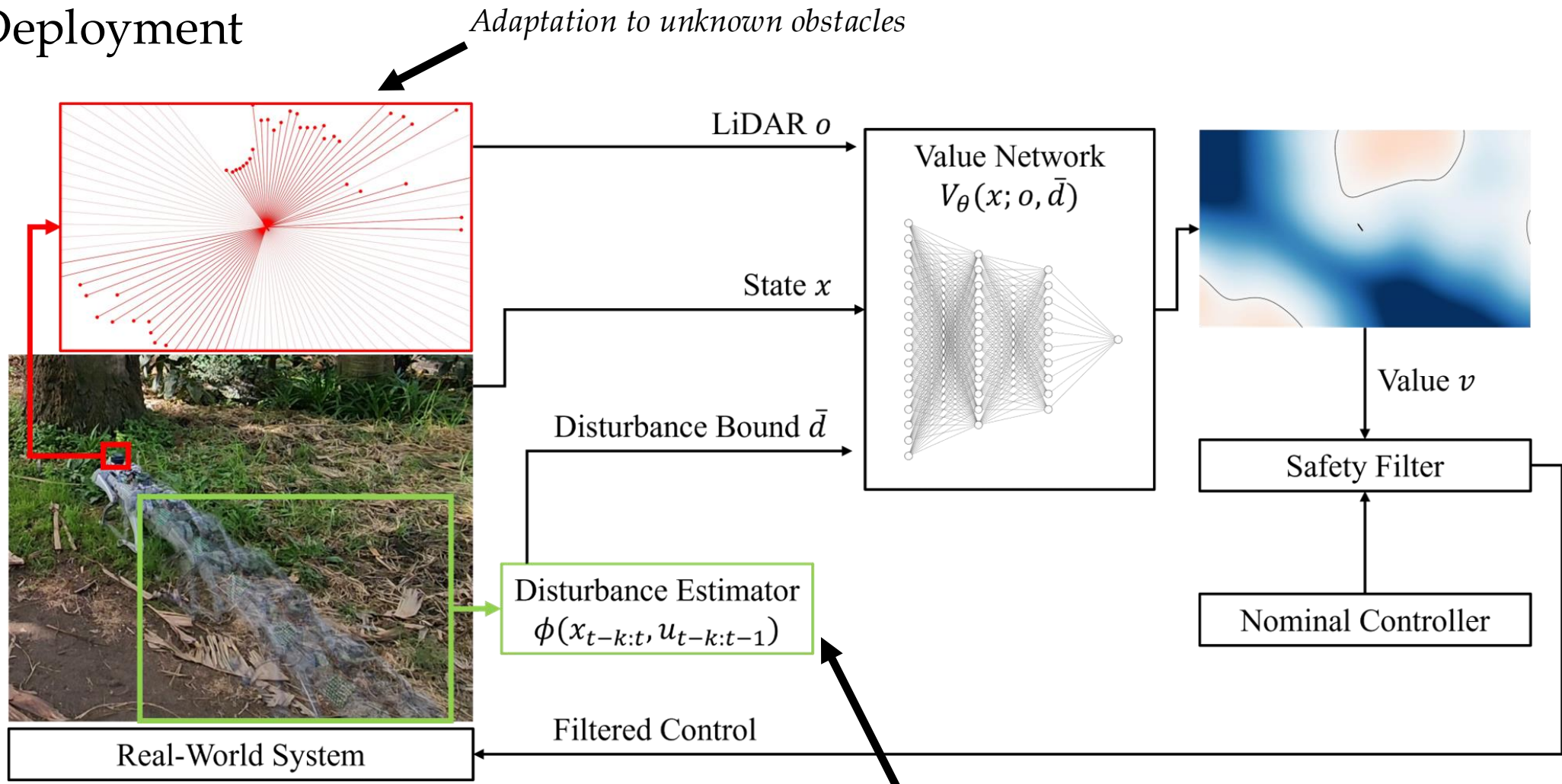


# Observation-Conditioned Reachability for Legged Locomotion



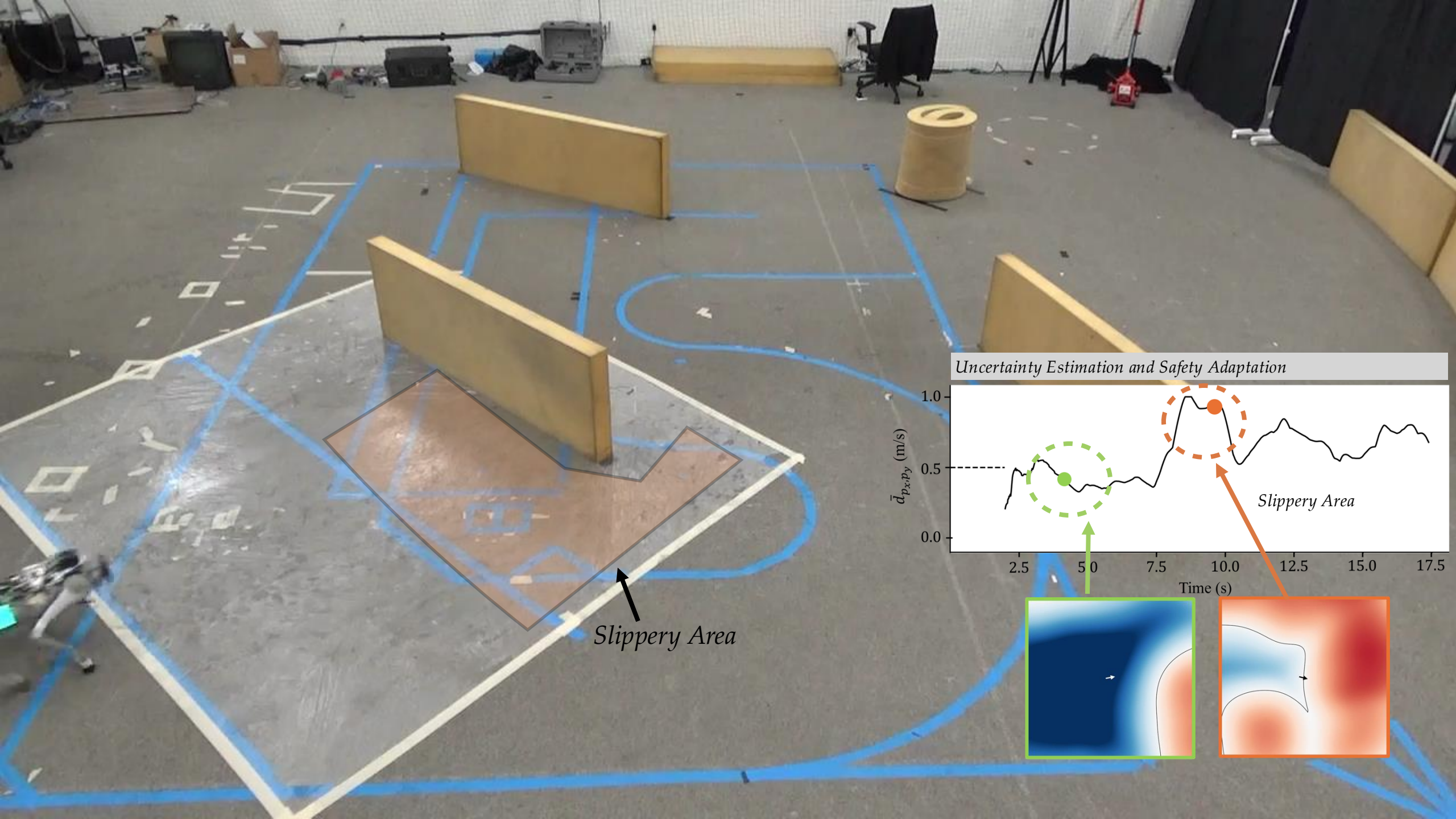
# Observation-Conditioned Reachability for Legged Locomotion

Deployment

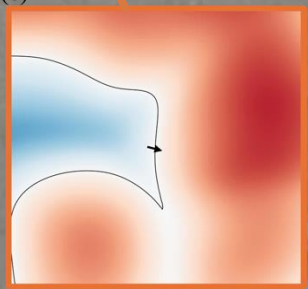
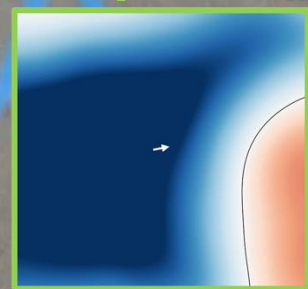
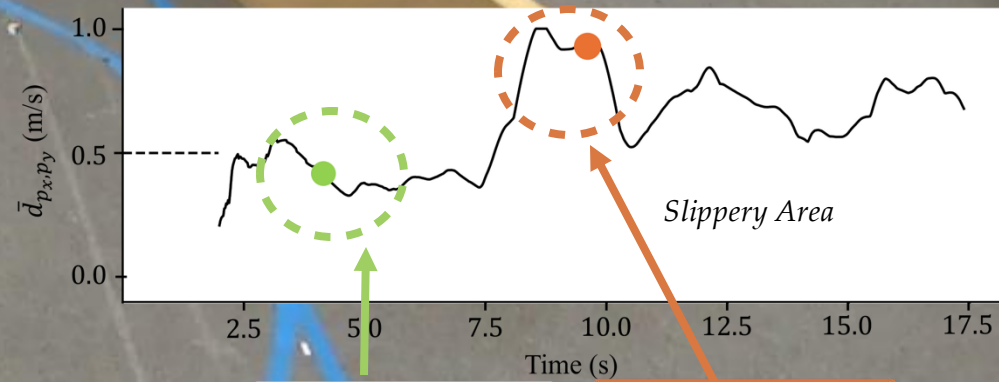


*Adaptation to unknown obstacles*

*Adaptation to dynamics uncertainty and uneven terrains*



*Uncertainty Estimation and Safety Adaptation*

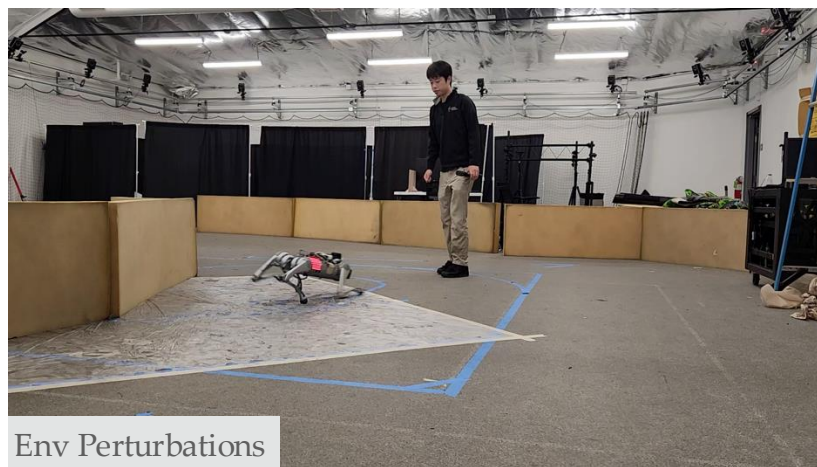
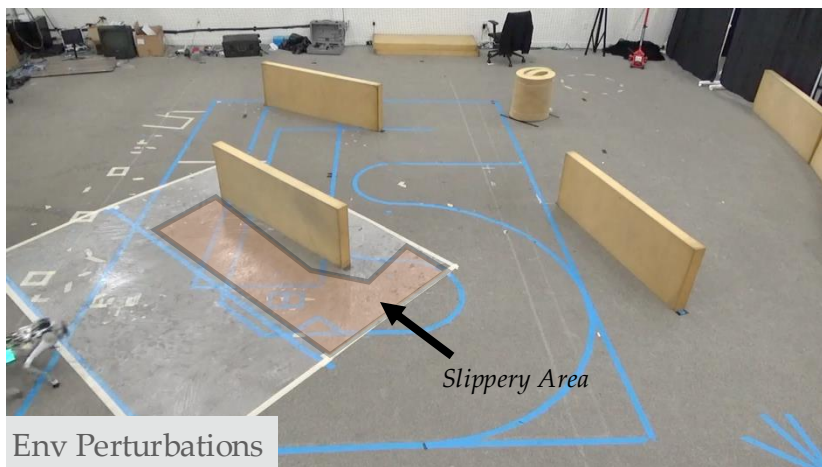
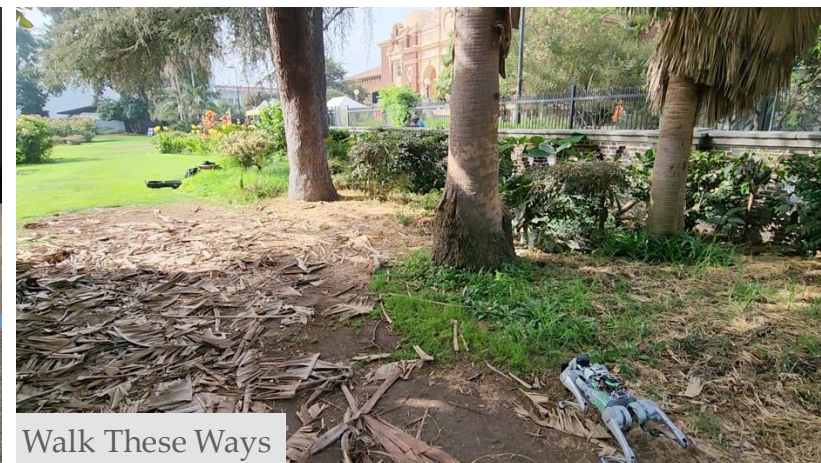
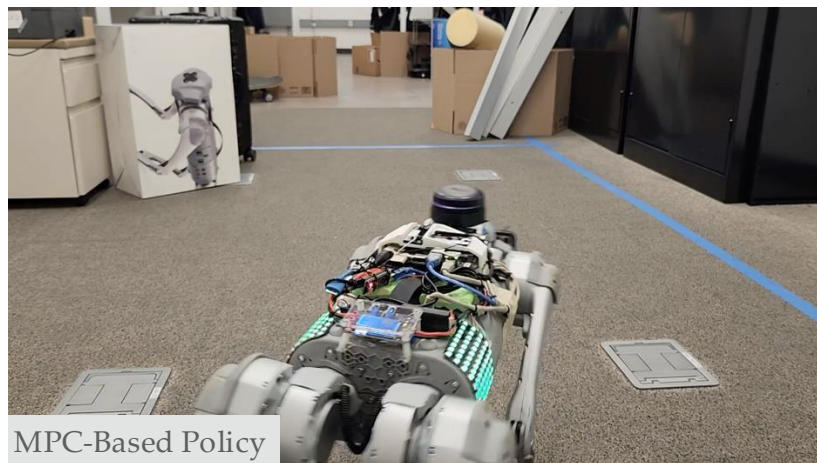


*Slippery Area*

*Slippery Area*



# “One Filter to Deploy Them All”



# Further Reading & Resources

## Adapting via Gaussian Processes

### A General Safety Framework for Learning-Based Control in Uncertain Robotic Systems

Jaime F. Fisac<sup>1</sup>, Anayo K. Akametalu<sup>1</sup>, Melanie N. Zeilinger<sup>1</sup>, Shahab Kaynama<sup>1</sup>, Jeremy Gillula<sup>1</sup>, and Claire J. Tomlin<sup>1</sup>

**Abstract**—The proven efficacy of learning-based control schemes strongly motivates their application to robotic systems operating in the physical world. However, guaranteeing correct operation during the learning process is currently an unresolved issue, which is of vital importance in safety-critical systems. We propose a general safety framework based on Hamilton-Jacobi reachability methods that can work in conjunction with an arbitrary learning algorithm. The method exploits approximate knowledge of the system dynamics to guarantee constraint satisfaction while minimally interfering with the learning process. We further introduce a Bayesian mechanism that refines the safety analysis as the system acquires new evidence, reducing initial conservativeness when appropriate while strengthening guarantees through real-time validation. The result is a least-restrictive, safety-preserving control law that intervenes only when the computed safety guarantees require it, or confidence in the computed guarantees decays in light of new observations. We prove theoretical safety

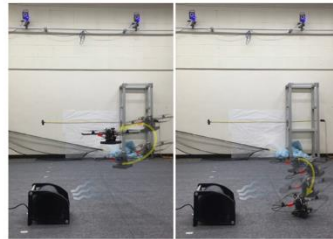


Fig. 1. Hummingbird quadrotor learning a vertical flight policy under uncertainty.

## Warm Starting

### Reachability-Based Safety Guarantees using Efficient Initializations

Sylvia L. Herbert, Shromona Ghosh, Somil Bansal, and Claire J. Tomlin

**Abstract**—Hamilton-Jacobi-Isaacs (HJI) reachability analysis is a powerful tool for analyzing the safety of autonomous systems. This analysis is computationally intensive and typically performed offline. Online, however, the autonomous system may experience changes in system dynamics, external disturbances, and/or the surrounding environment, requiring updated safety guarantees. Rather than restarting the safety analysis, we propose a method of “warm-start” reachability, which uses a user-defined initialization (typically the previously computed solution). By starting with an HJI function that is closer to the solution than the standard initialization, convergence may take fewer iterations.

In this paper we prove that warm-starting will result in guaranteed conservative solutions by over-approximating the states that must be avoided to maintain safety. We additionally prove that for many common problem formulations, warm-starting will result in exact solutions. We demonstrate our method on several illustrative examples with a double integrator, and also on a more practical example with a 10D quadrotor

for efficiently updating the computation as new information is acquired. There are some methods for speeding up this computation using decomposition [8], and there are other efficient approaches that require simplified problem formulations and/or dynamics [9–15]. The methods in [9, 16–19], can handle more complex dynamics, but may be less scalable or unable to represent complex sets. Efficient reachability analysis remains challenging for general system dynamics and problem setups.

Warm-starting in the optimization community involves using an initialization that acts as a “best guess” of the solution, and therefore may converge in fewer iterations (if convergence can be achieved). Recent work applied this warm-starting idea to create a “discounted reachability” formulation for infinite-time horizon problems [20, 21]. By using a discount factor, this formulation guarantees con-

## Parameterized Reachability

### Safety Assurances for Human-Robot Interaction via Confidence-aware Game-theoretic Human Models

Ran Tian<sup>\*</sup>, Liting Sun<sup>\*</sup>, Andrea Bajcsy<sup>\*</sup>, Masayoshi Tomizuka, and Anca D. Dragan

**Abstract**—An outstanding challenge with safety methods for human-robot interaction is reducing their conservatism while maintaining robustness to variations in human behavior. In this work, we propose that robots use confidence-aware game-theoretic models of human behavior when assessing the safety of a human-robot interaction. By treating the influence between the human and robot as well as the human’s rationality as unobserved latent states, we succinctly infer the degree to which a human is following the game-theoretic interaction model. We leverage this model to restrict the set of feasible human controls during safety verification, enabling the robot to confidently modulate the conservatism of its safety monitor online. Evaluations in simulated human-robot scenarios and ablation studies demonstrate that imbuing safety monitors with confidence-aware game-theoretic models enables both safe and efficient human-robot interaction. Moreover, evaluations with real traffic data show that our safety monitor is less conservative than traditional safety methods in real human driving scenarios.

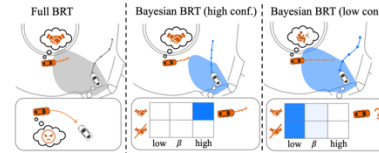


Fig. 1: Robot car (white) merges into a round-about with a nearby human-driven car (orange). (left) Human accommodates for robot, but robot is overly conservative and protects against the full backwards reachable tube (BRT). (center) Our Bayesian BRT infers how the human is influenced by the robot and shrinks the set of unsafe states. (right) When the human does not behave according to the model, the robot

### Parameter-Conditioned Reachable Sets for Updating Safety Assurances Online

Javier Borquez<sup>1</sup>, Kensuke Nakamura<sup>2</sup>, Somil Bansal<sup>1</sup>

**Abstract**—Hamilton-Jacobi (HJ) reachability analysis is a powerful tool for analyzing the safety of autonomous systems. However, the provided safety assurances are often predicated on the assumption that once deployed, the system or its environment does not evolve. Online, however, an autonomous system might experience changes in system dynamics, control authority, external disturbances, and/or the surrounding environment, requiring updated safety assurances. Rather than restarting the safety analysis from scratch, which can be time-consuming and often intractable to perform online, we propose to compute parameter-conditioned reachable sets. Assuming expected system and environment changes can be parameterized, we treat these parameters as virtual states in the system and leverage recent advances in high-dimensional reachability analysis to solve the corresponding reachability problem offline. This results in a family of reachable sets that is parameterized by the environment and system factors. Online, as these factors

However, safety assurances are typically provided for given environment conditions and system dynamics. Safe motion planning methods [13]–[18] combine the above safety assurance methods with online trajectory planning to ensure safety in a priori unknown environments. However, these methods typically impose restrictive assumptions on the system dynamics or the environment to ensure safety. Furthermore, they often do not consider changes in system dynamics, such as changes in control authority or disturbance bounds, and require a motion planning algorithm that can operate in real-time, which itself is challenging to obtain for nonlinear systems.

Another approach for providing safety assurances for dynamical systems is via Hamilton-Jacobi (HJ) Reachability

## Local Updates to Safety Value

### An Efficient Reachability-Based Framework for Provably Safe Autonomous Navigation in Unknown Environments

Andrea Bajcsy<sup>\*</sup>, Somil Bansal<sup>\*</sup>, Eli Bronstein, Varun Tolani, Claire J. Tomlin

**Abstract**—Real-world autonomous vehicles often operate in a priori unknown environments. Since most of these systems are safety-critical, it is important to ensure they operate safely in the face of environment uncertainty, such as unseen obstacles. Current safety analysis tools enable autonomous systems to reason about safety given full information about the state of the environment a priori. However, these tools do not scale well to scenarios where the environment is being sensed in real time, such as during navigation tasks. In this work, we propose a novel, real-time safety analysis method based on Hamilton-Jacobi reachability that provides strong safety guarantees despite environment uncertainty. Our safety method is planner-agnostic and provides guarantees for a variety of mapping sensors. We demonstrate our approach in simulation and in hardware to provide safety guarantees around a state-of-the-art vision-based, learning-based planner. Videos of our approach and experiments are available on the project website<sup>1</sup>.

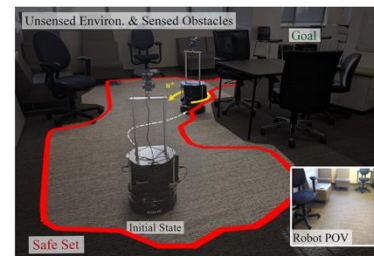


Fig. 1. Overview: We consider the problem of safe navigation from an

## One Filter to Deploy Them All: Robust Safety for Quadrupedal Navigation in Unknown Environments

Albert Lin<sup>1,2</sup>, Shuang Peng<sup>1</sup>, and Somil Bansal<sup>2</sup>

<sup>1</sup>University of Southern California <sup>2</sup>Stanford University

Project Website: [https://sia-lab-git.github.io/One\\_Filter\\_to\\_Deploy\\_Them\\_All](https://sia-lab-git.github.io/One_Filter_to_Deploy_Them_All)

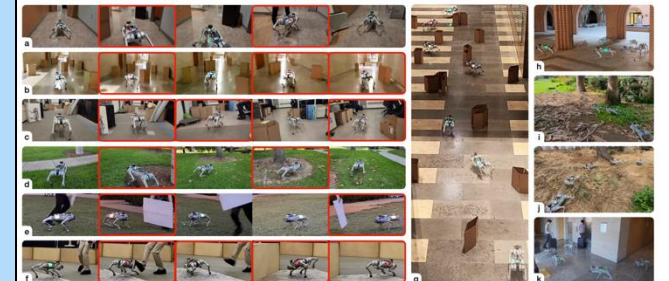


Fig. 1: Our proposed observation-conditioned reachability-based (OCR) safety-filter framework automatically safeguards different controllers in diverse settings without a priori access to the controllers or environments. A trained OCR value network governs the switch between nominal and filtered control using an onboard LiDAR sensor. The framework successfully safeguards a variety of high-level planners, including (a) learning-based, (b, d, g, h, i, j) human teleoperated, and (e) naive planners, on top of different low-level locomotion policies, including (a, f, i, j, k) learning-based and (b, c, d, e, g, h) model-based policies. Safety is maintained despite (a, b, c) narrow corridors, (d, i, j) rough terrains, (e, k) dynamic obstacles, (f) external disturbances, and (h) collision-seeking human teleoperation.

**Abstract**—As learning-based methods for legged robots rapidly grow in popularity, it is important that we can provide safety assurances efficiently across different controllers and environments. Existing works either rely on a priori knowledge of the environ-

ment or require restrictive assumptions on the system dynamics. In this work, we propose a novel, real-time safety analysis method based on Hamilton-Jacobi reachability that provides strong safety guarantees despite environment uncertainty. Our safety method is planner-agnostic and provides guarantees for a variety of mapping sensors. We demonstrate our approach in simulation and in hardware to provide safety guarantees around a state-of-the-art vision-based, learning-based planner. Videos of our approach and experiments are available on the project website<sup>1</sup>.