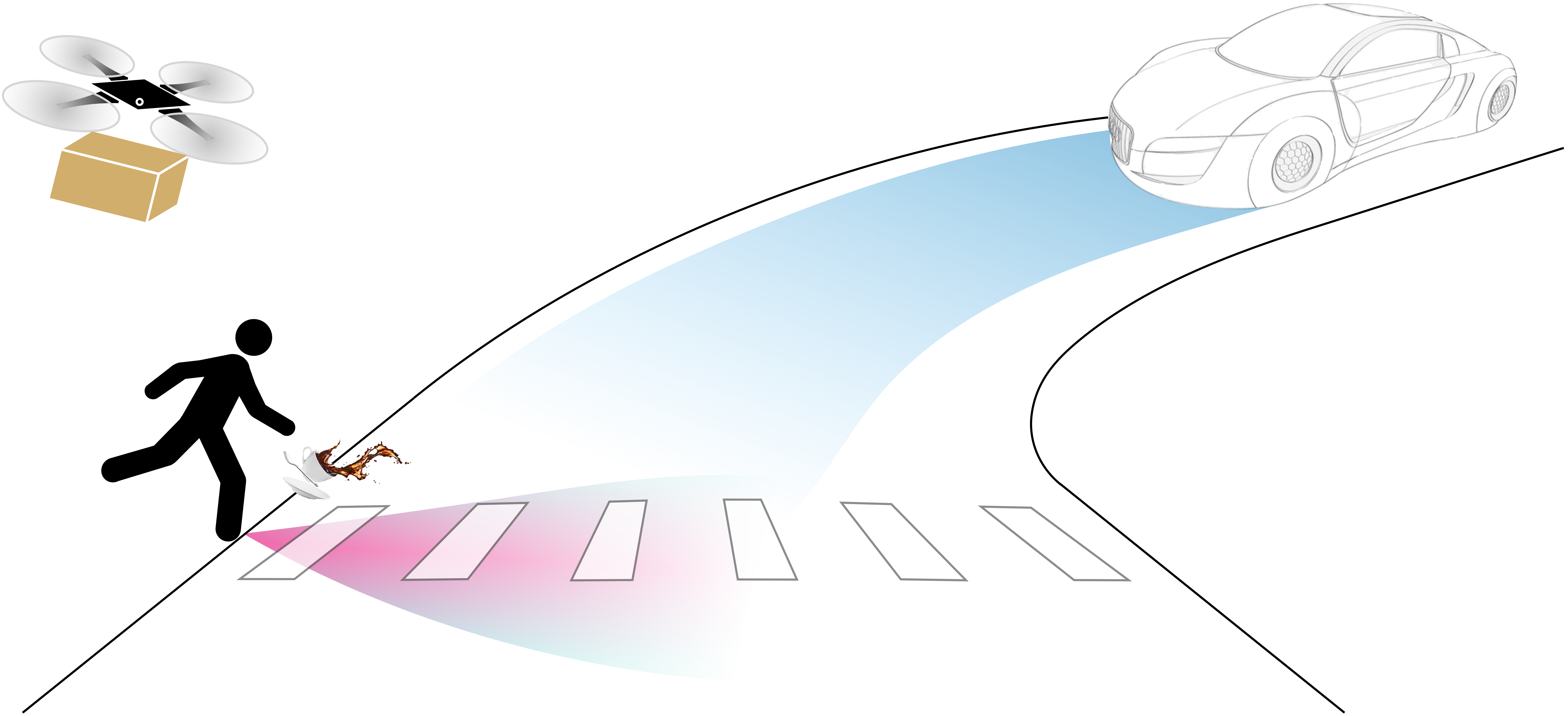


SAFE REINFORCEMENT LEARNING

PRINCETON
Electrical and Computer Engineering

Jaime Fernández Fisac

SAFE ROBOTICS LABORATORY

A Little About Me

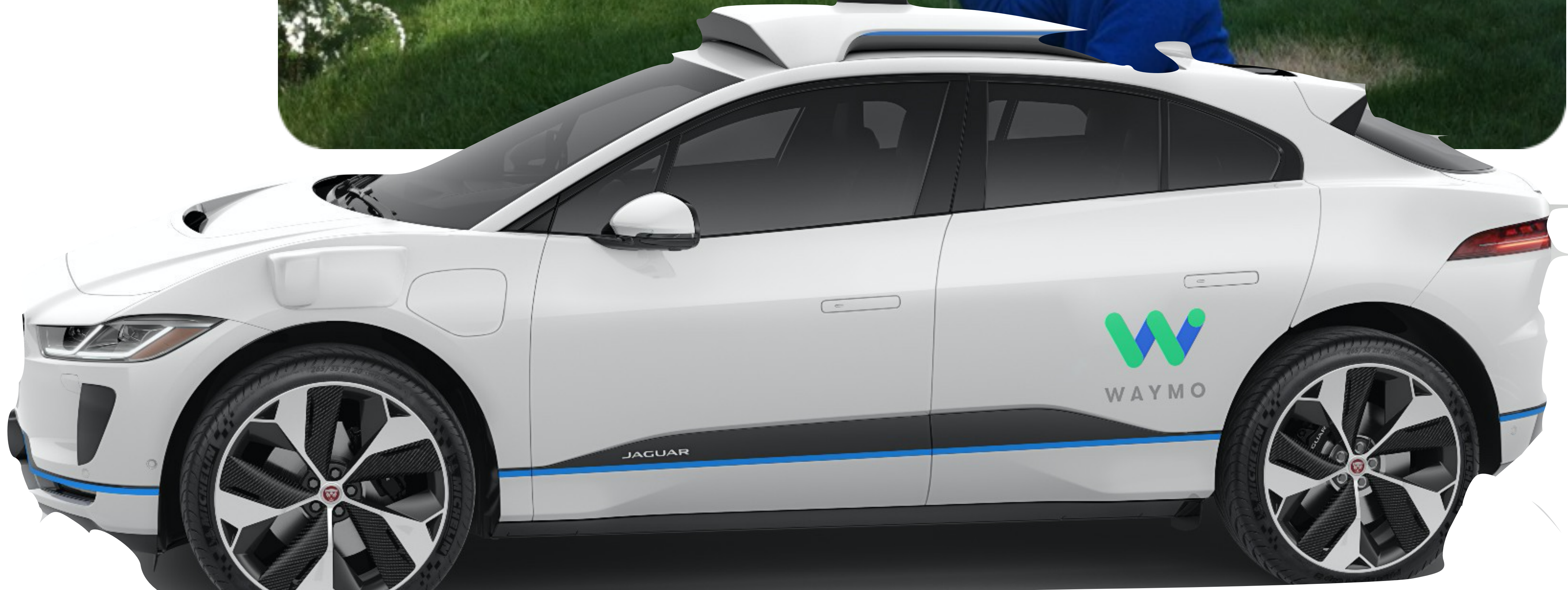
I’m an Assistant Professor at Princeton

Grew up and went to college in Spain

Got interested in *robot safety* while working with drones

PhD at UC Berkeley → Research Scientist at Waymo

same lab as Vicenç!



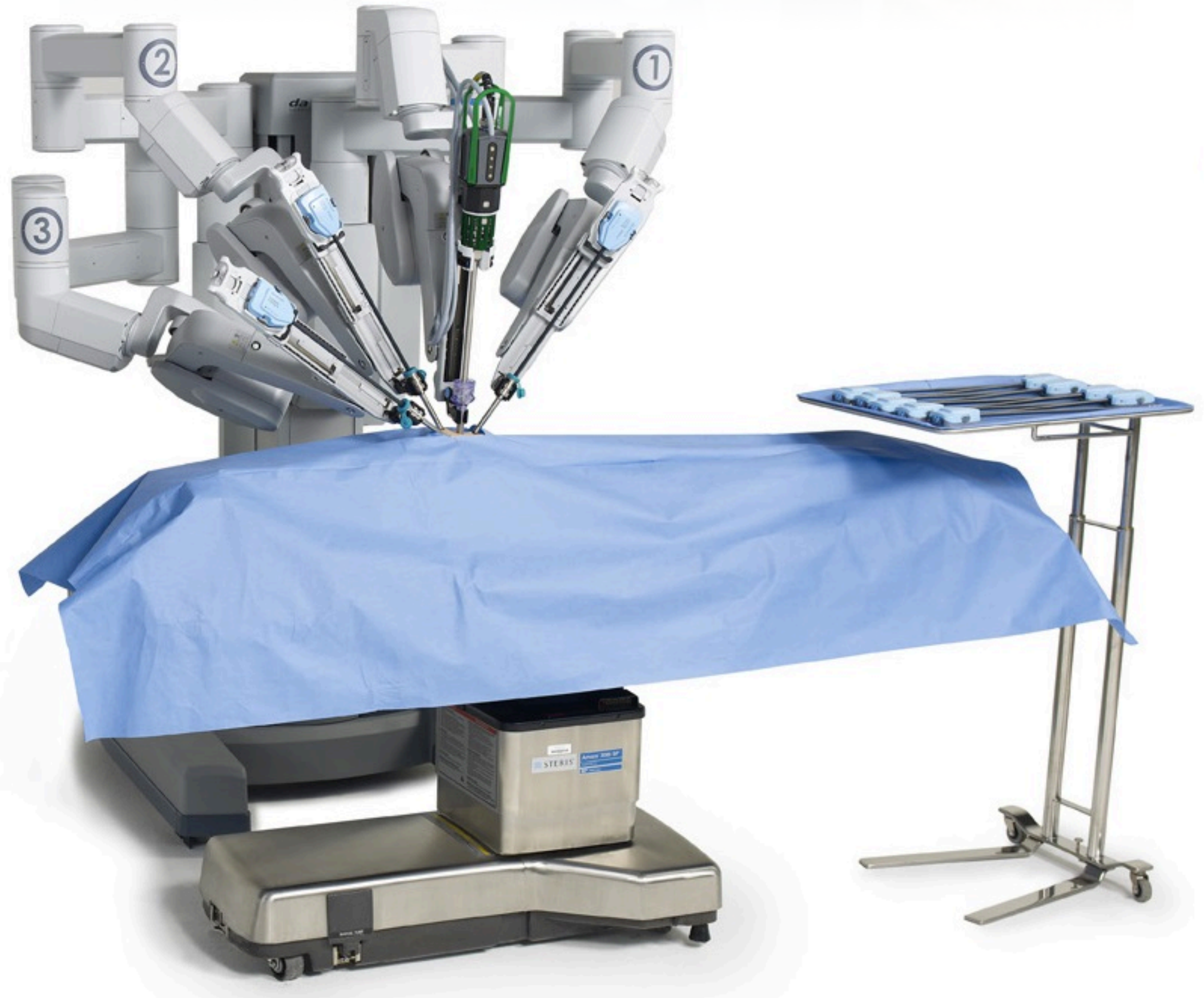
The Safe Robotics Lab

Goal: enable robots to operate safely around people

Theory + algorithms for active safety under uncertainty

Focus on changing & interactive environments

Control theory + AI + game theory + cognitive science




zipline

Why Have Robots Learn?

Robotics has existed for decades as a field primarily focused on industrial processes.

Repetitive (though complex) tasks

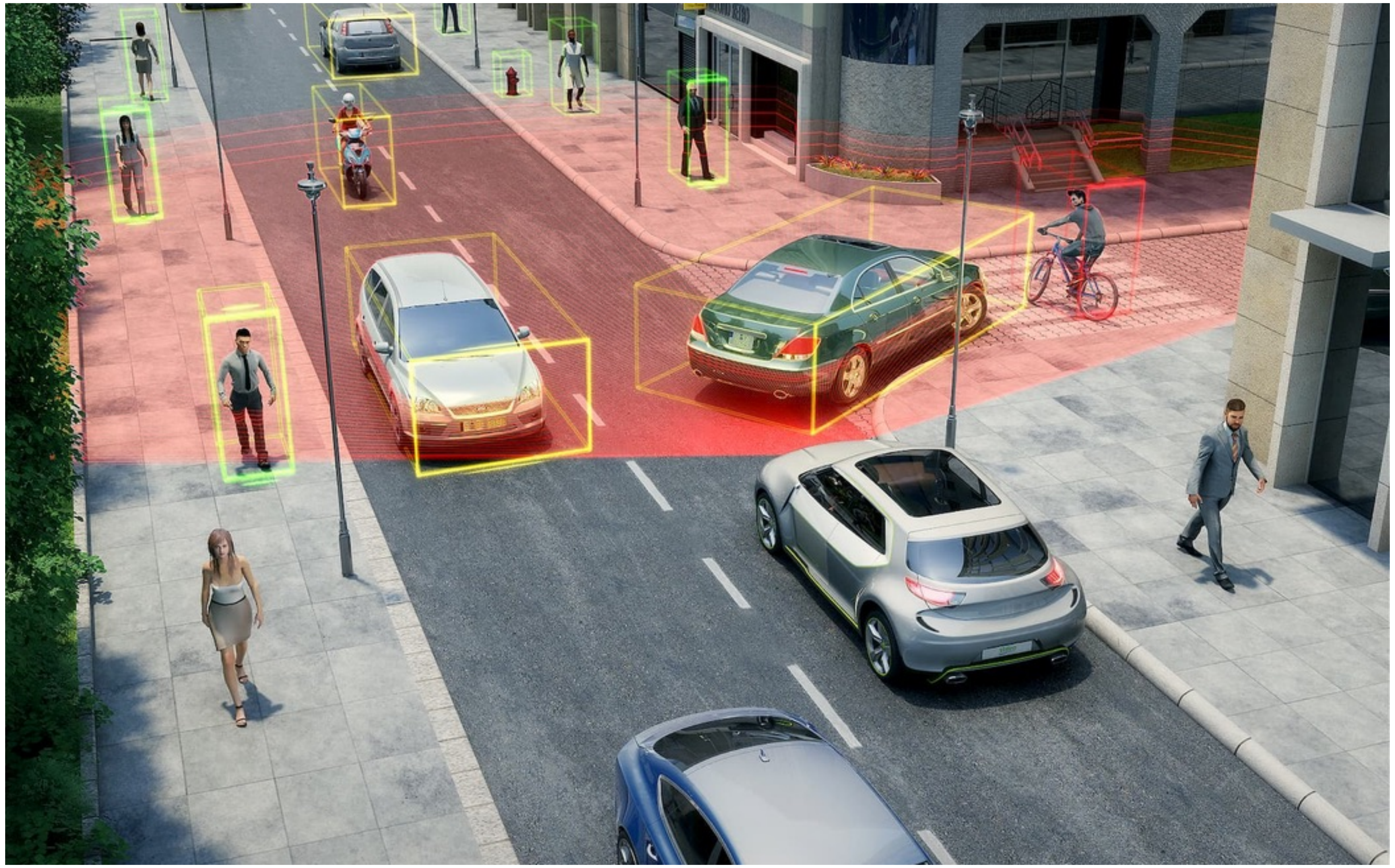
Structured environments

Well-understood dynamics

Isolated from humans

Why Have Robots Learn?

Advances in sensing, decision-making and control → new opportunities extend beyond factories



~~Repetitive (though complex) tasks~~

~~Structured environments~~

~~Well-understood dynamics~~

~~Isolated from humans~~

Mitigating Uncertainty Through Adaptation

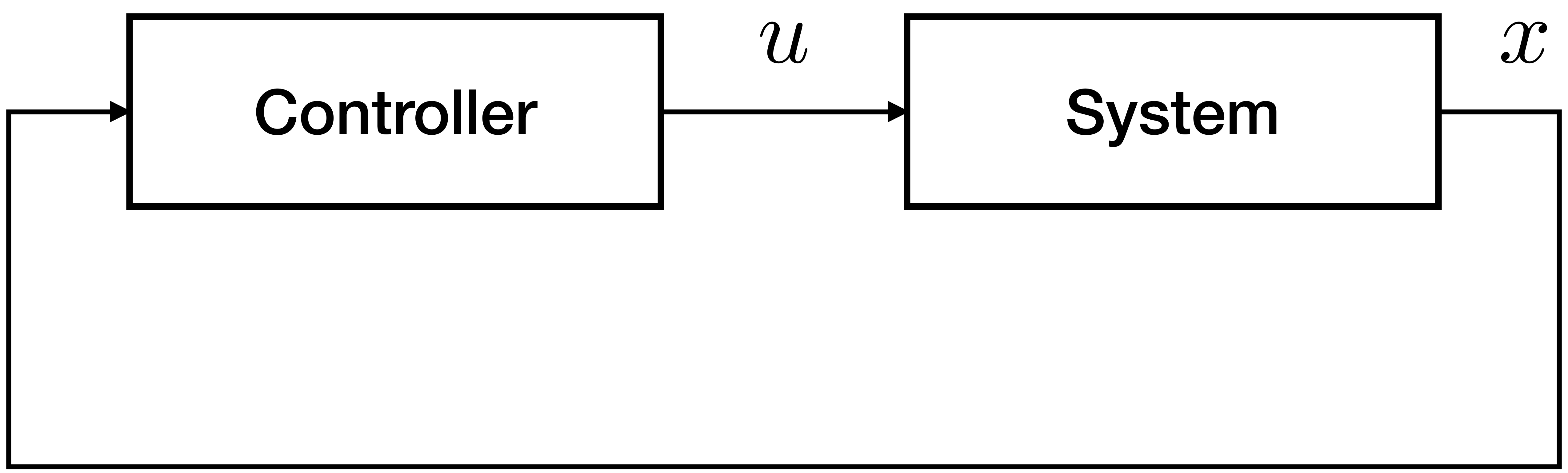
~~Design a fixed control policy that works in all conditions the system may find.~~

Adjust the variable control policy to the specific conditions encountered.

Exploit new information as it comes in.

Maintain consistent performance in the presence of changing conditions.

Optimal Control (1950s–60s)

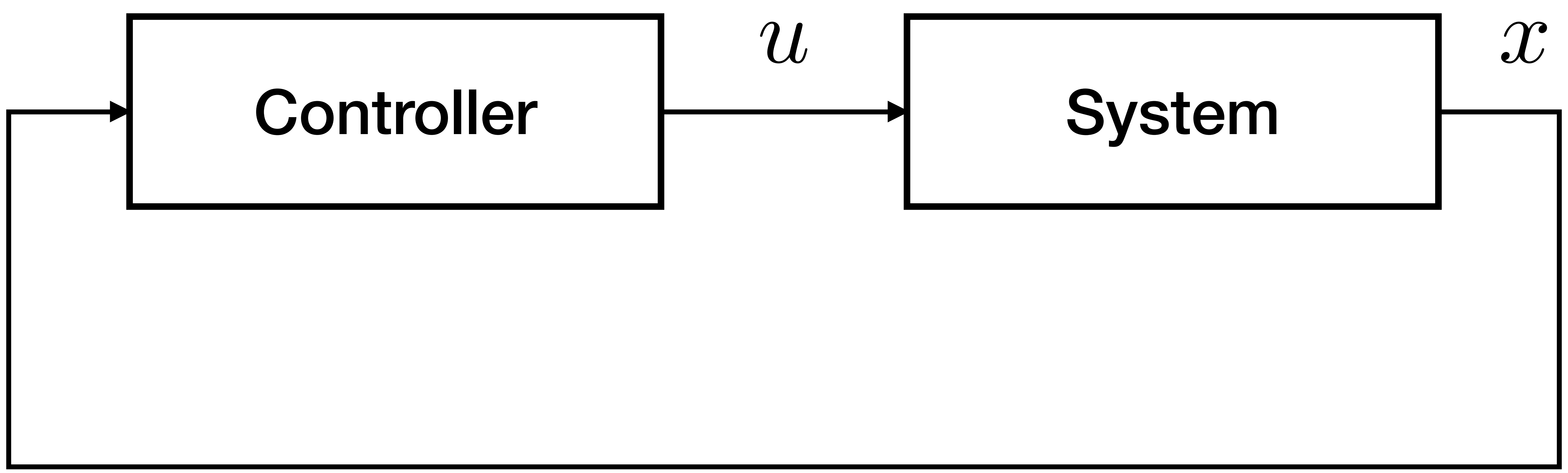


$$\max_{\theta} \mathbb{E}_{\mathbf{d}} \; J\left(\mathbf{x}_{x,t}^{\pi_\theta,\mathbf{d}}\right)$$

Performance Objective

Can we adjust the control policy π_θ over time based on the robot's experience?

Adaptive Control (1960s–70s)

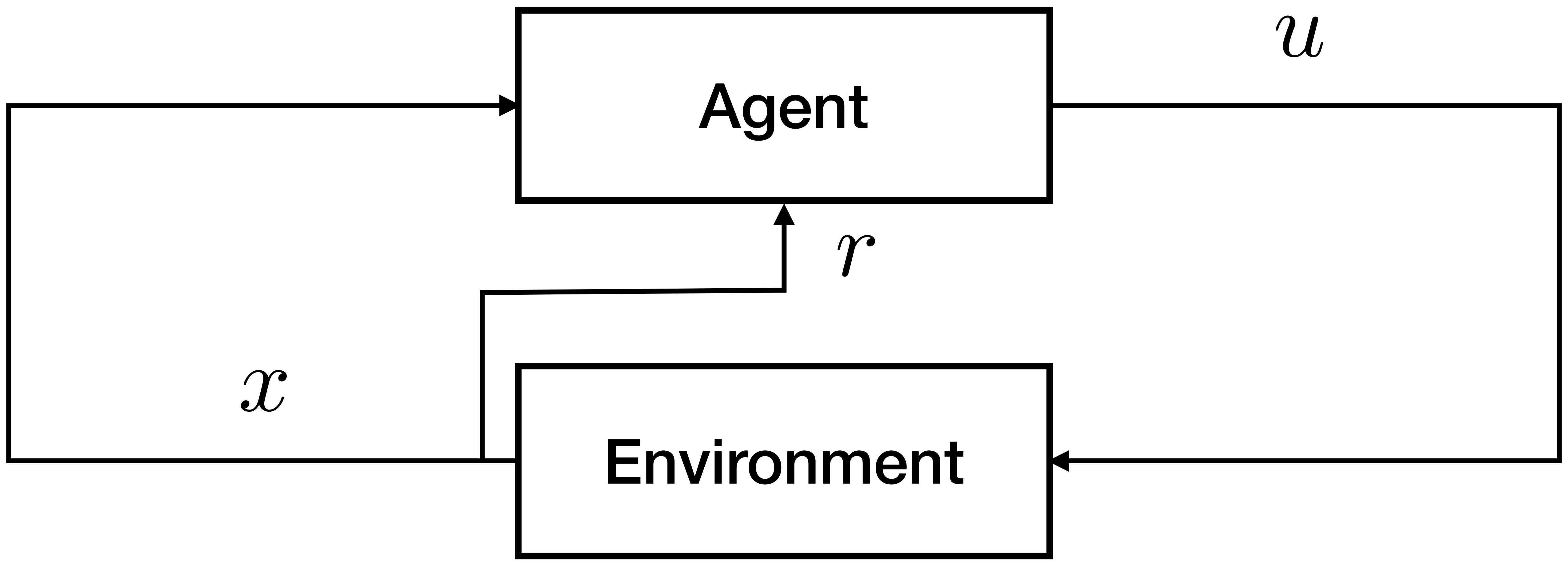


$$\max_{\theta} \mathbb{E}_{\mathbf{d}} \; J\left(\mathbf{x}_{x,t}^{\pi_\theta, \mathbf{d}}\right)$$

Performance Objective

Can we adjust the control policy π_θ over time based on the robot's experience?

Reinforcement Learning

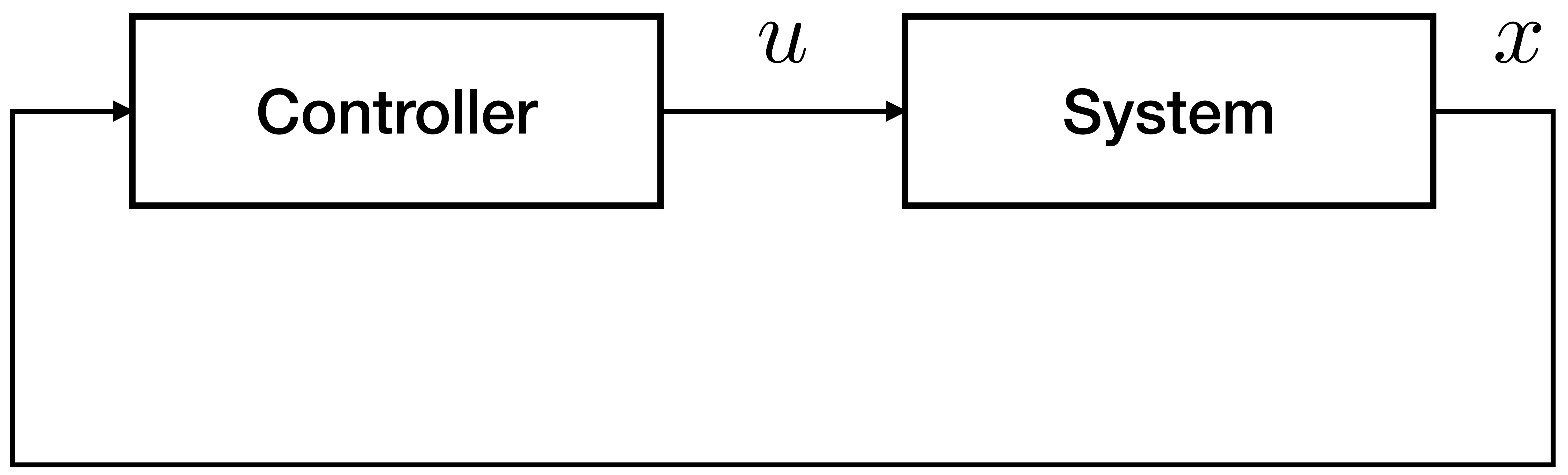


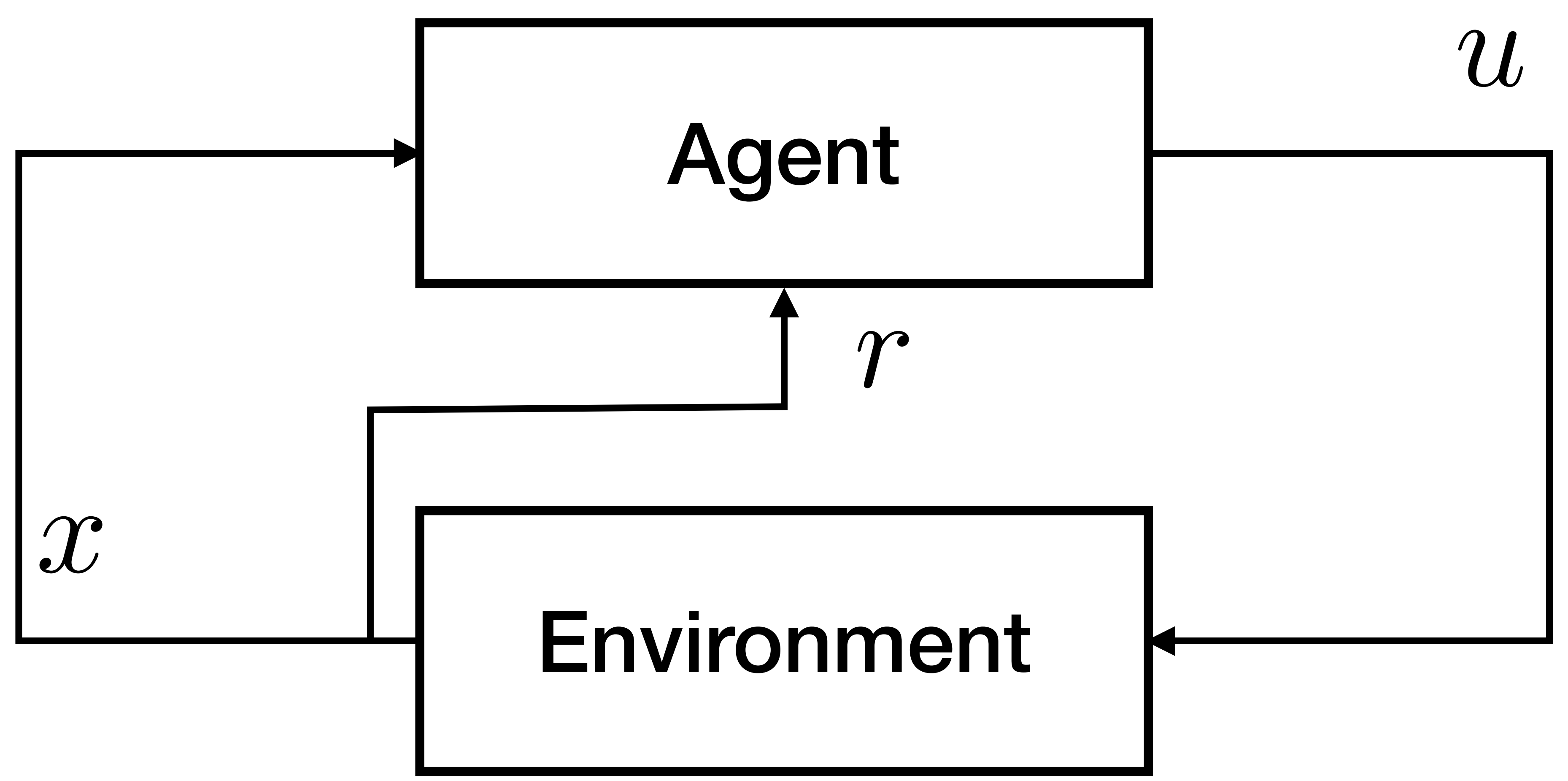
$$\max_{\theta} \mathbb{E}_{\mathbf{d}} \sum_{t=0}^{\infty} \gamma^t r_t$$

Time discount factor

Additive/Average
Performance Objective

Learning-Based Control





Adaptive Control

Continual operation (single shot)

Typical objectives: stability, tracking error

World model: dynamical system

Reinforcement Learning

Episodic operation (many trials)

Typical objectives: accrued reward

World model: Markov chain (MDP)

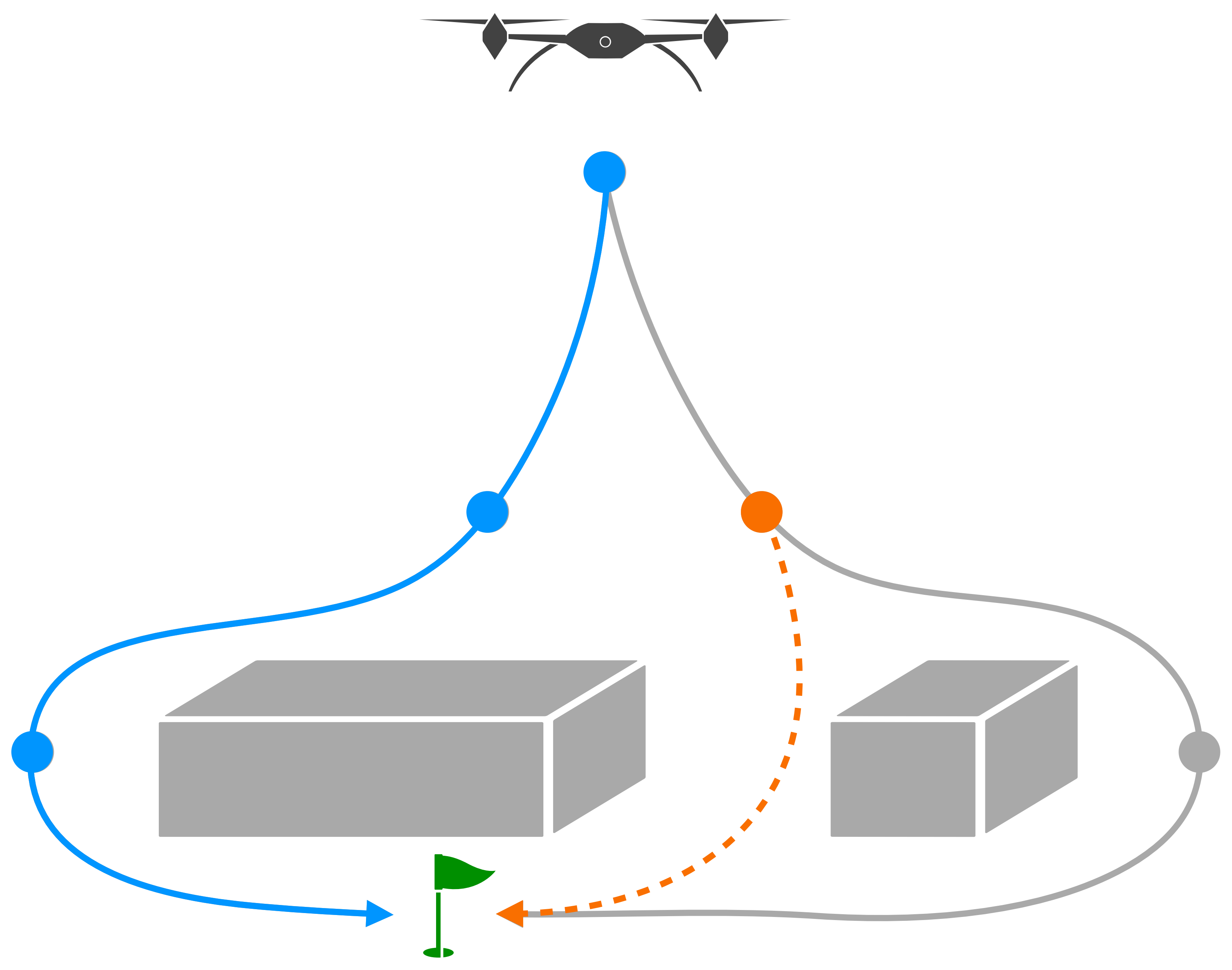
The Dual Role of Control

Adaptive Control

Investigation

VS

Action



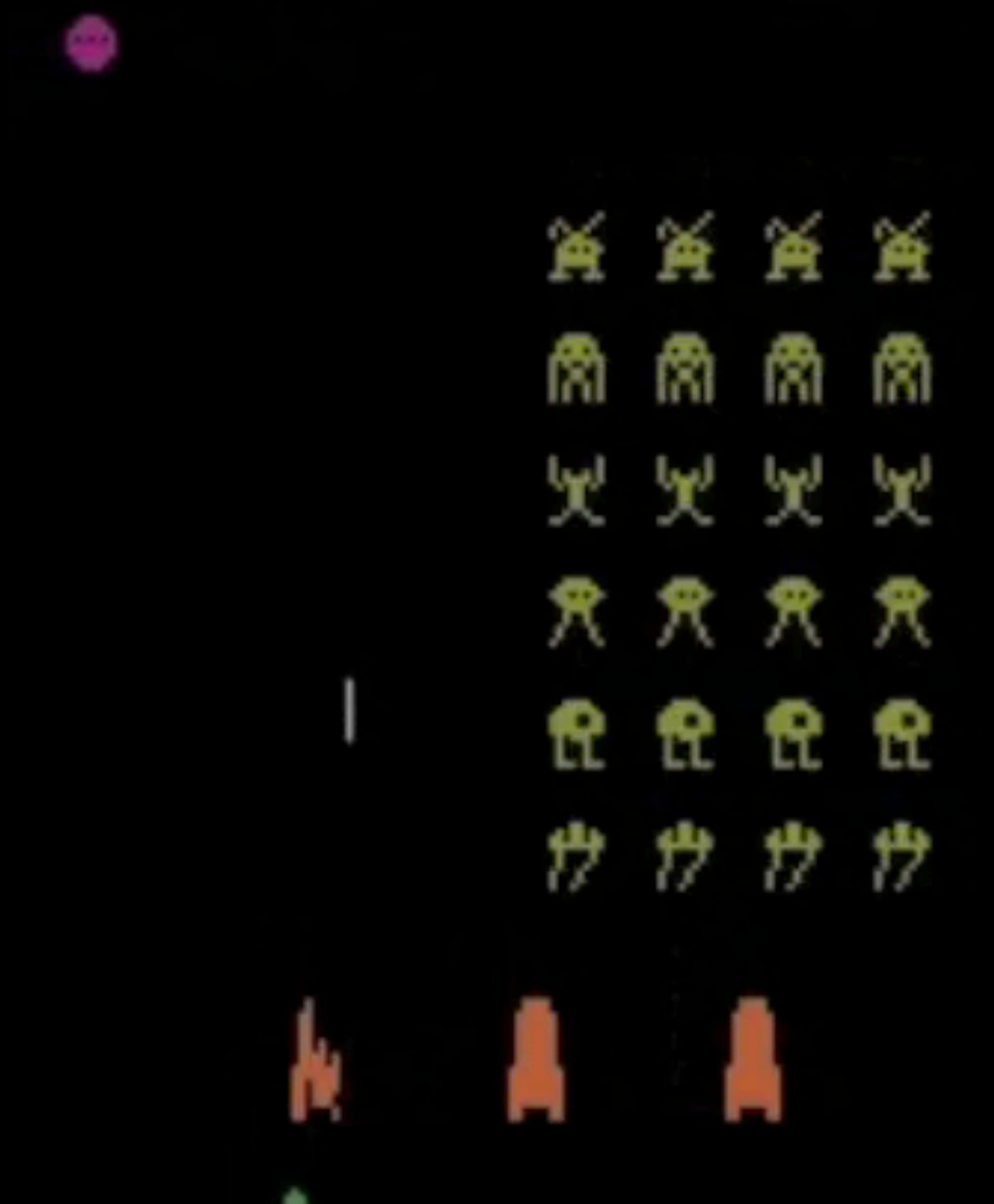
Reinforcement Learning

Exploration

VS

Exploitation

Note: neither of these functions can succeed if we fail to preserve *safety*.



[Mnih et al (DeepMind). Nature 2015]

[Silver et al (DeepMind). Nature 2016]

[Heess et al. (DeepMind). 2017]

[Heess et al. (DeepMind). 2017]

[Heess et al. (DeepMind). 2017]

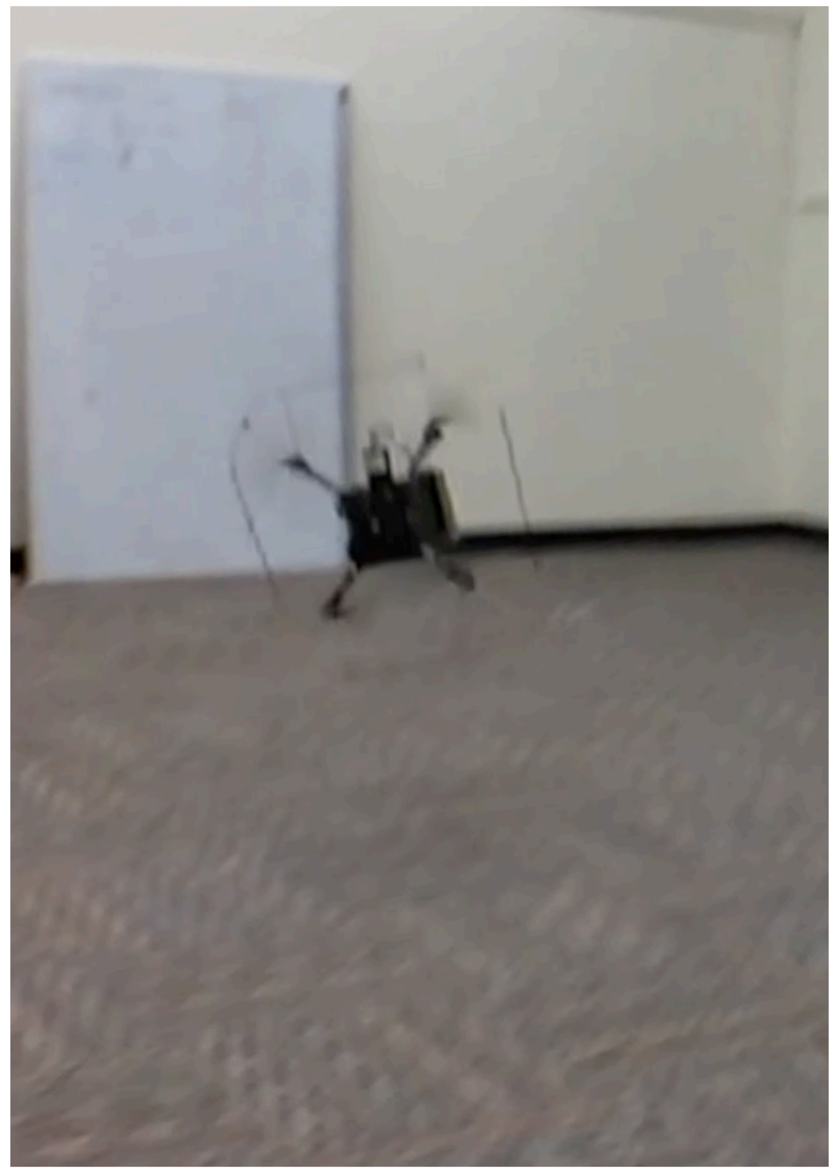
Safety-Critical System

Any system in which there **exist** potential outcomes or failure modes that are deemed **unacceptable**, typically due to injury, loss of life, or severe material damage.

Failure Set: **all** system states that are unacceptable (we *never* want to reach them).

Unsafe Set: states from which it is **not possible to avoid** entering a failure state in the future.

Safe



Unsafe

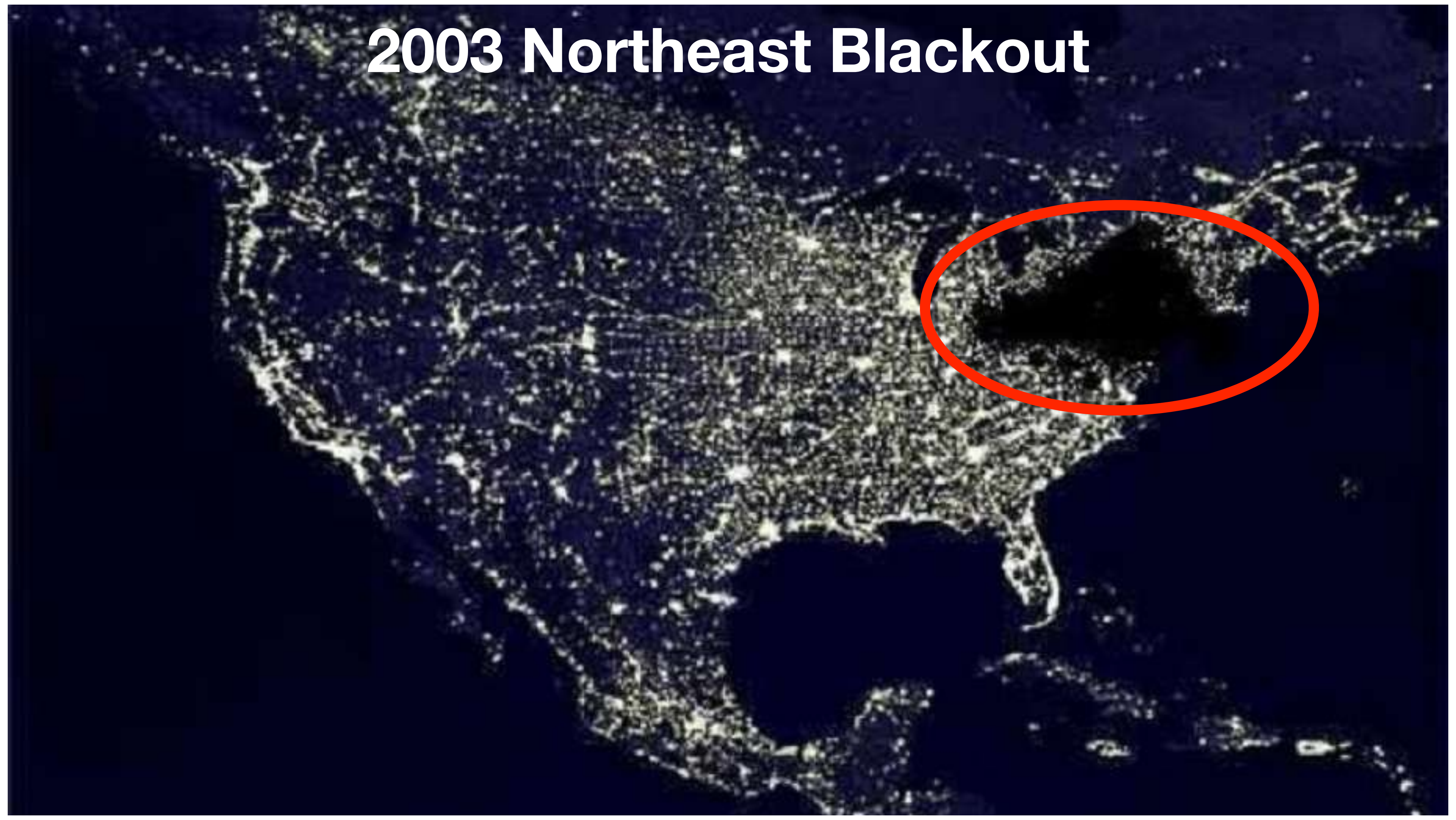
Failure

Is This a *Safety-Critical* System?

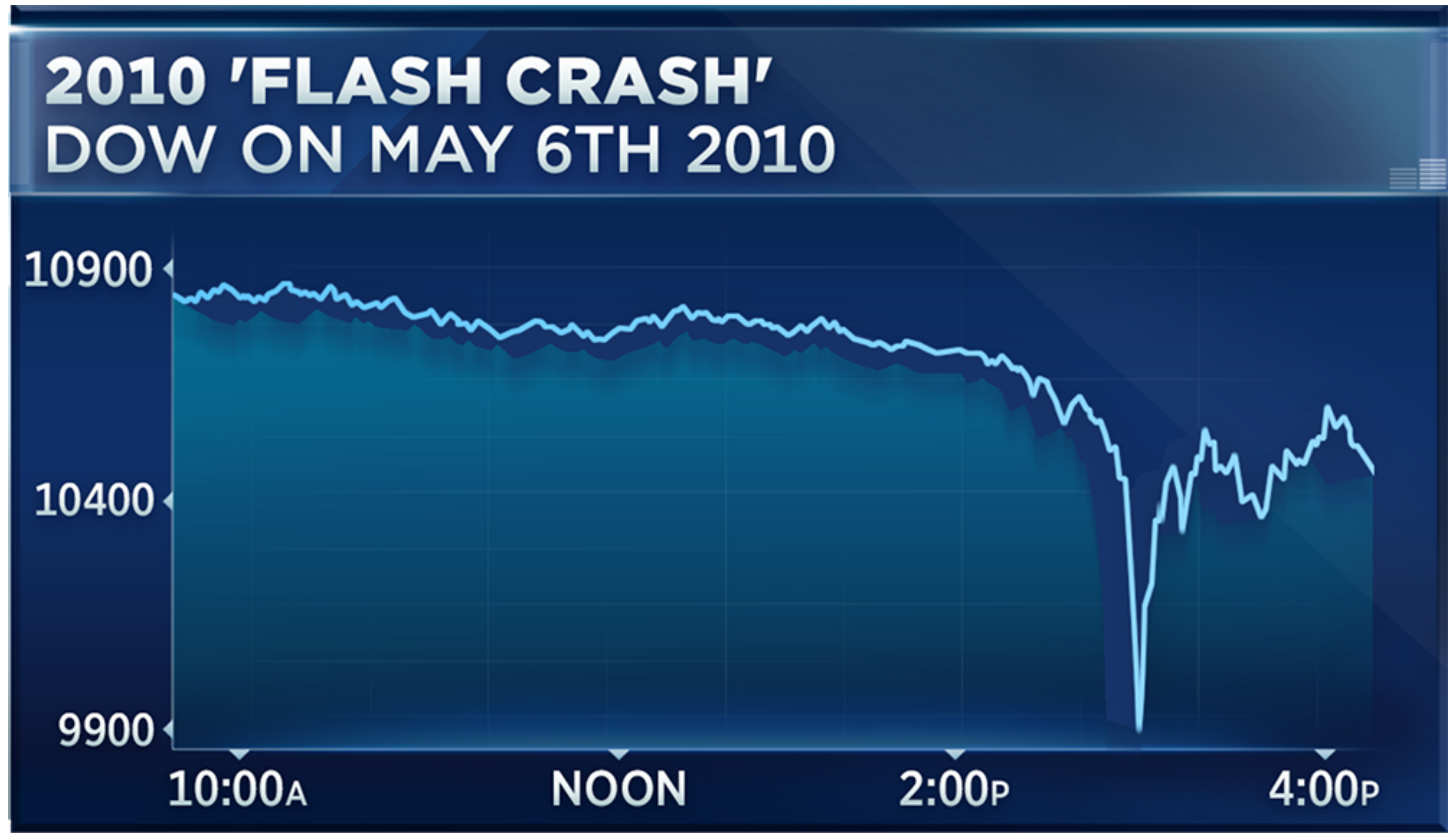
Is This a *Safety-Critical* System?


6:14
78°
8 abc
TULSA

Critical Failures of Cyber-Physical Systems

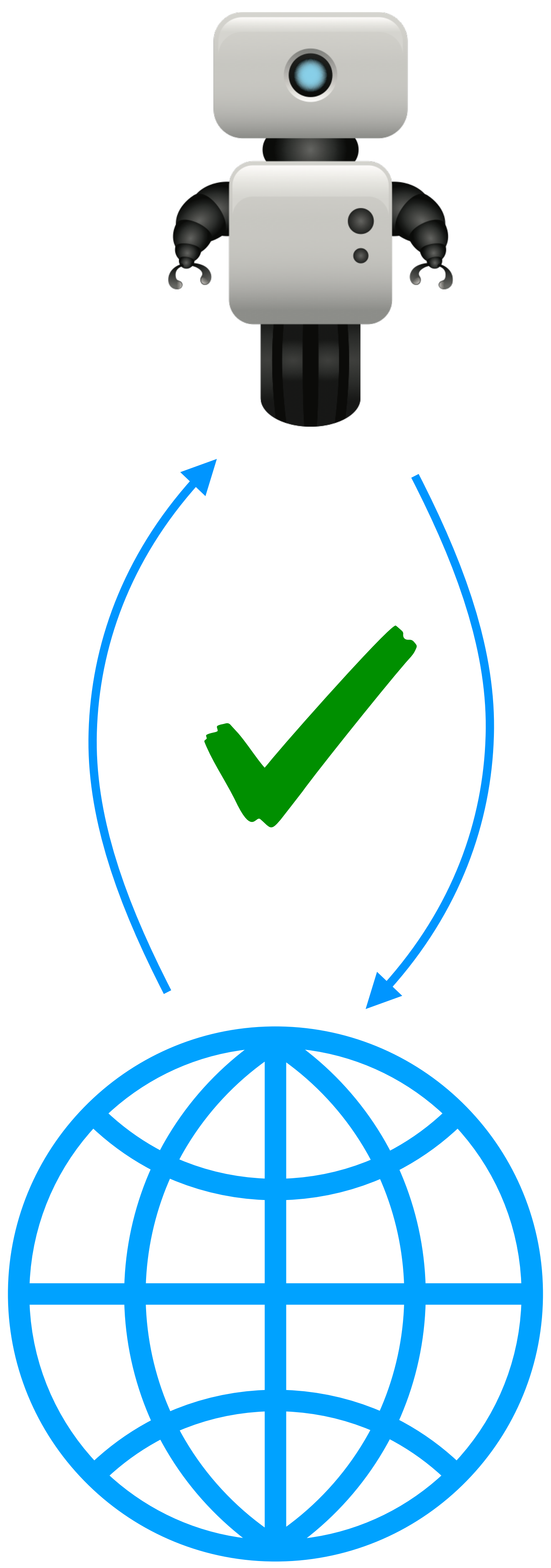


+50M people without power for 2 days



$1T loss in market value

Autonomous System Safety Analysis



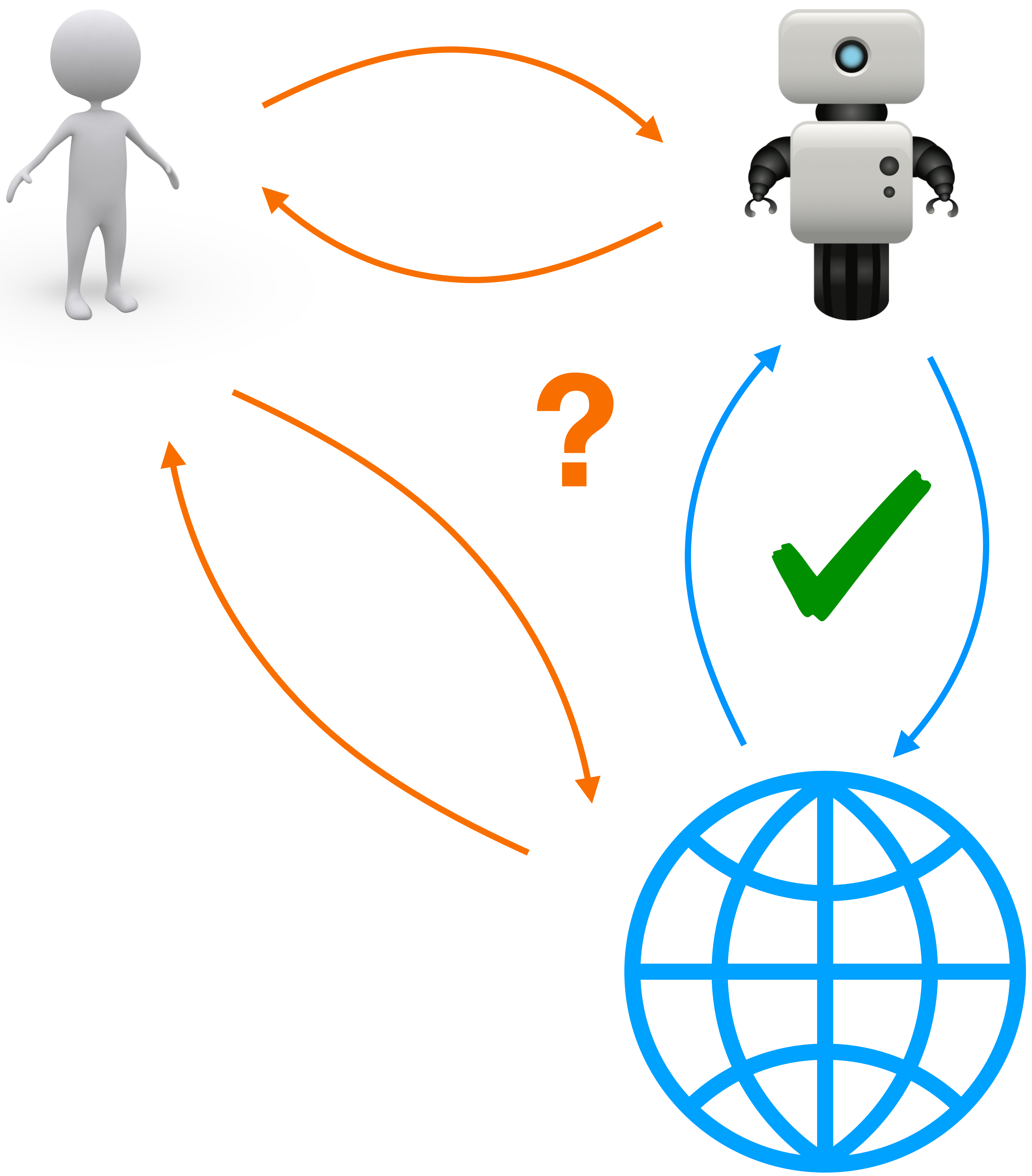
The Long Tail

'Disgruntled' former Waymo self-driving car operator arrested for causing car crash

The 31-year-old swerved his car in front of the autonomous vehicle and then slammed on his brakes

By Andrew J. Hawkins | @andyjayhawk | Feb 13, 2020, 5:27pm EST

Safety is not just up to the Autonomous System

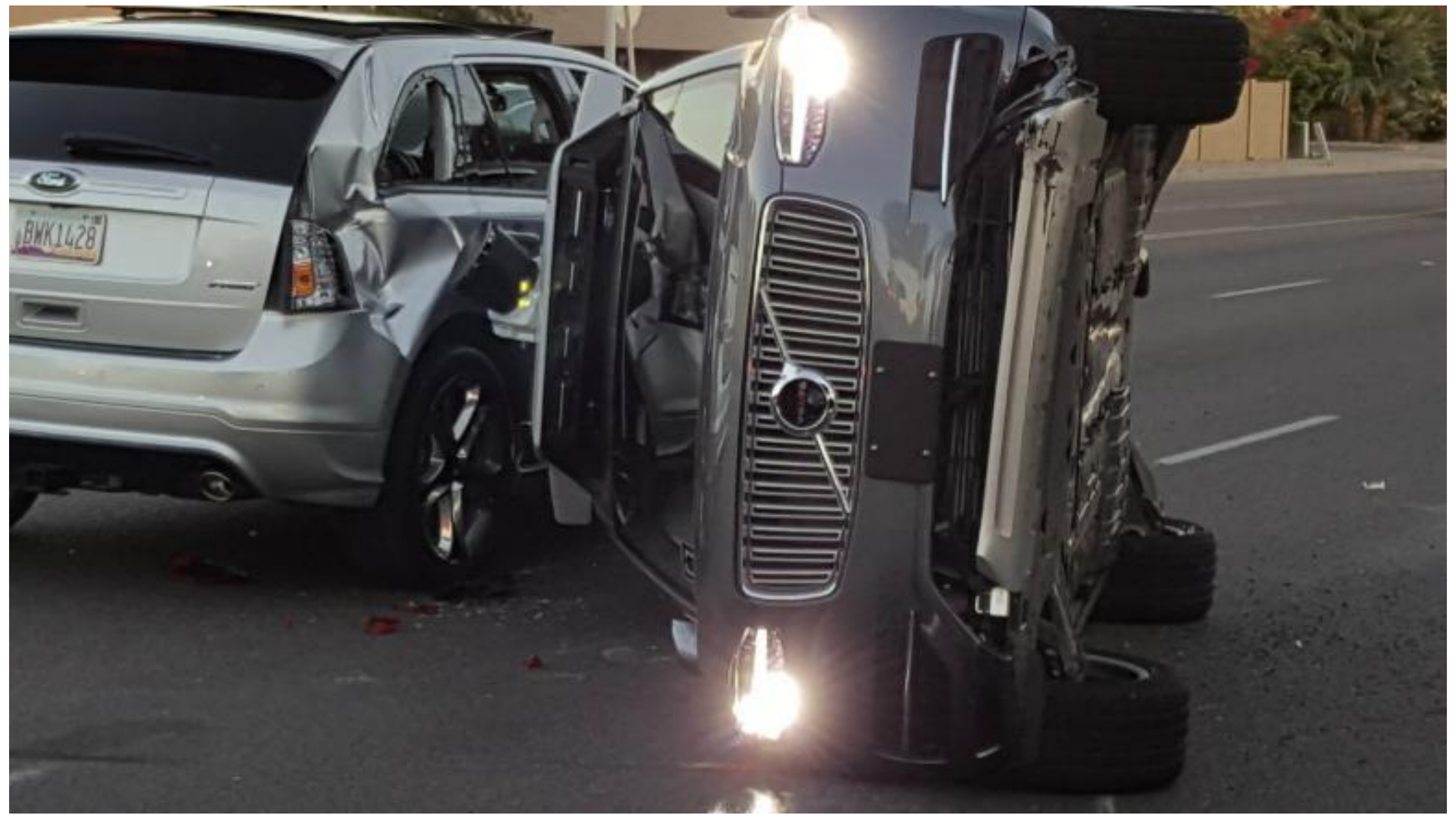


Safety is compromised when humans don't behave the way AI systems assume, or vice versa.

Google, 2016

Turned into bus that did not stop

Uber, 2017



Hit by driver who did not yield

Tesla, 2018

Driver did not take over "fast enough"

YouTube: “Tesla Autopilot tried to kill me.”

Towards Intelligible *if-then* Safety Guarantees

Reliable “hard” guarantees on *<u>under what conditions</u>* the system can/cannot fail. Trust

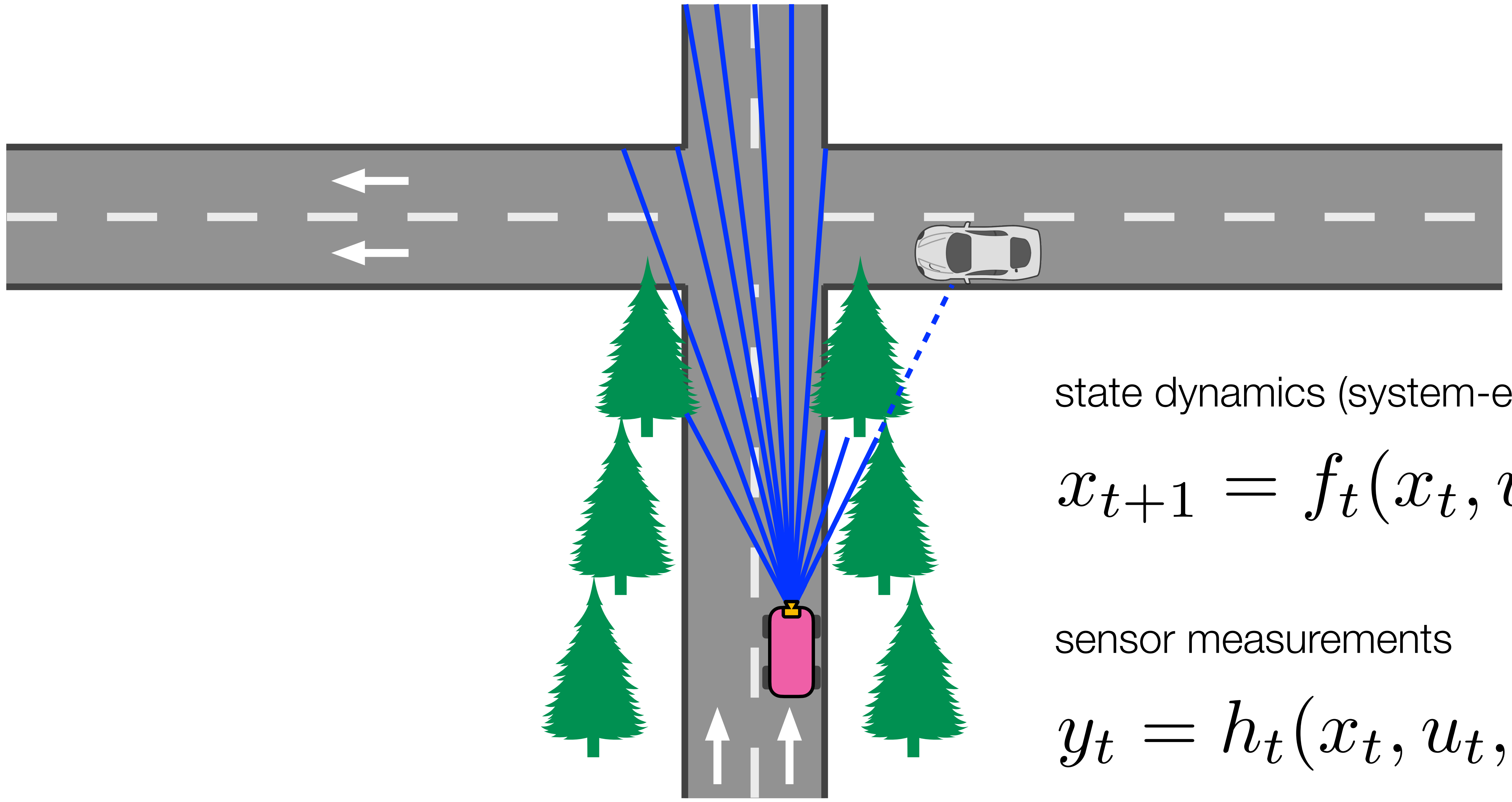
Transparent to both system designers and the public. Social contract

Checkable and enforceable at runtime by autonomy stack. Active safety

Traceable retrospectively/counterfactually in the event of a failure/near miss. Accountability

Operational Design Domain (ODD) Safety Theory

ODD: set of conditions under which the system must operate correctly and safely.



state dynamics (system-environment)

$$x_{t+1} = f_t(x_t, u_t, d_t)$$

sensor measurements

$$y_t = h_t(x_t, u_t, d_t)$$

uncertainty (environment) realization (incl. behavior of other agents)

$$\mathbf{d} := (d_0, d_1, \dots)$$

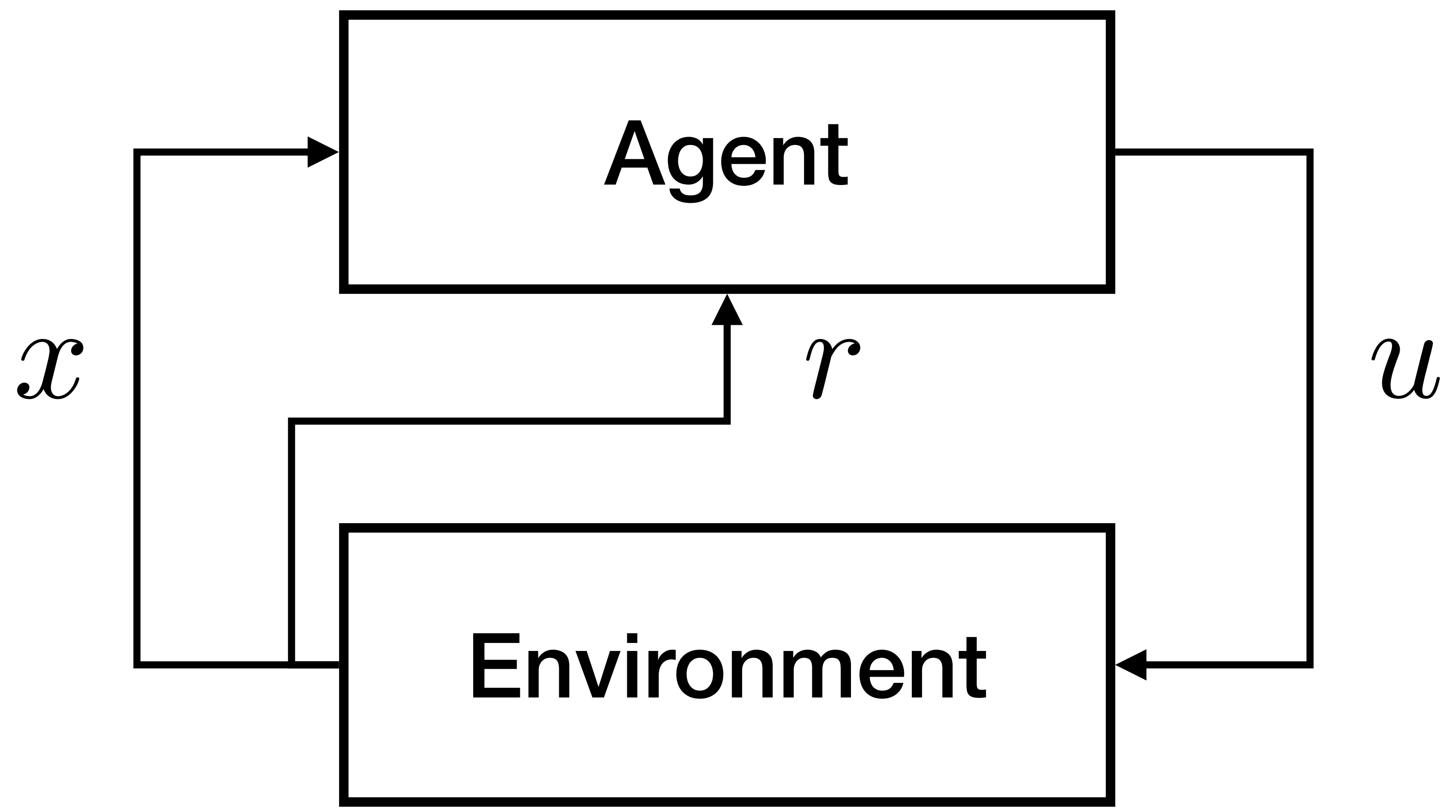
$$\boxed{\mathbb{O} := (\mathcal{X}_0, \mathbb{D}, \mathcal{F})}$$

Operational Design Domain

deployment conditions | environment realization | failure conditions

ODD Theorems: let the environment realization satisfy $\mathbf{d} \in \mathbb{D}$, then the system's operation from any deployment condition $x_0 \in \mathcal{X}_0$ satisfies $x_t \notin \mathcal{F}$ for all time $t \geq 0$.

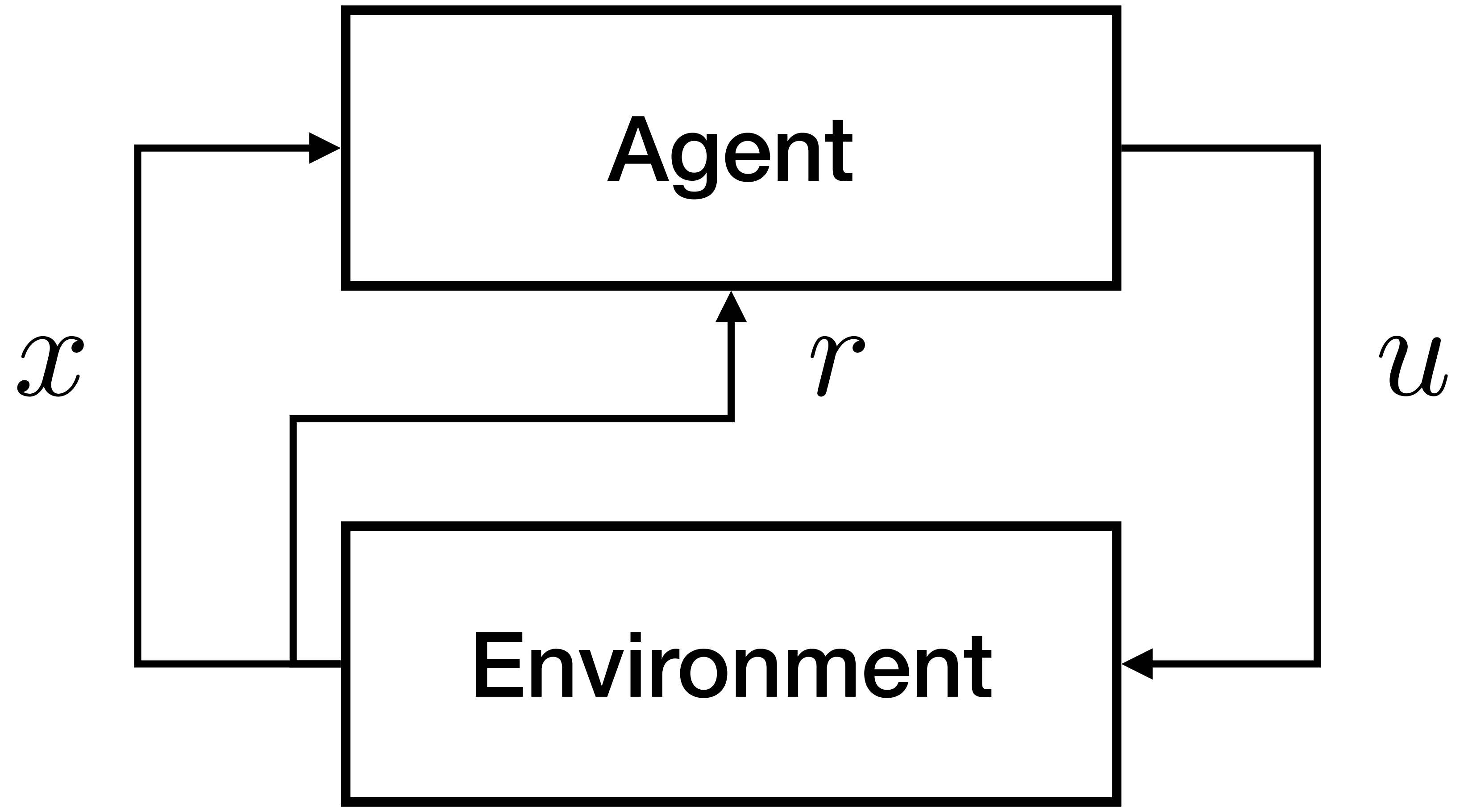
Reinforcement Learning and Constraints



$$\max_{\pi} \mathbb{E} \sum_{t=0}^{\infty} \gamma^t r\big(\mathbf{x}(t)\big)$$ Additive/Average Performance

$$\text{s.t.} \quad l\big(\mathbf{x}(t)\big) \geq 0 \quad \forall t \geq 0$$ Property Satisfaction

Reinforcement Learning and Constraints



$$\max_{\pi} \mathbb{E} \sum_{t=0}^{\infty} \gamma^t r\big(\mathbf{x}(t)\big)$$

Additive/Average Performance

$$\text{s.t.} \quad \inf_{t \geq 0} \; l\big(\mathbf{x}(t)\big) \geq 0$$

Property Satisfaction

The Safe Learning Problem

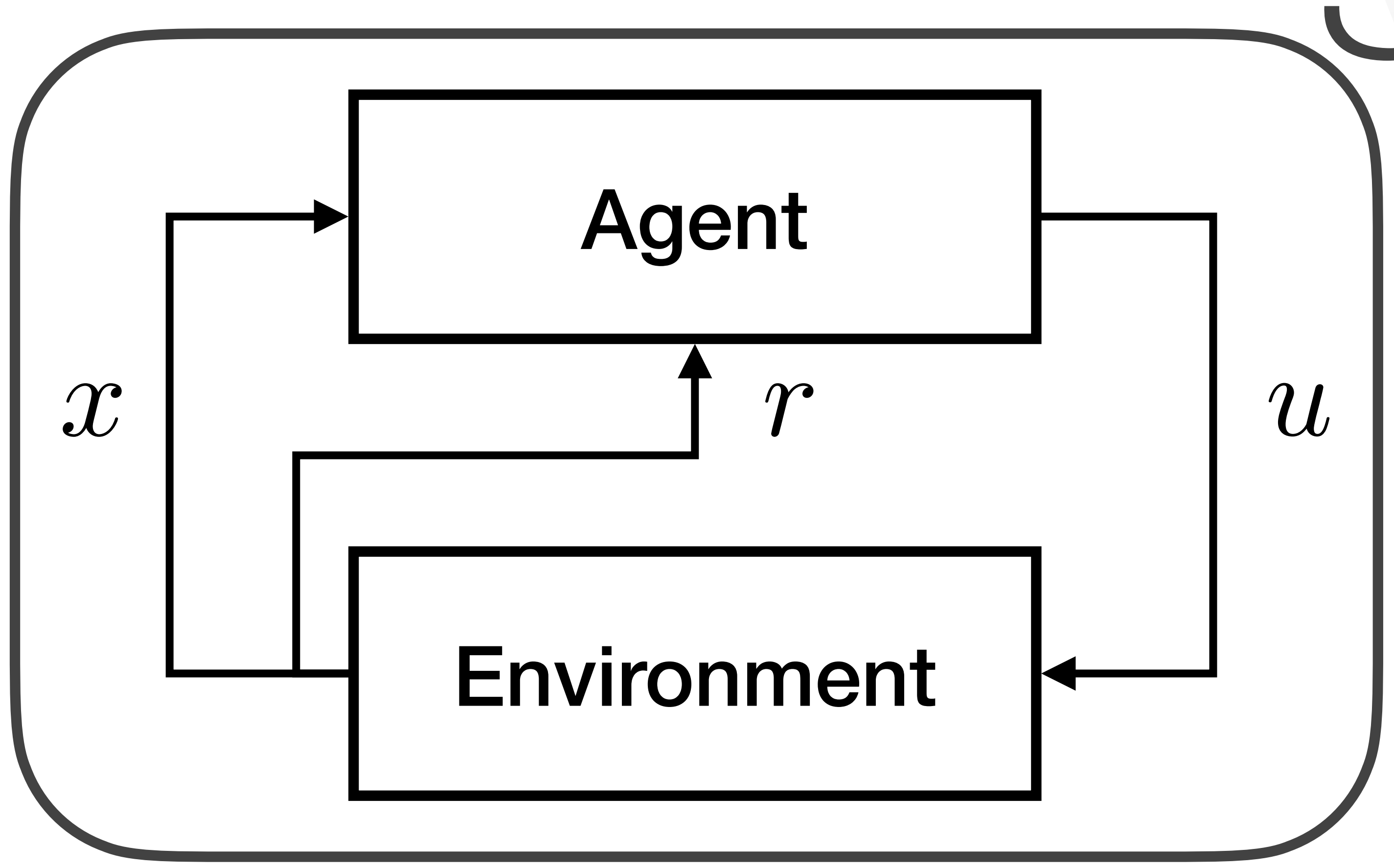
CAN ROBOTIC SYSTEMS **LEARN** BY EXPERIENCE

WHILE ALWAYS SATISFYING **SAFETY** CONSTRAINTS?

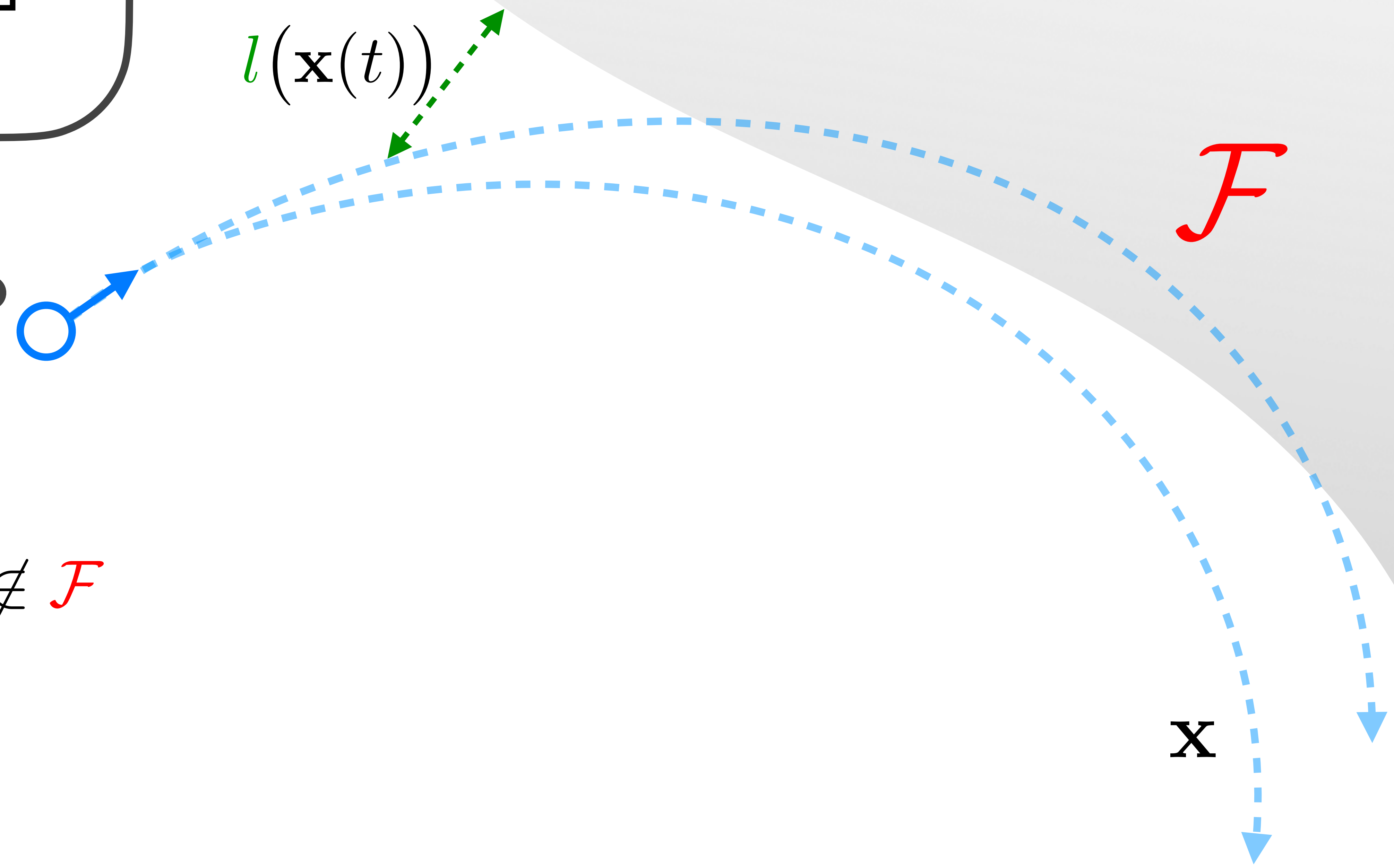
Can the learned control policy maintain safety once it is deployed on the robot?

Can the learning process maintain safety while the robot is learning its control policy?

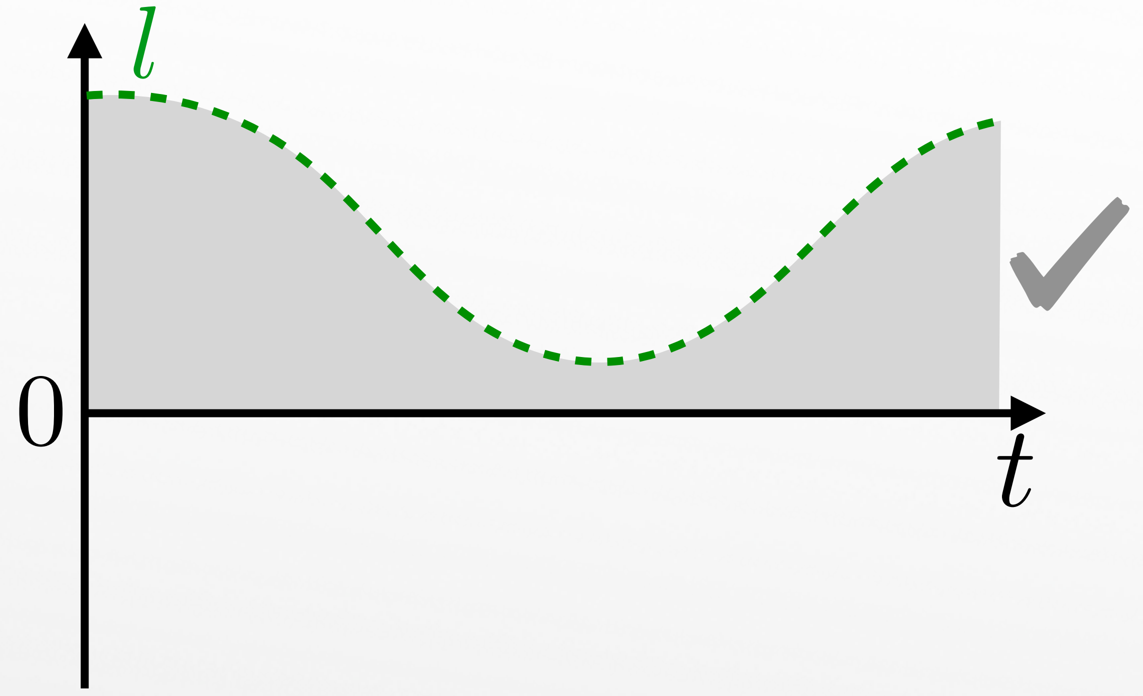
Safety Bellman Equation

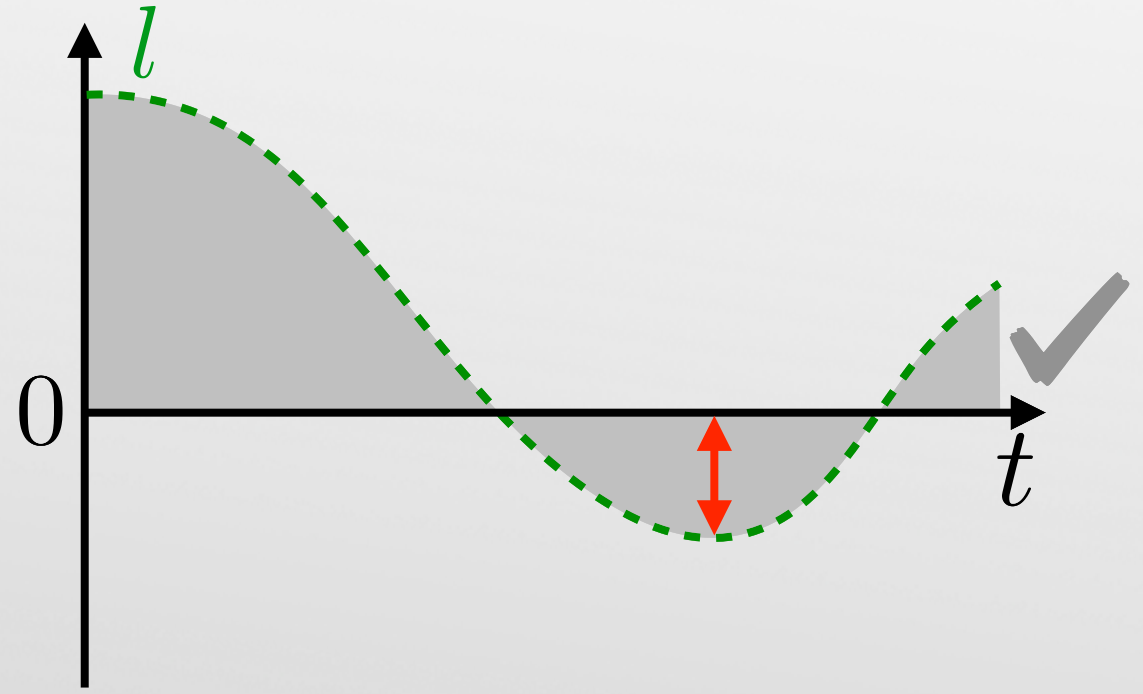


$x_{t+1} = f(x_t, u_t)$

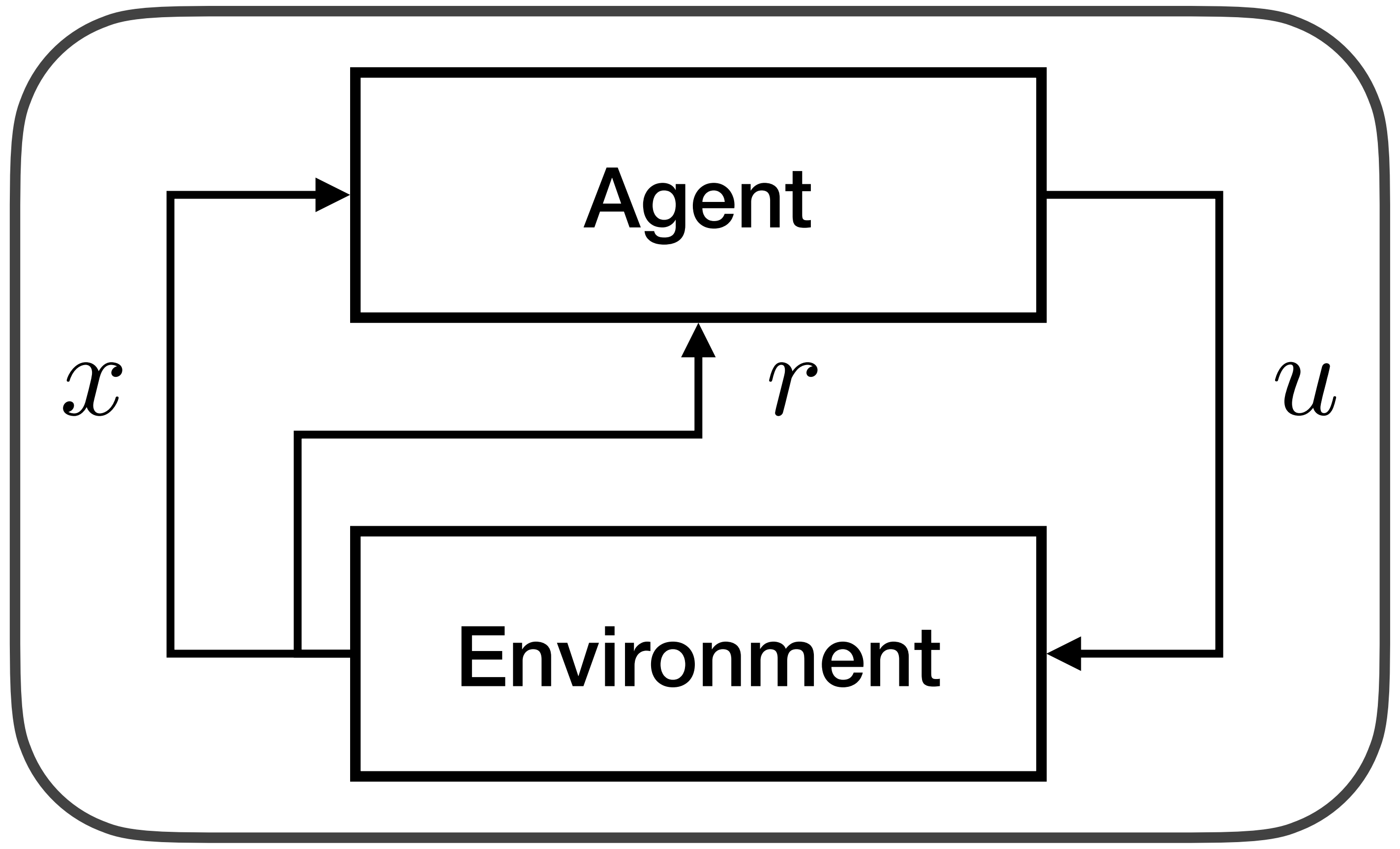


$\exists\, \mathbf{u} : \forall t, \mathbf{x}(t) \notin \mathcal{F}$

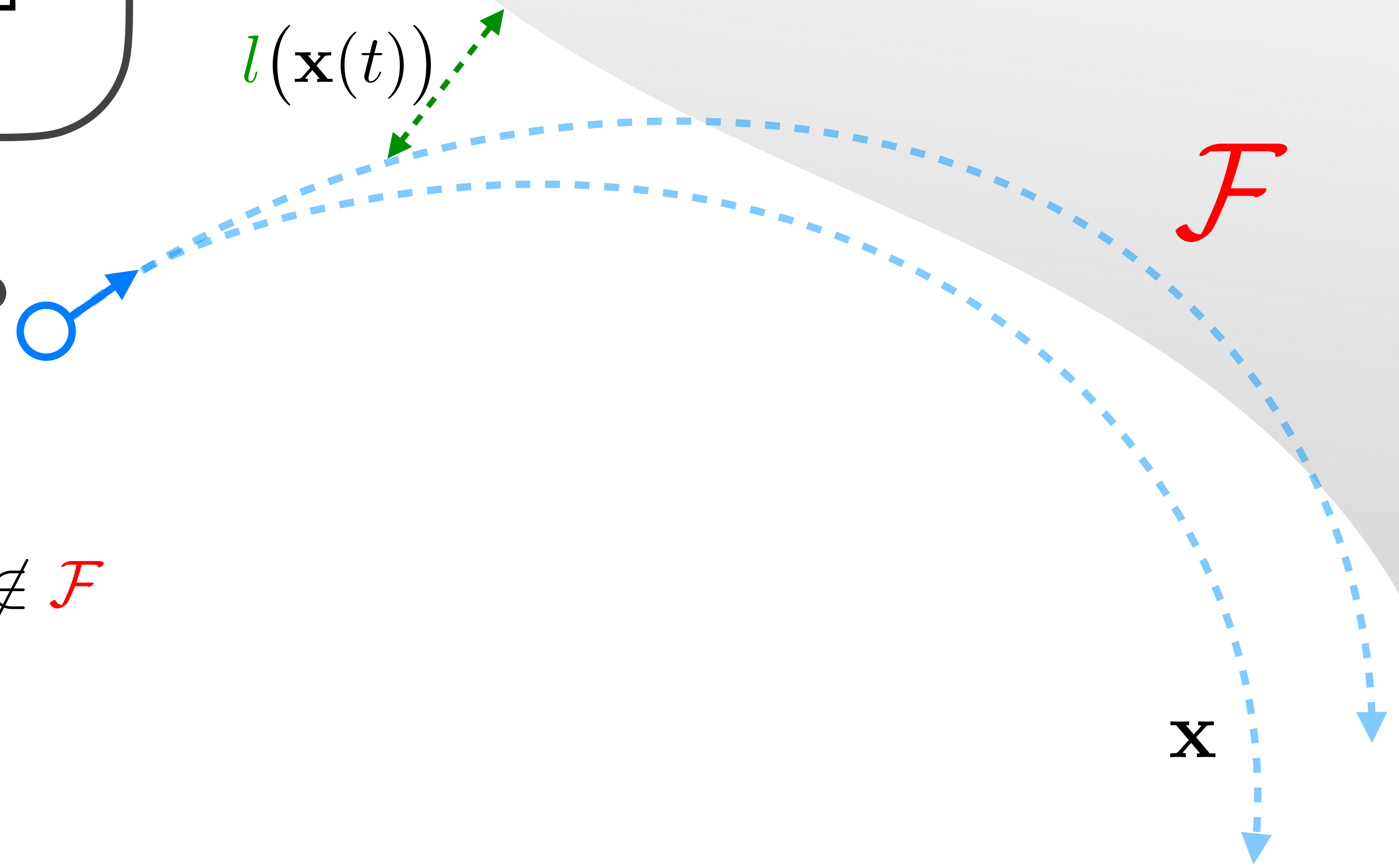




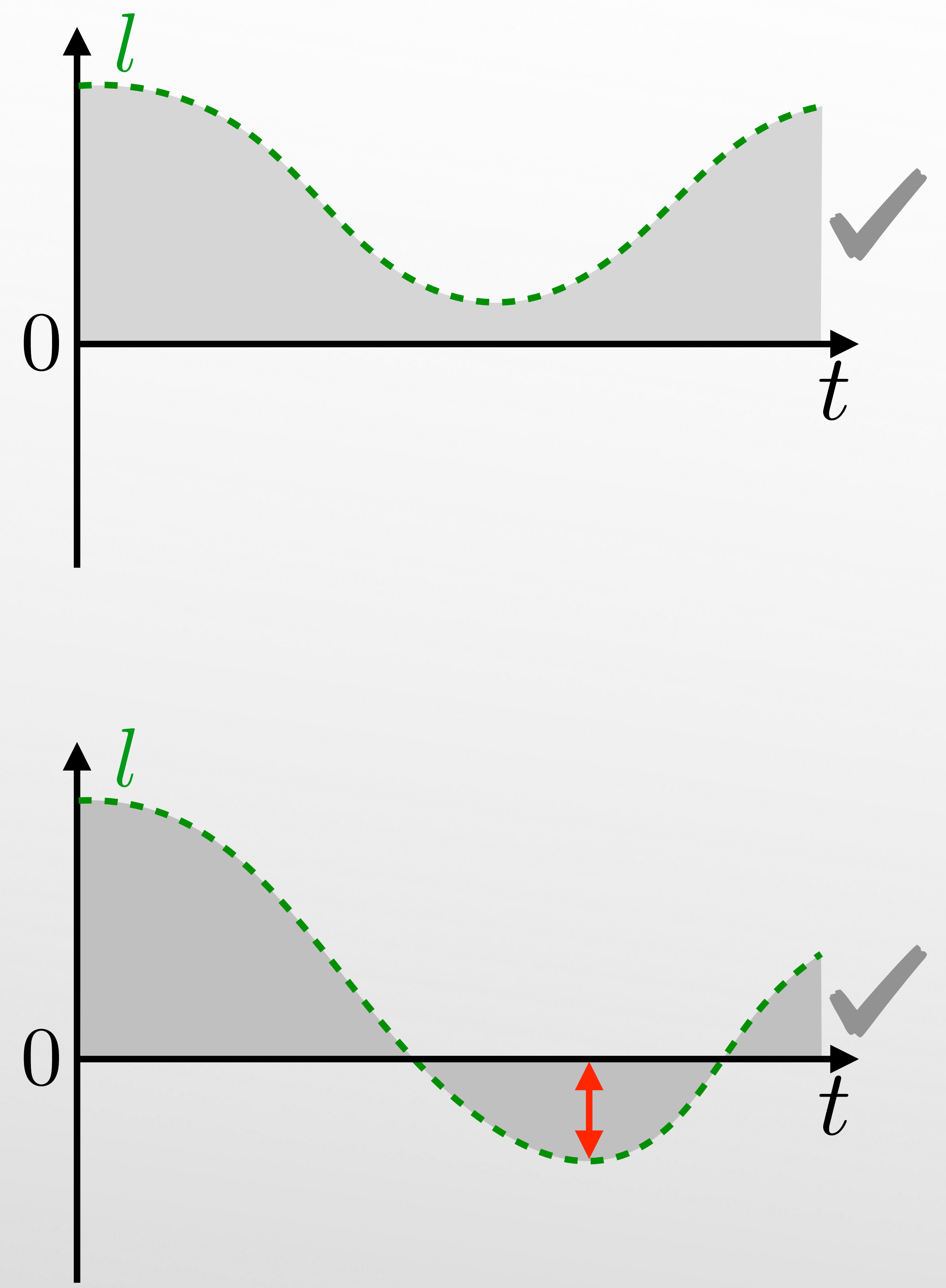
A Safety Bellman Backup



$\dot{x} = f(x, u)$



$\exists\, \mathbf{u} : \forall t, \mathbf{x}(t) \notin \mathcal{F}$



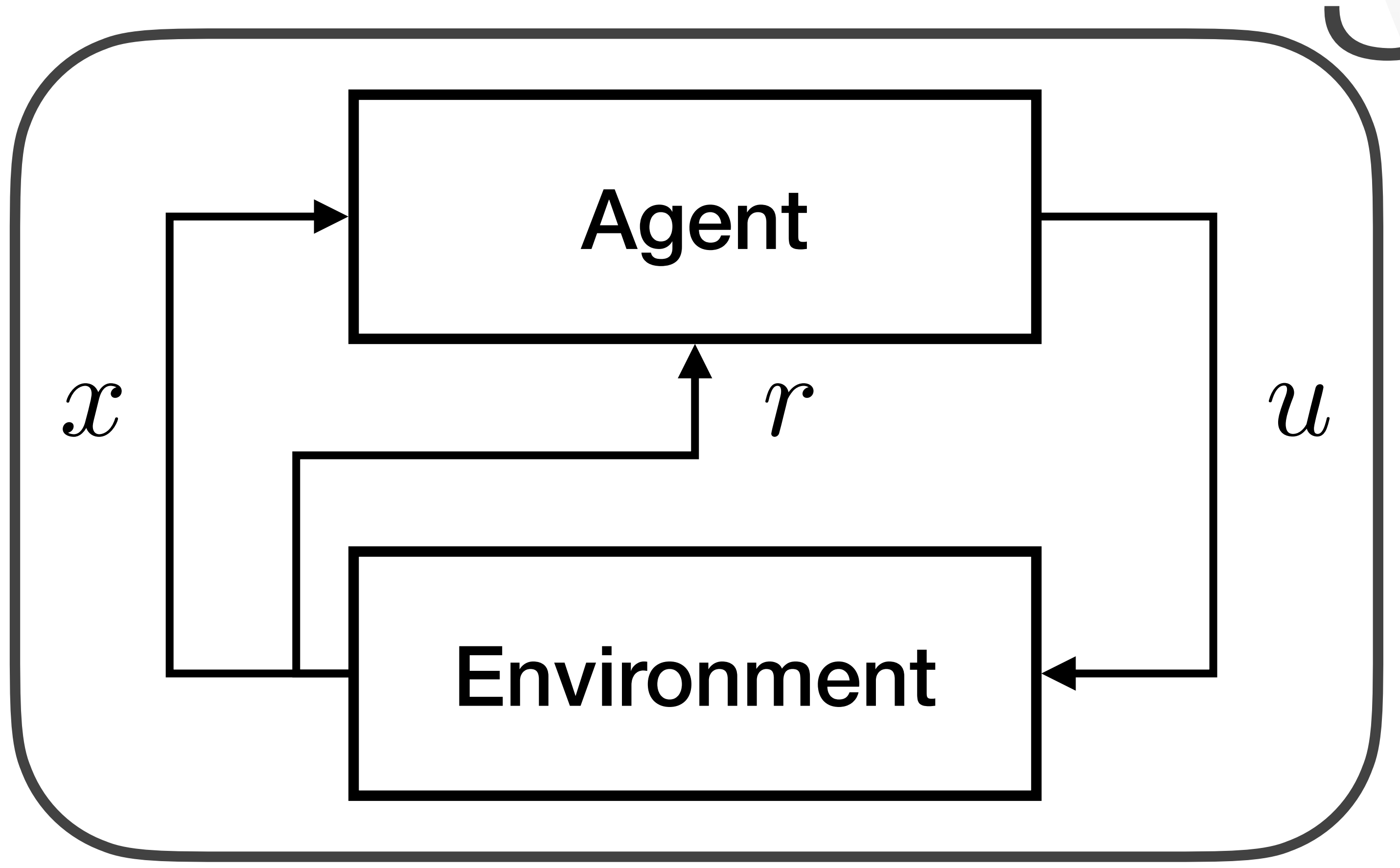
$$V(x) = \max_u \min \left\{ l(x), V\big(f(x,u)\big) \right\}$$

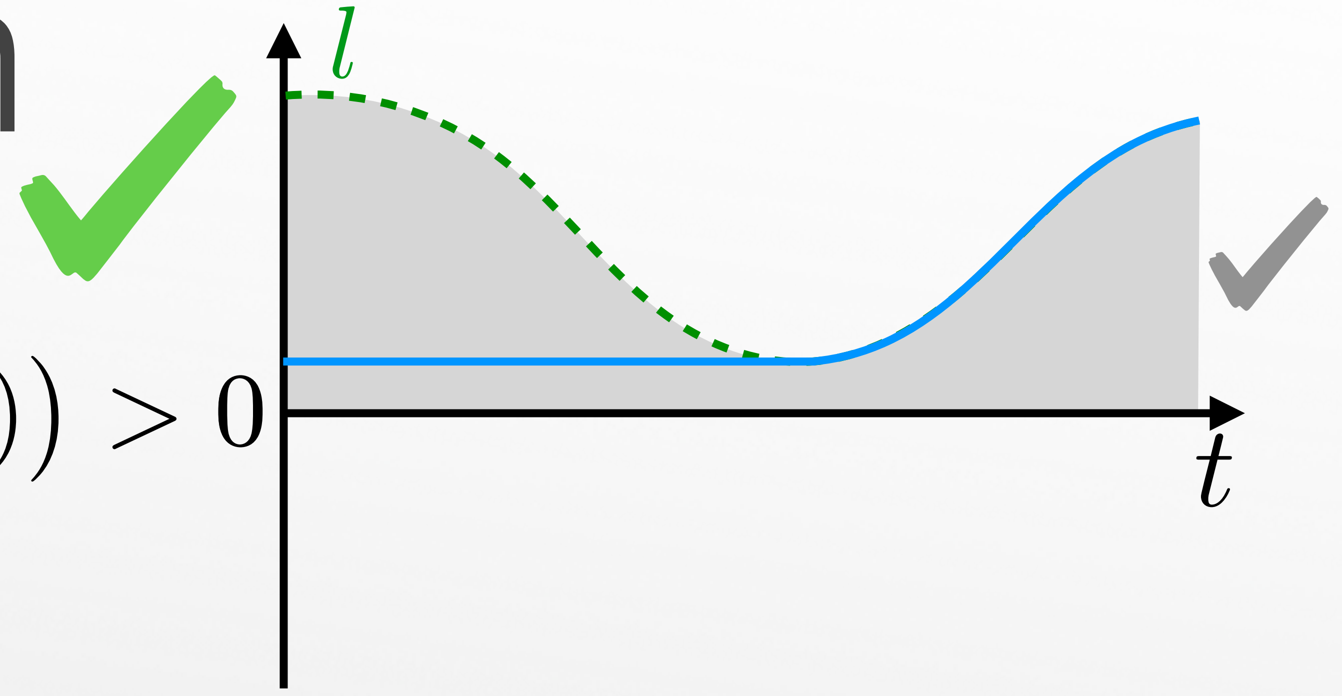
Keep the lowest value between *now* and *later*.

$$V(x) = \max_u \min \left\{ l(x), V\left(f(x,u)\right) \right\}$$

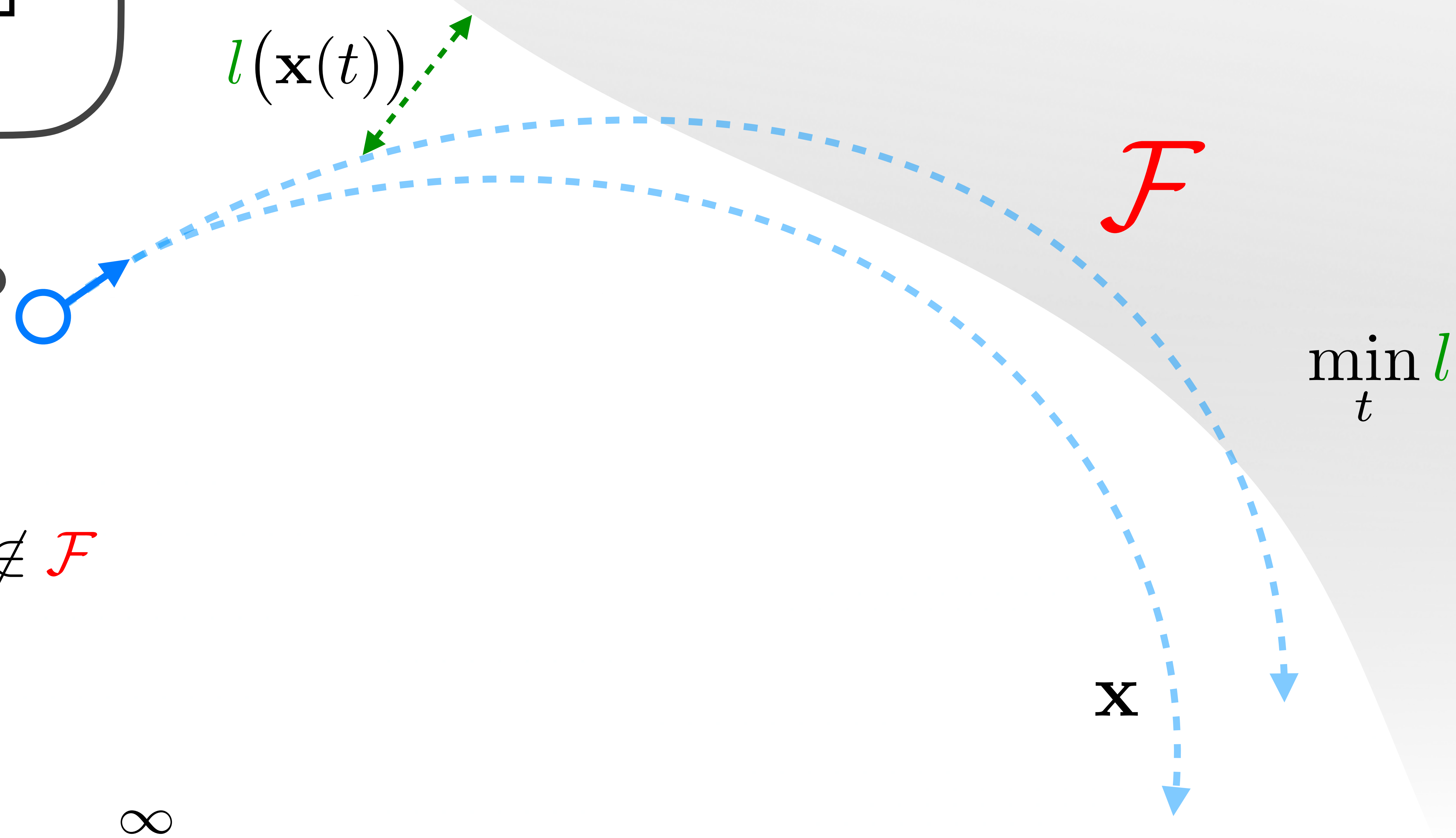
Keep the lowest value between *now* and *later*.

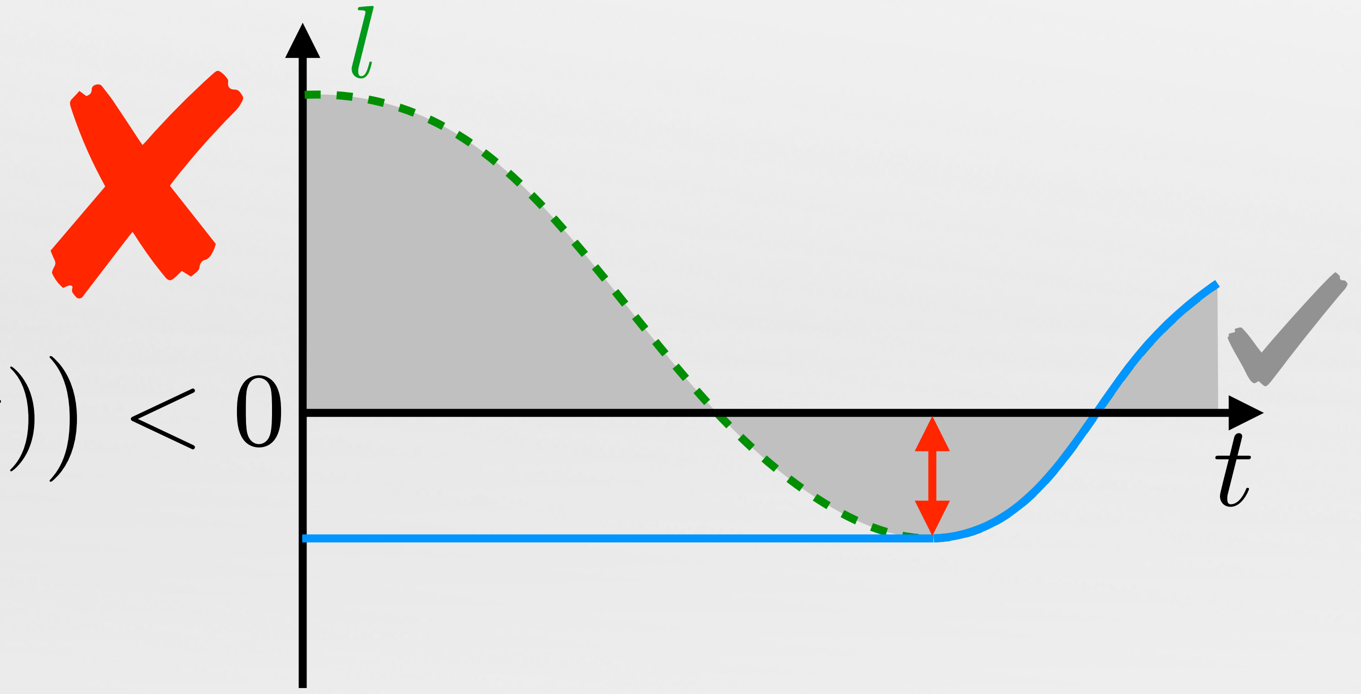
Safety Bellman Equation





$$\min_t l(\mathbf{x}(t)) > 0$$





$$\min_t l(\mathbf{x}(t)) < 0$$

$$x_{t+1} = f(x_t, u_t)$$

$$\exists\, \mathbf{u} : \forall t, \mathbf{x}(t) \notin \mathcal{F}$$

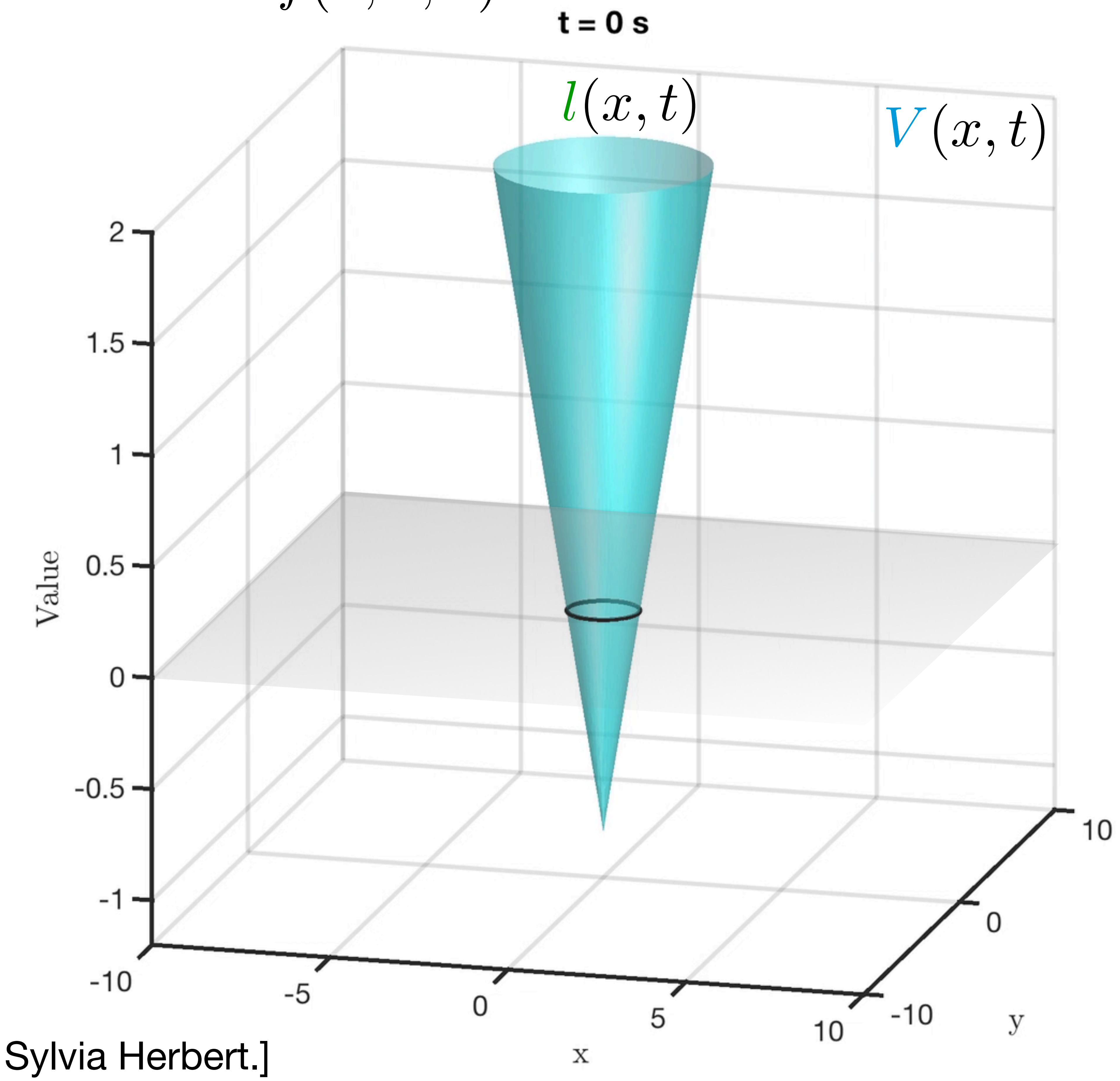
Lagrange (sum) $$J(\mathbf{x}) = \sum_{t=0}^{\infty} \gamma^t r(\mathbf{x}(t)) \qquad V(x) = \max_u \Big[r(x,u) + \gamma V\big(f(x,u)\big) \Big]$$

Reachability (min) $$J(\mathbf{x}) = \inf_{t \geq 0} l(\mathbf{x}(t)) \qquad V(x) = \max_u \Big[\min \big\{ l(x), V\big(f(x,u)\big) \big\} \Big]$$

Hamilton-Jacobi Safety Analysis

Safe set: states from which the controller can keep the system from entering any failure state in the future $\iff V(x) \geq 0$

$\dot{x} = f(x, u, d)$



[Animation by Sylvia Herbert.]

Continuous-Time Dynamic Programming

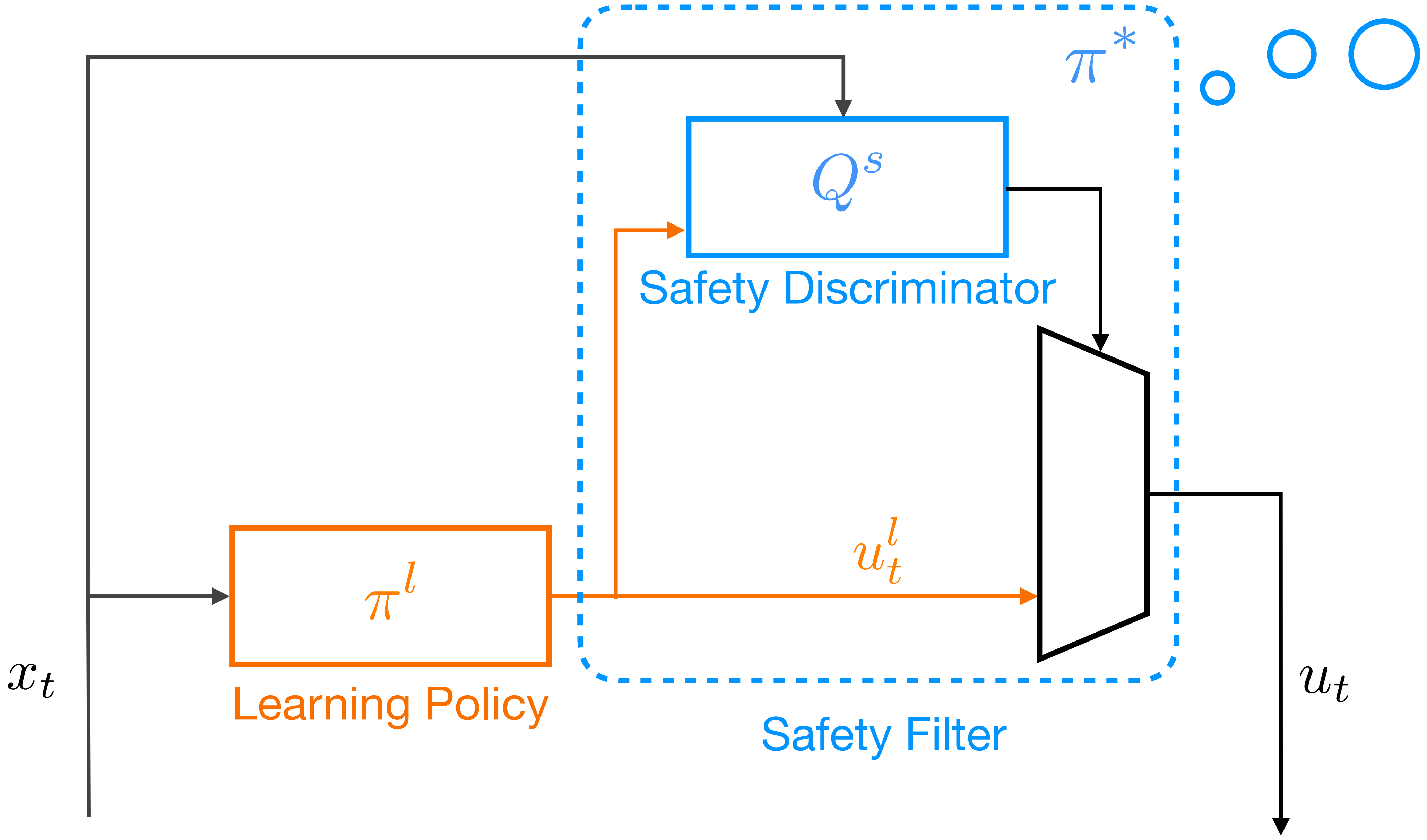
$$0 = \min \left\{ l(x,t) - V(x,t), \frac{\partial V(x,t)}{\partial t} + \max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} \nabla_x V(x,t)^\top f(x,u,d) \right\}$$

Discrete-Time Dynamic Programming

$$V(x,t) = \max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} \min \left\{ l(x,t), V\big(x + f(x,u,d)\Delta t, t + \Delta t\big) \right\}$$

[**Fisac**, Chen, Tomlin, and Sastry. HSCC 2015]

Learning with Safety Filters



HJI safety analysis | Control barrier function | Model-predictive shielding | Safety critic

Safety Discriminator

$Q^s(x,u) > 0 \implies$ known π^s maintains safety after taking action u from x.

Safety Filter

$\pi^*(x,u)$ modifies the proposed action to avoid future safety violations.

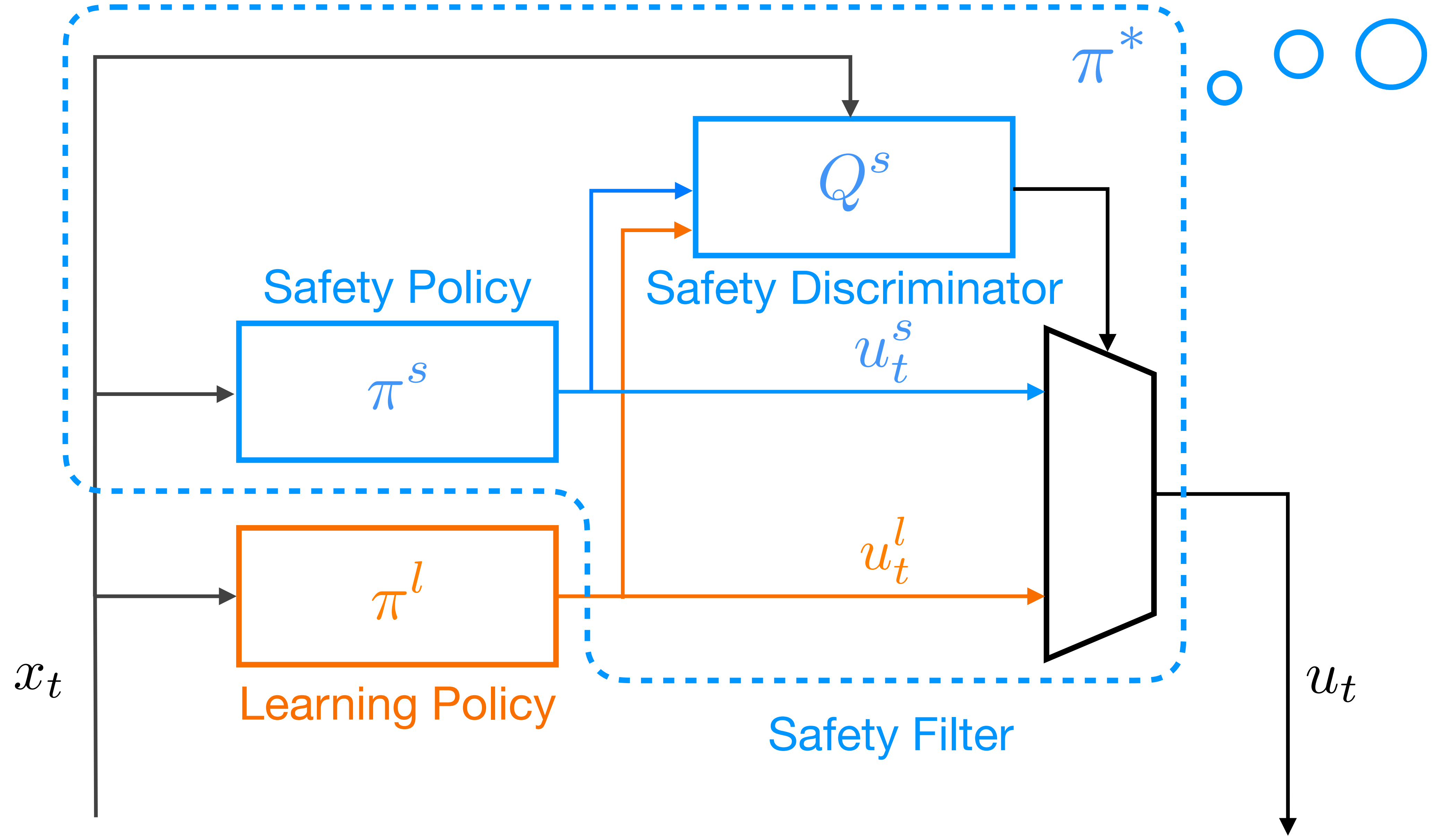
Theorem: any safety filter π^* that enforces

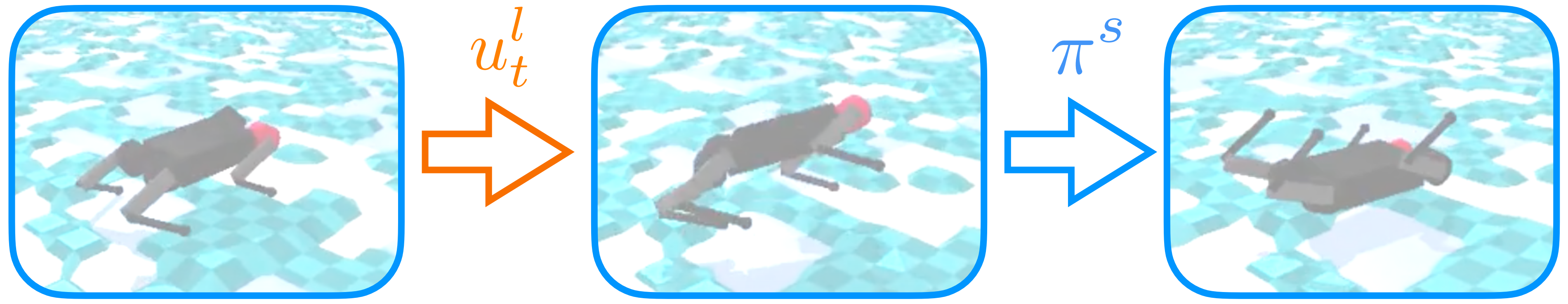
$$Q^s\big(x, \pi^*(x, u^l)\big) > 0 \qquad (1)$$

maintains safety for all time from any initial state x_0 that is safe under π^s, i.e. $Q^s\big(x_0, \pi^s(x_0)\big) > 0$.

Moreover, (1) is recursively enforceable, since π^* can always choose $\pi^s(x)$.

Model-Predictive Shielding





Safety Discriminator: Policy Rollout

$$Q^s(x,u) := \mathbb{1}\{\exists \tau \in [t_1, T], \mathbf{x}^{\pi^s}_{x_1,t_1}(\tau) \in \Omega^\pi \wedge \forall s \in [t_1, \tau], \mathbf{x}^{\pi^s}_{x_1,t_1}(s) \notin \mathcal{F}\}$$

$x_1 := \mathbf{x}^u_{x,t_0}(t_1)$
$t_1 := t_0 + \Delta t$

Safety Filter: Policy Switch

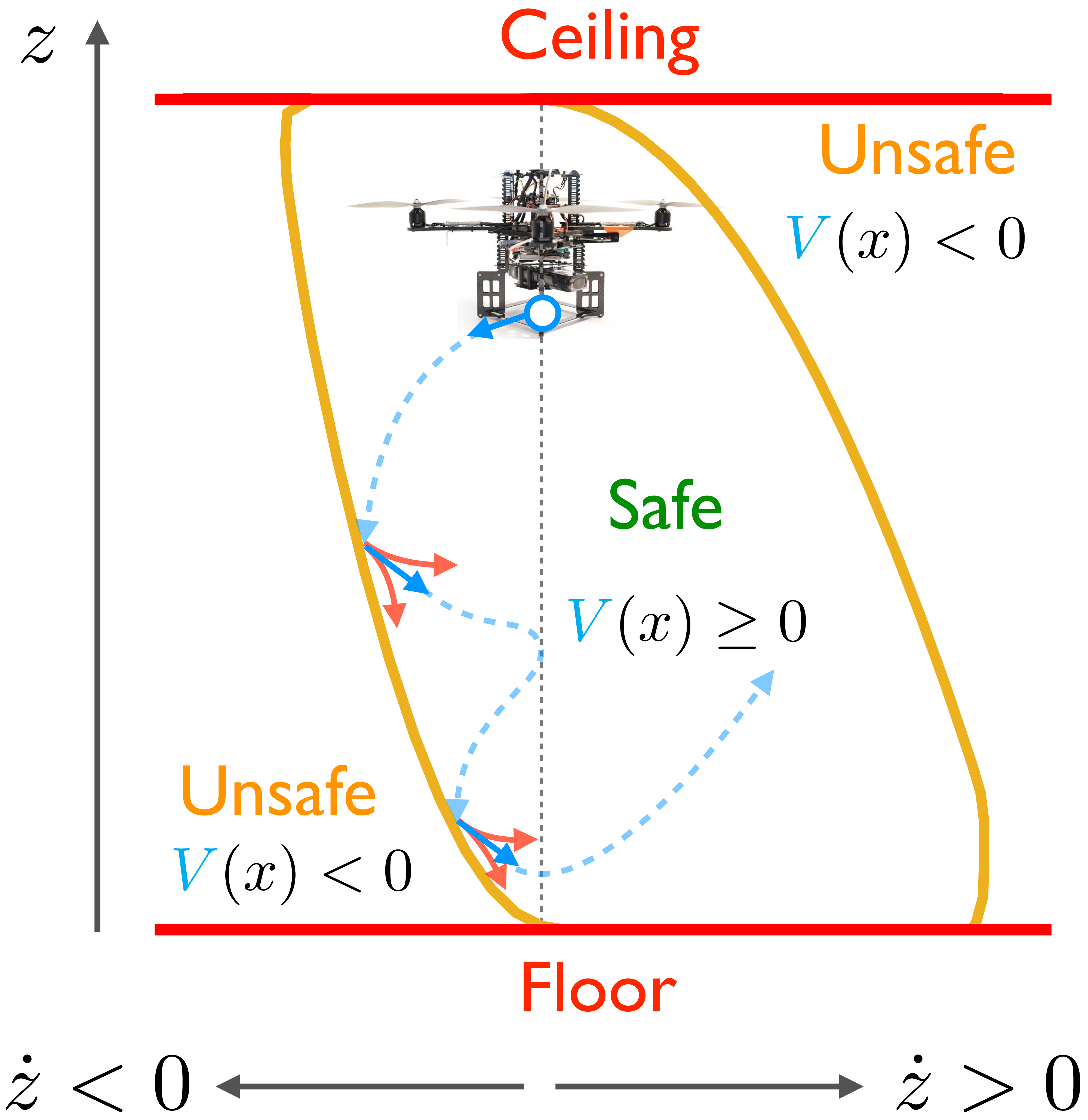
$\pi^*(x,u)$ replaces u^l by u^s whenever $Q^s(x,u^l) = 0.$

Theorem: the safety filter π^* given by

$$\pi^*(x,u^l) := \begin{cases} u^l, & Q^s(x,u^l) = 1 \\ \pi^s(x), & Q^s(x,u^l) = 0 \end{cases}$$

maintains safety for all time from any initial state x_0 that is safe under π^s, i.e. $Q^s(x_0, \pi^s(x_0)) > 0$.

Hamilton-Jacobi Safety Analysis



$$\ddot{z} = k_T u - g + d$$

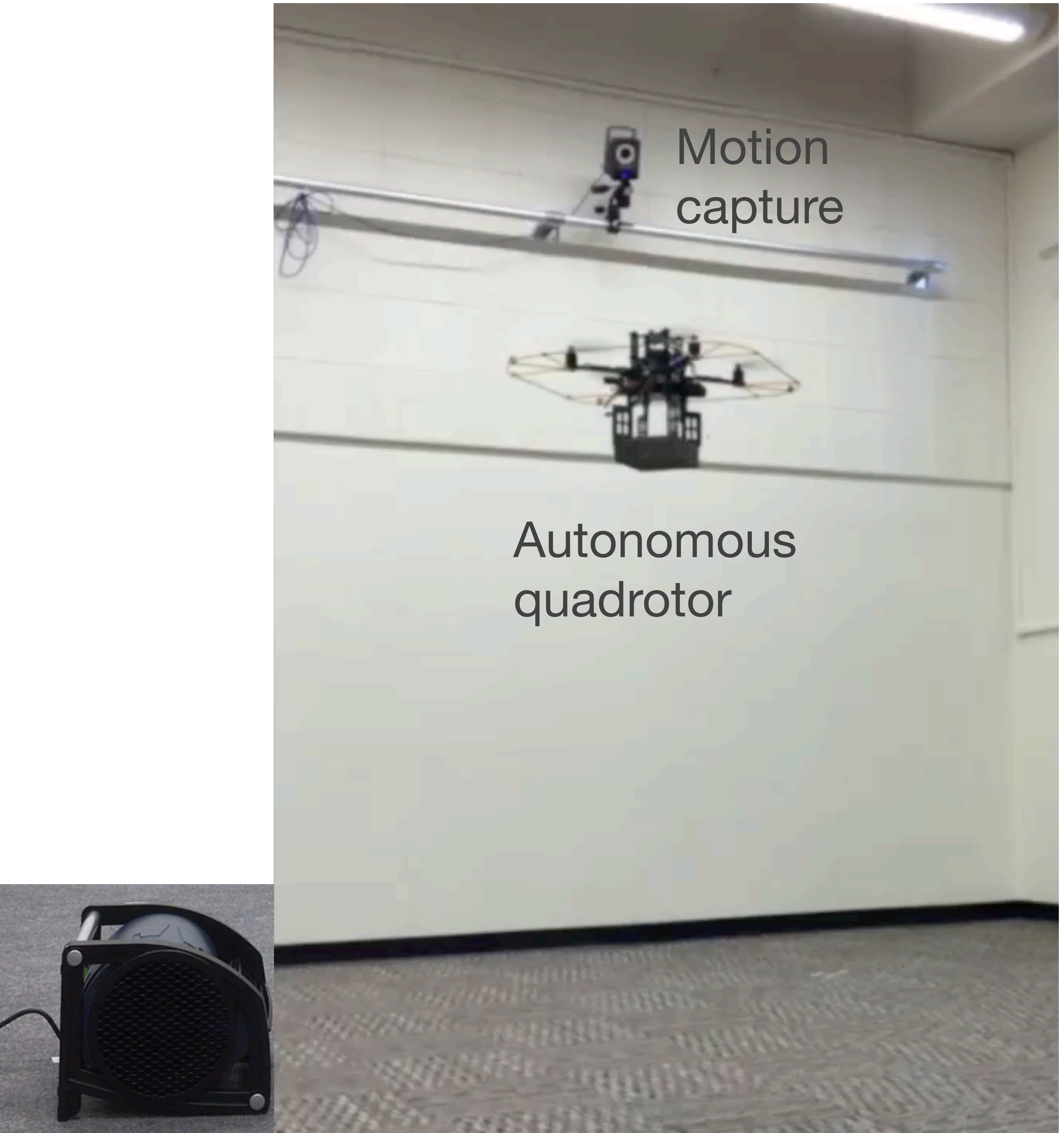
$$u \in \mathcal{U} \qquad d \in \hat{\mathcal{D}}(x)$$

Theorem: the least-restrictive control law

$$u(x) \in \begin{cases} \mathcal{U} & V(x) \geq \epsilon > 0 \\ \{u^*(x)\} & V(x) < \epsilon \end{cases}$$

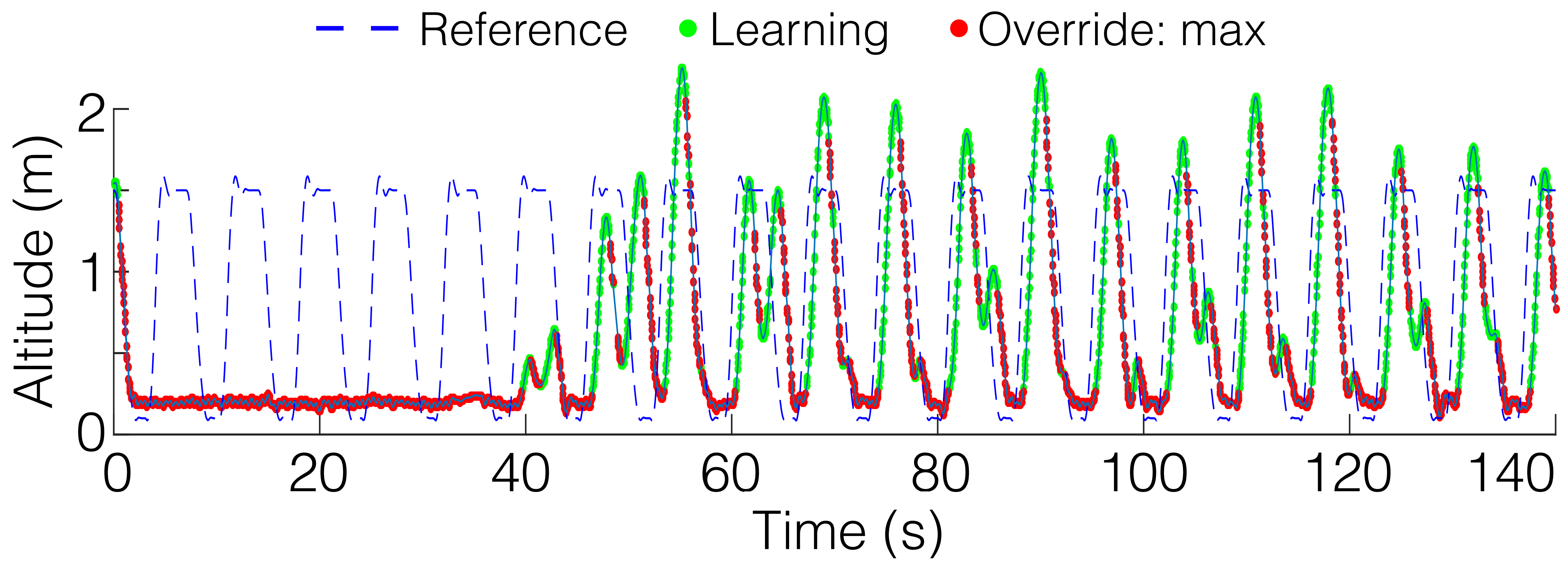
renders the ***safe set*** controlled-invariant if $d \in \operatorname{int}\hat{\mathcal{D}}(x)$ on the boundary $\{V(x) = 0\}$.

Learning With Safety Envelope Protection



Policy gradient reinforcement learning

(feature weights initialized to 0)



From Fall to Flight

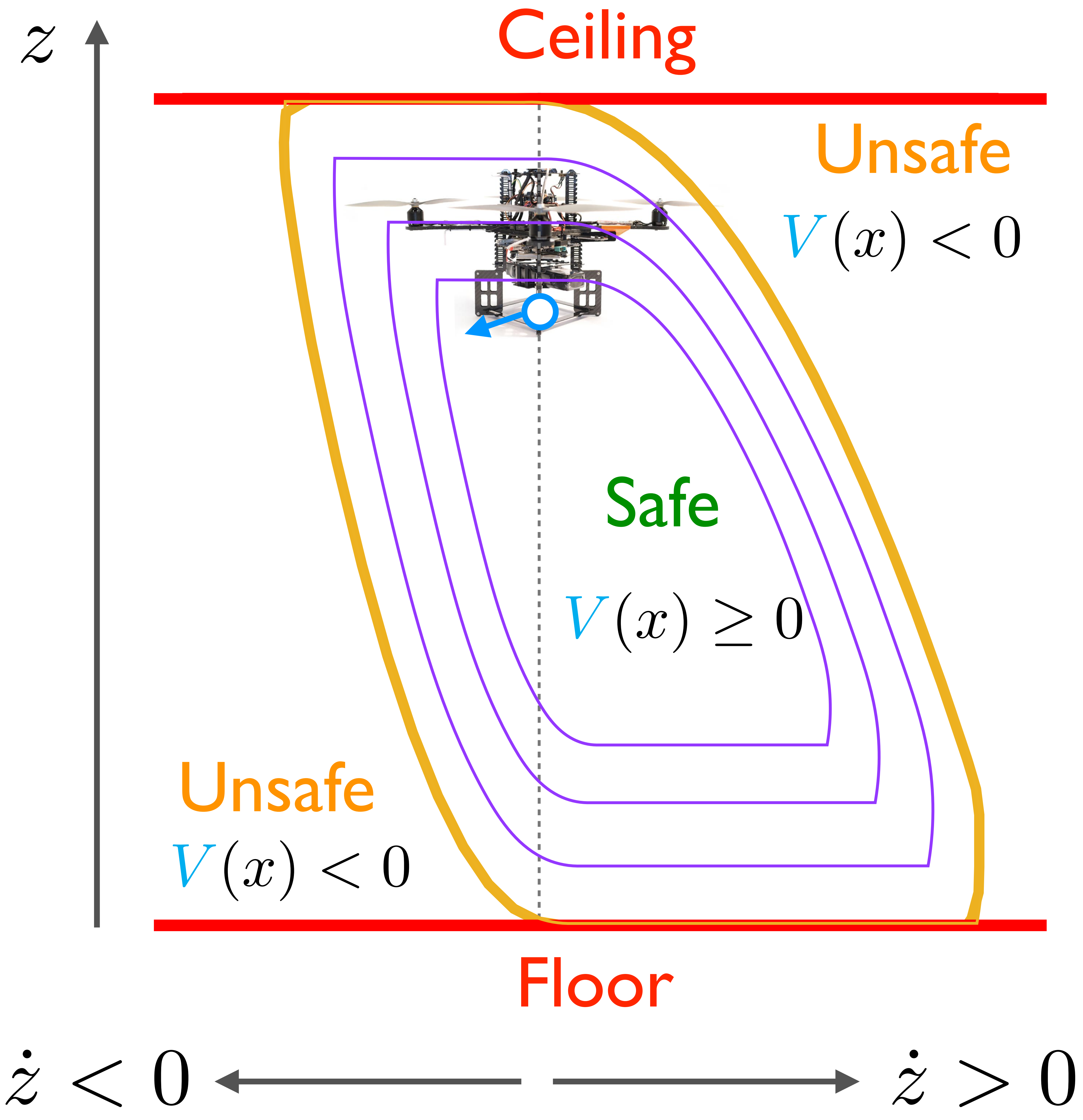
A guarantee is as good as the model it is based on.

Model error is inevitable in real-world environments.

Assurance: high-confidence statement about a *real system*
(typically based on model guarantees)

Guarantee: proven theoretical property of a *model*

The Safety Onion



$$\ddot{z} = k_T u - g + d$$

$$u \in \mathcal{U} \qquad d \in \hat{\mathcal{D}}(x)$$

Theorem: the least-restrictive control law

$$u(x) \in \begin{cases} \mathcal{U} & V(x) \geq \alpha + \epsilon \qquad \epsilon > 0 \\ \{u^*(x)\} & V(x) < \alpha + \epsilon \qquad \alpha \geq 0 \end{cases}$$

renders the α-*level set* controlled-invariant if $d \in \text{int}\hat{\mathcal{D}}(x)$ on the boundary $\{V(x) = \alpha\}$.

Use **data-driven analysis** to update the probability that each theoretical safety level curve is **usable for the real system.**

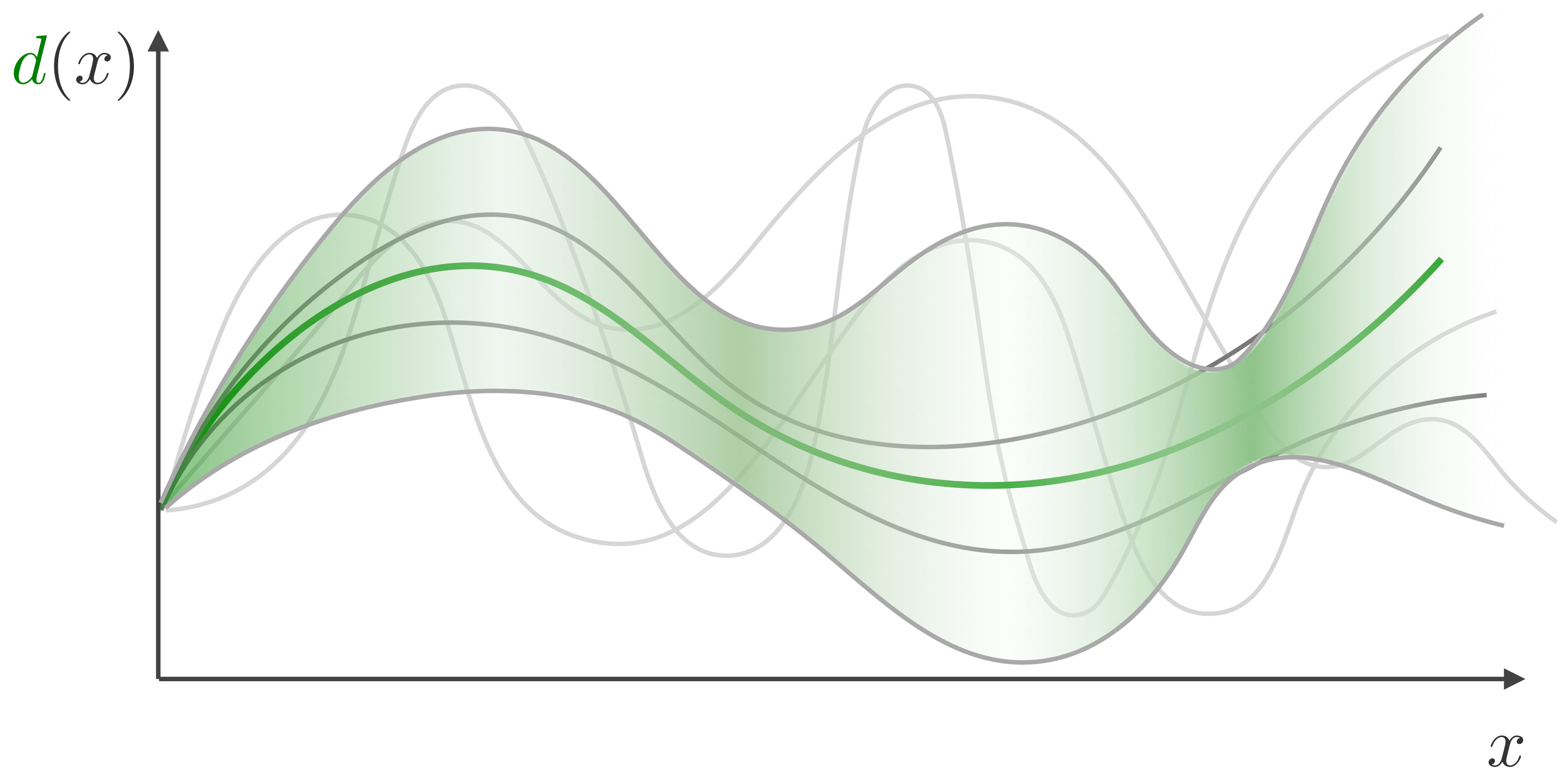
Bayesian Quantification of Model Error

$$\dot{x} = f(x, u, d)$$

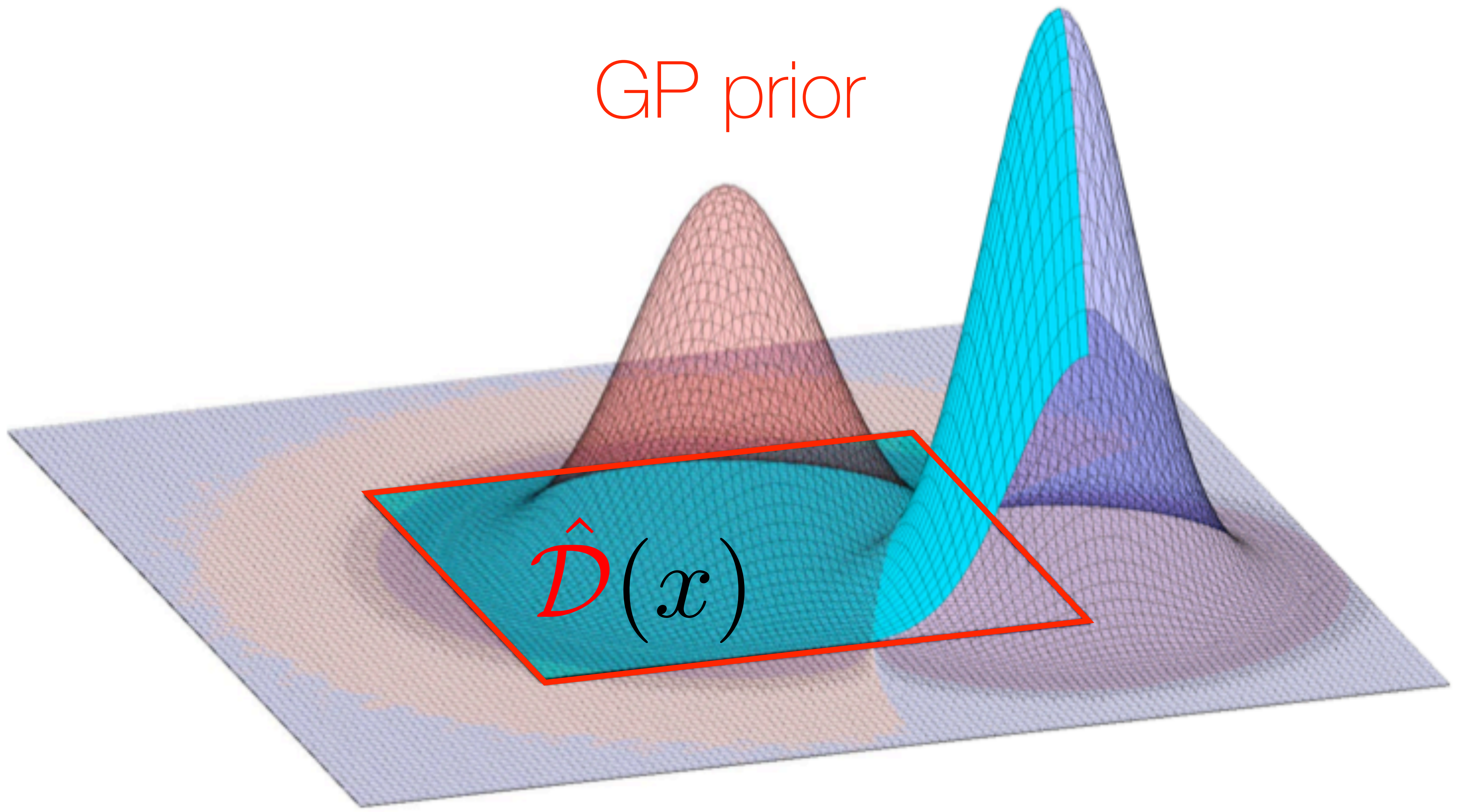
$$d \in \hat{\mathcal{D}}(x)$$

Observations: $\mathbf{X}, \mathbf{d}$

Gaussian process (GP)

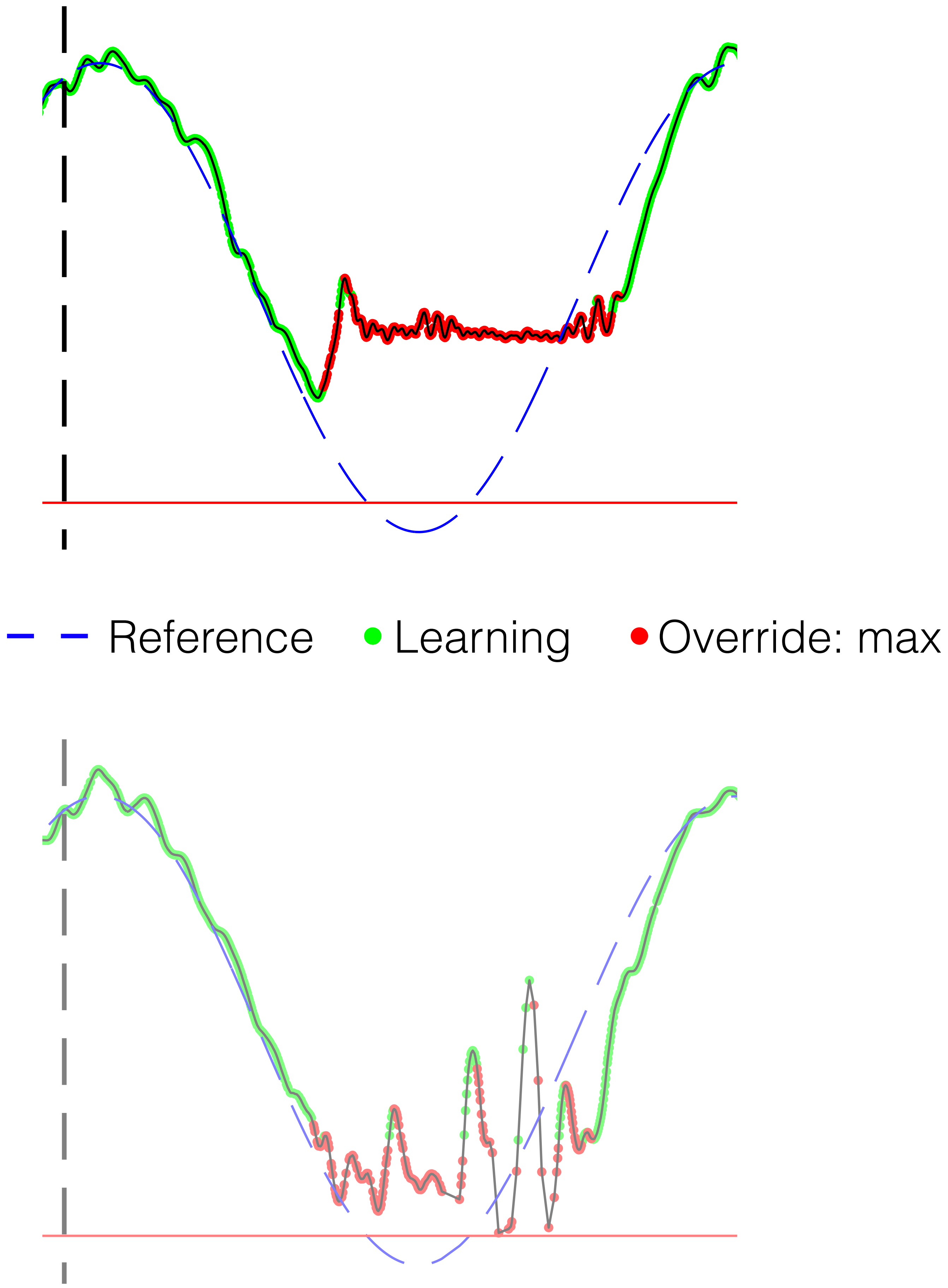


$$d(\cdot) \sim \mathcal{GP}\big(\mu(\cdot), k(\cdot, \cdot)\big)$$



$$P(d(x) \in \hat{\mathcal{D}}(x) | \mathbf{X}, \mathbf{d})$$

Preserving Safety in Unforeseen Conditions



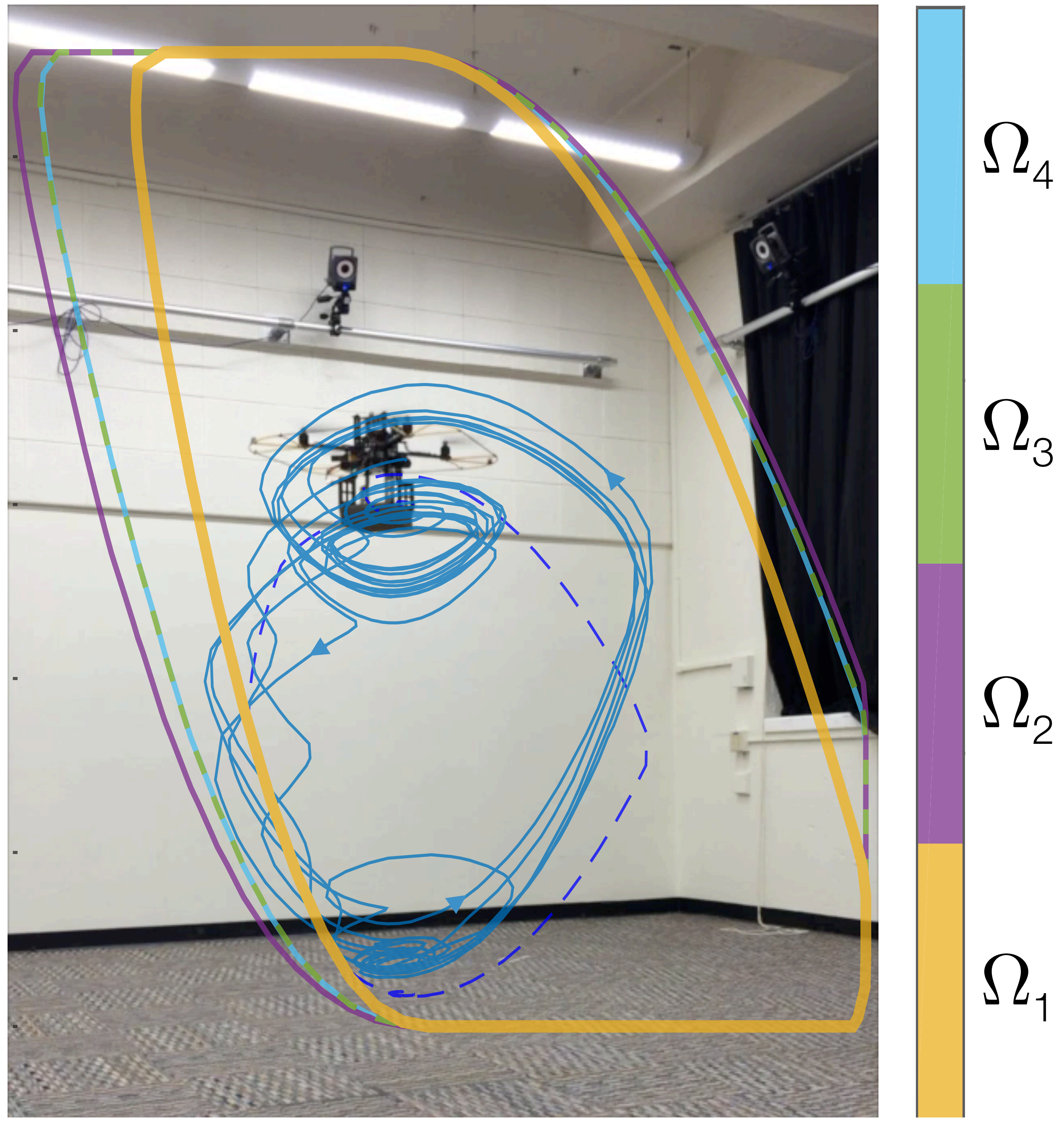
Bayesian safety validation

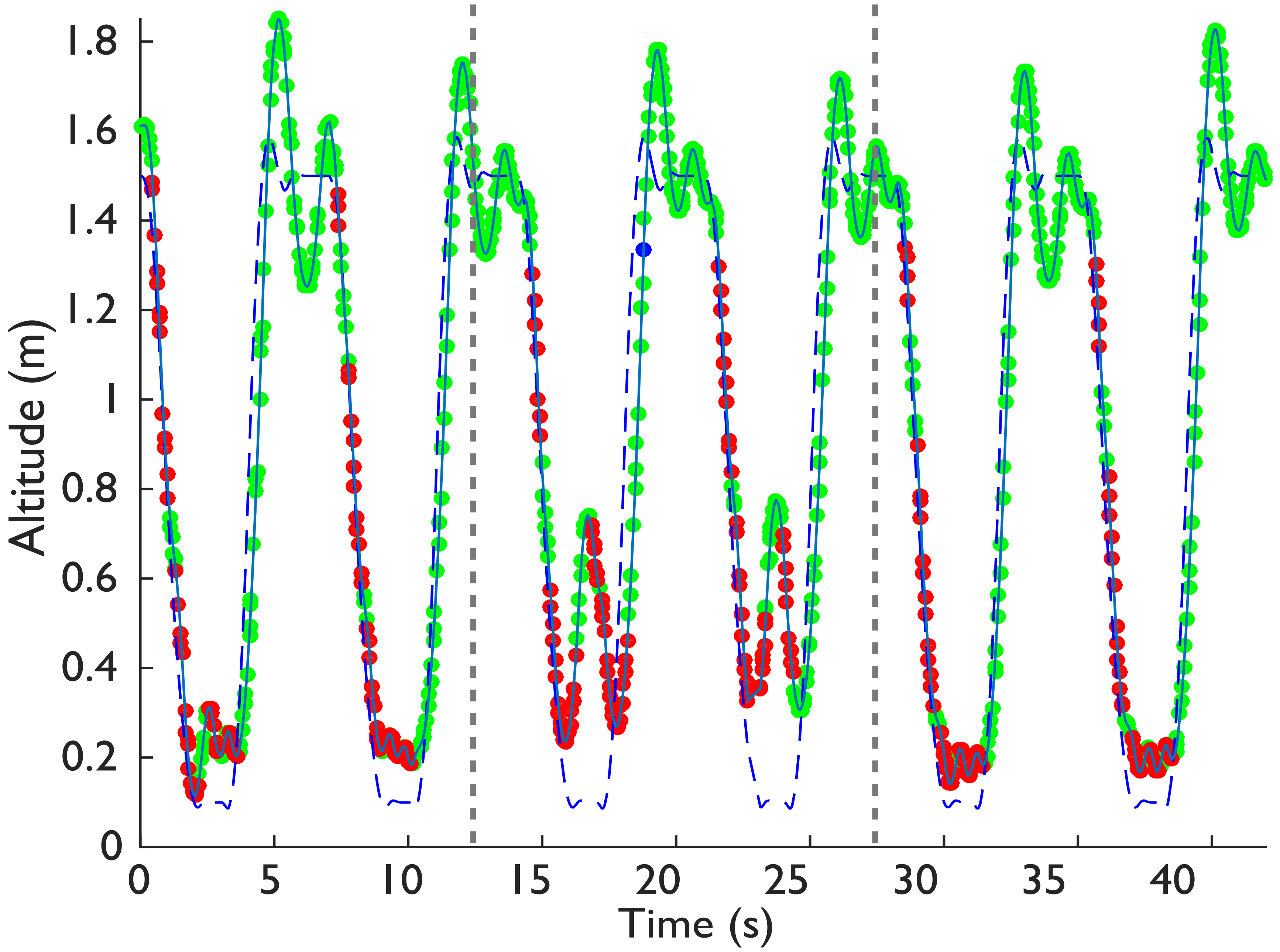
Assume safety analysis holds

Iteratively Recomputing Safety

As we acquire new information about the environment (in this case the unknown part $d(x)$ of the dynamics) we can recompute the safe set and safety policy.

At the same time, online (local) safety validation helps prevent reliance on any overly optimistic analysis resulting from incomplete or misleading exploration.





Hamilton-Jacobi safety analysis is a powerful decision-making tool.

It is also extremely costly for high-dimensional nonlinear systems.

What if we don’t have a Hamilton-Jacobi safety value function?

Is there a next best thing?

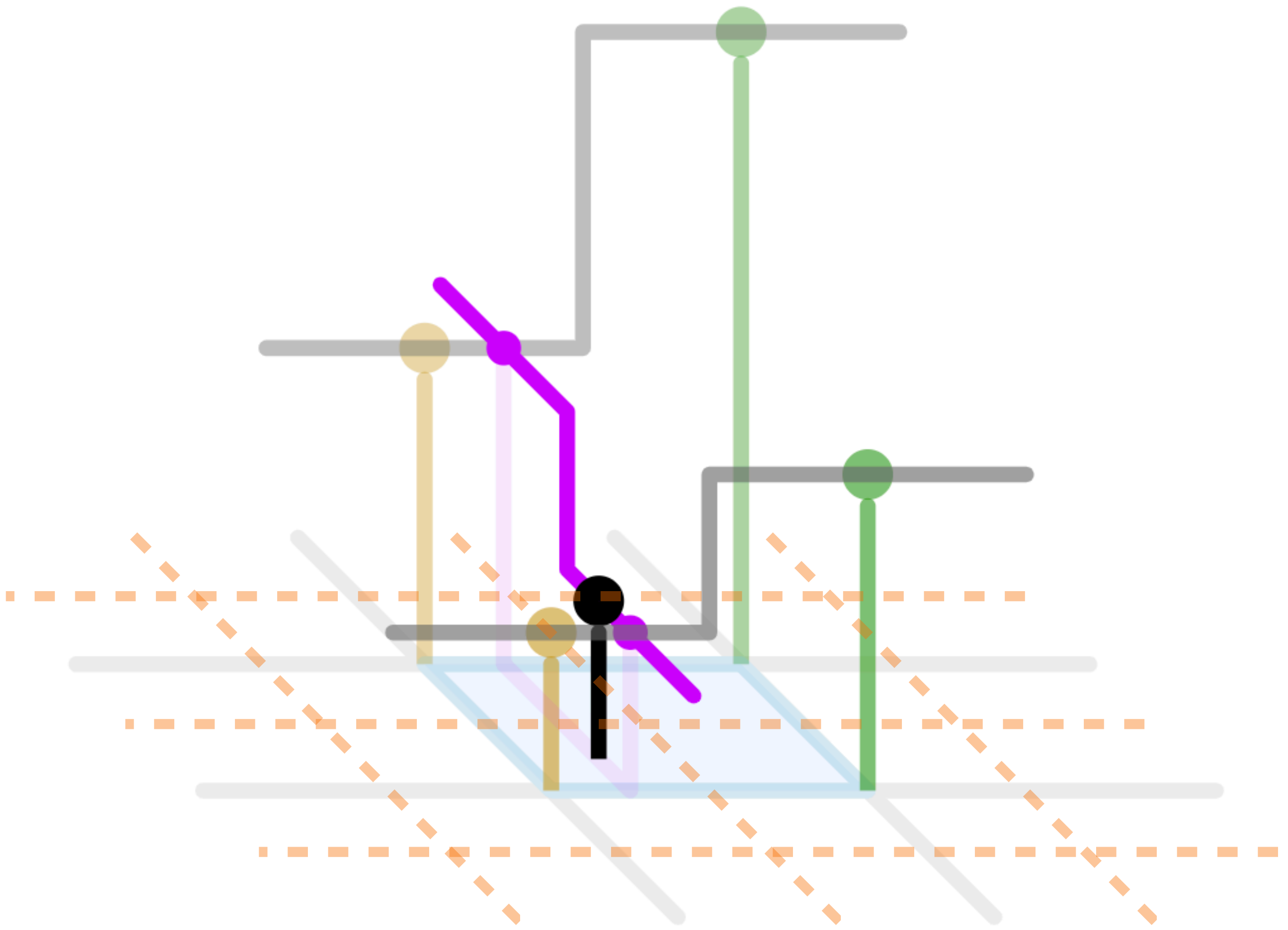
Empirical Safety: Rapid Motor Adaptation

Can we use **learning** to compute best-effort safety controllers

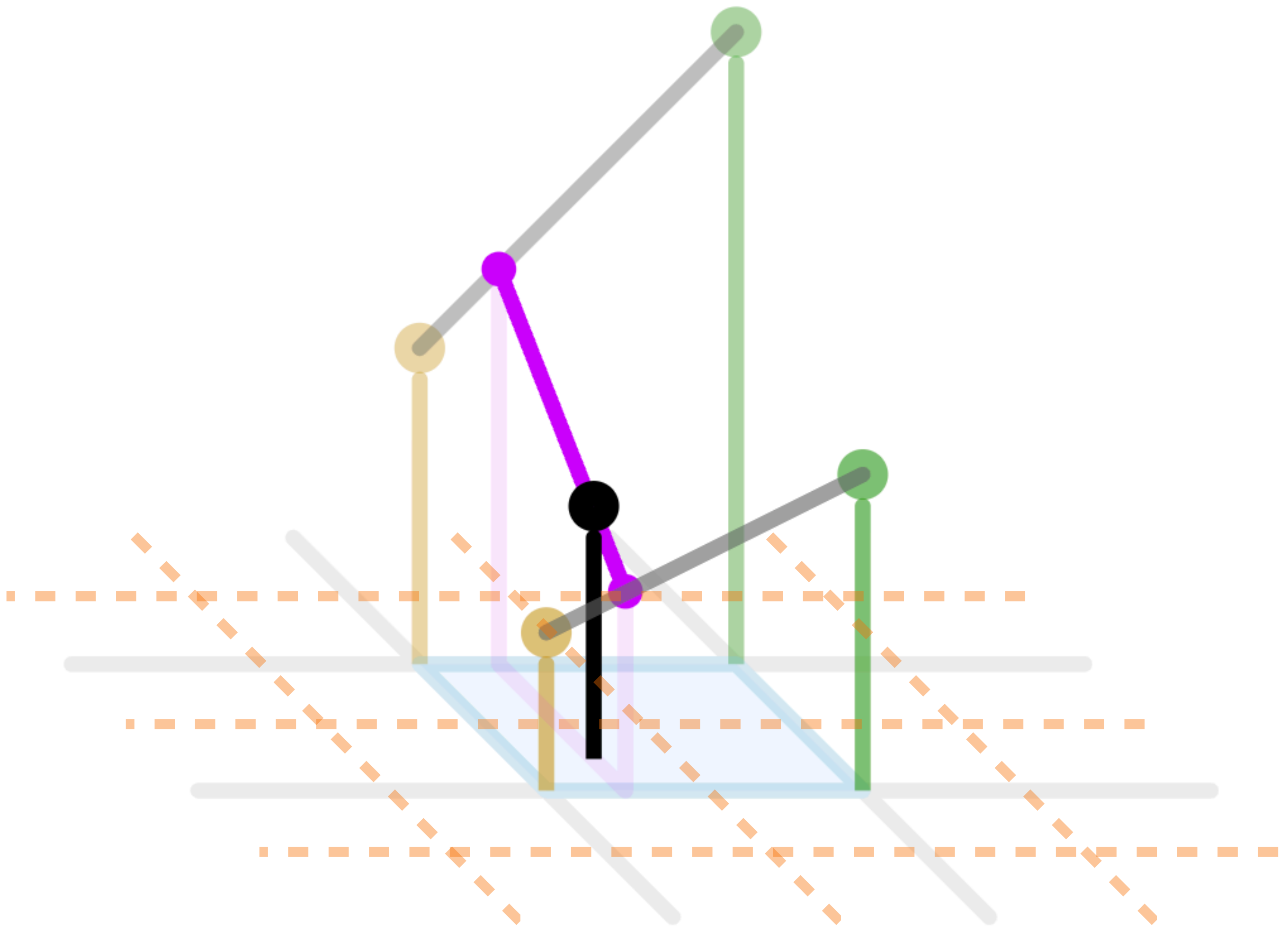
with sound **guarantees**?

From Grids to Neural Networks

A grid discretization is essentially a (rather inefficient) function approximator.



Nearest-neighbor interpolation



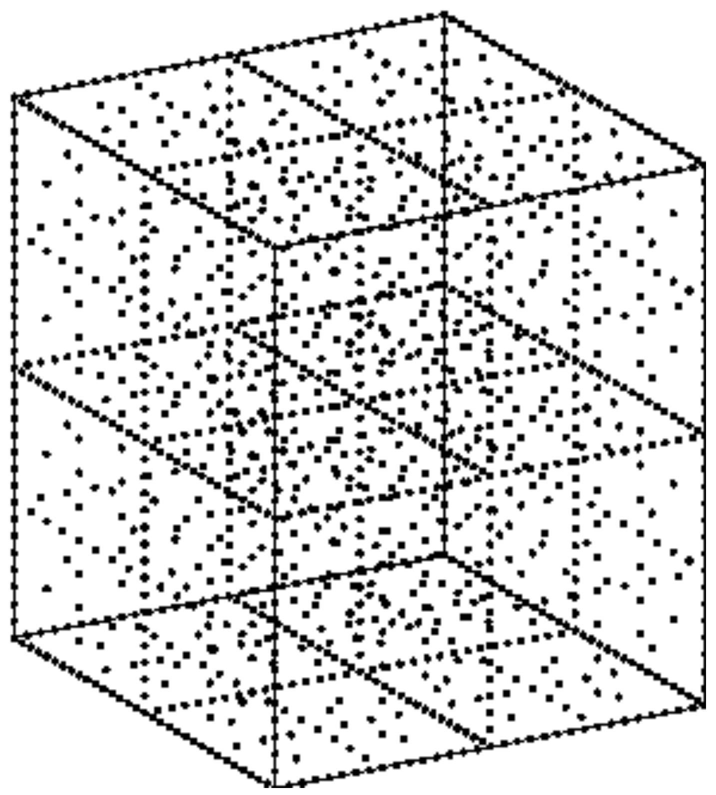
Multilinear interpolation

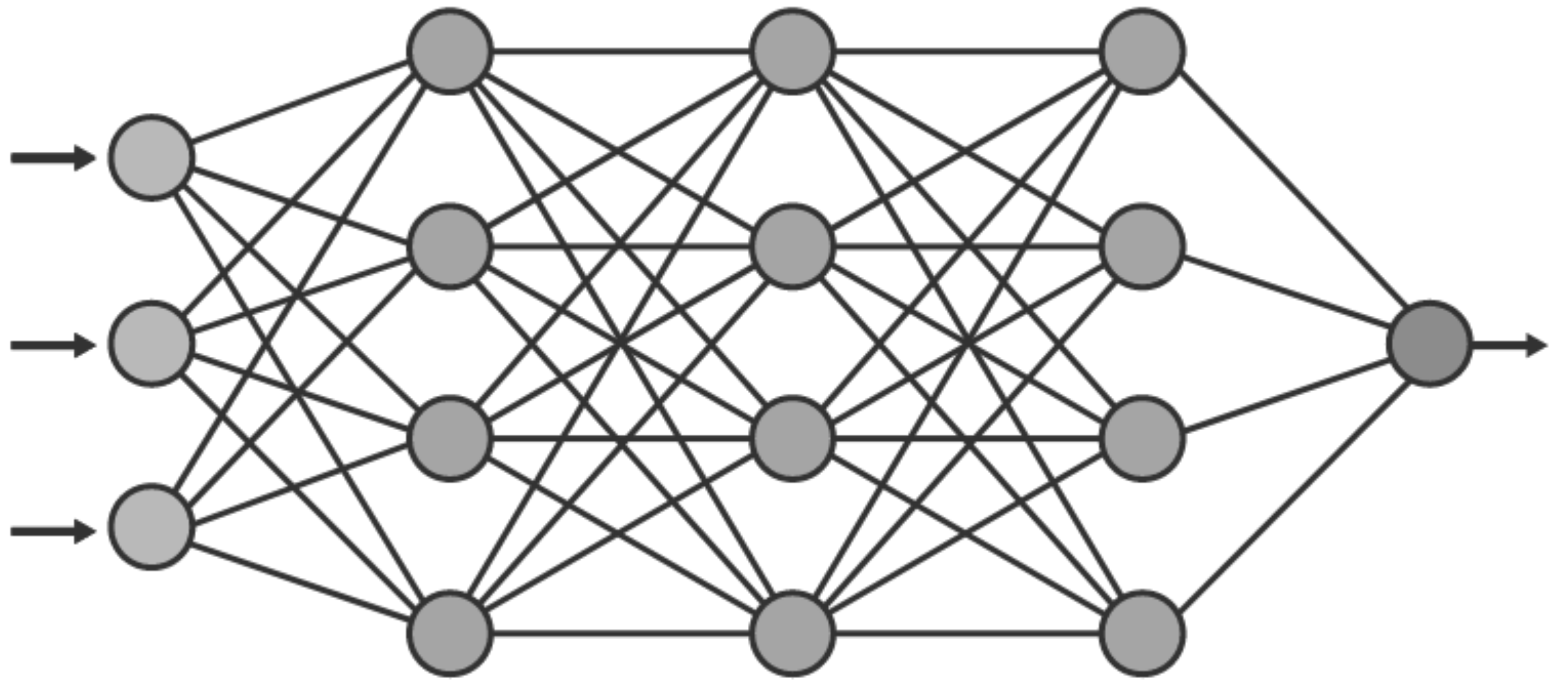
Can we represent the value function/optimal policy in a more scalable form?

From Grids to Neural Networks

A neural network is a "universal" gradient-trainable function approximator.

can represent any continuous function to arbitrary accuracy with enough units *(Universal Representation Theorem)*





"Universal" approximator ✓

Resolution-complete methods ✓

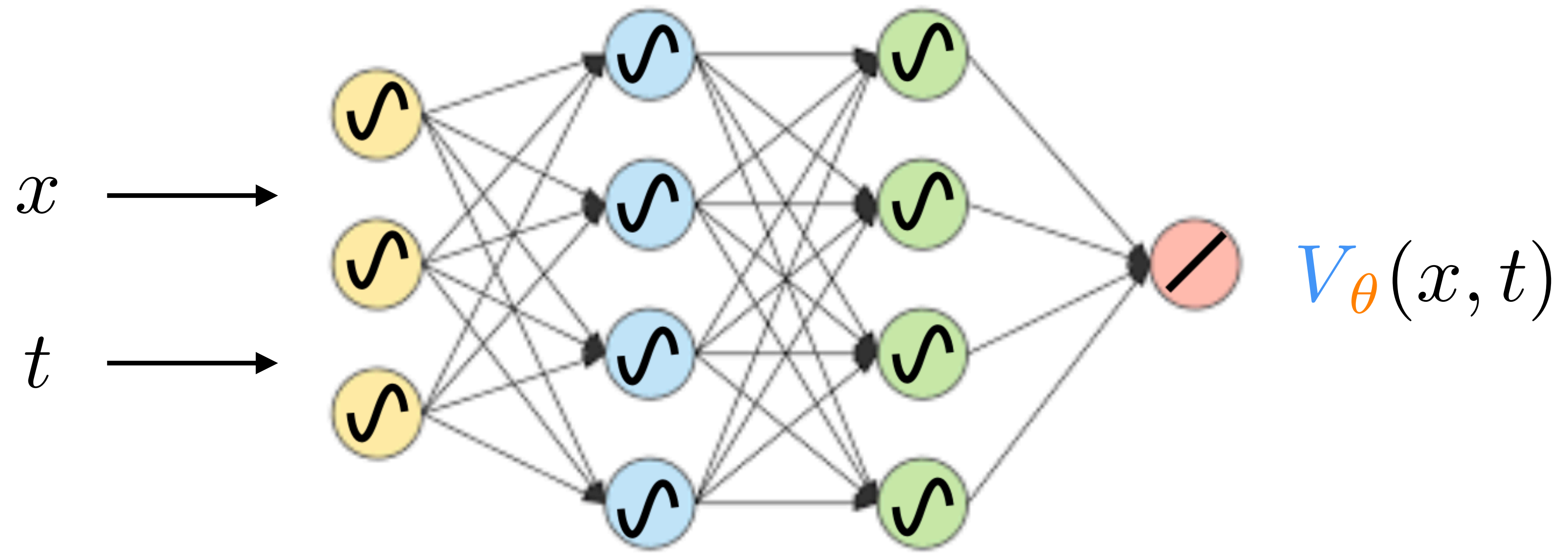
"Universal" approximator ✓

Resolution-complete methods ?

Similar to deep RL, we can *approximate* the safe control problem using neural networks.

Self-Supervised Value Learning: DeepReach

Deep representation: the value function is represented through a neural network V_θ.



$$\min\left\{\partial_t V + \max_u \min_d \nabla_x V^\top f(x,u,d), l(x,t) - V(x,t)\right\} = 0$$

$$V(\cdot,T) \equiv l(\cdot,T)$$

HJI safety equation (variational inequality)

Loss function: penalizes HJI error incurred by the learned value function V_θ.

$$L(\theta) := \sum_i \left|\min\left\{\partial_t V_\theta + \max_u \min_d \nabla_x V_\theta^\top f(x_i,u,d), l(x_i,t_i) - V_\theta(x_i,t_i)\right\}\right| + \lambda\left|l(x_i,T) - V_\theta(x_i,T)\right|$$

Training loss

Self-supervision: repeatedly sample a batch of space-time points $\{(x_i,t_i)\}$ and update V_θ.

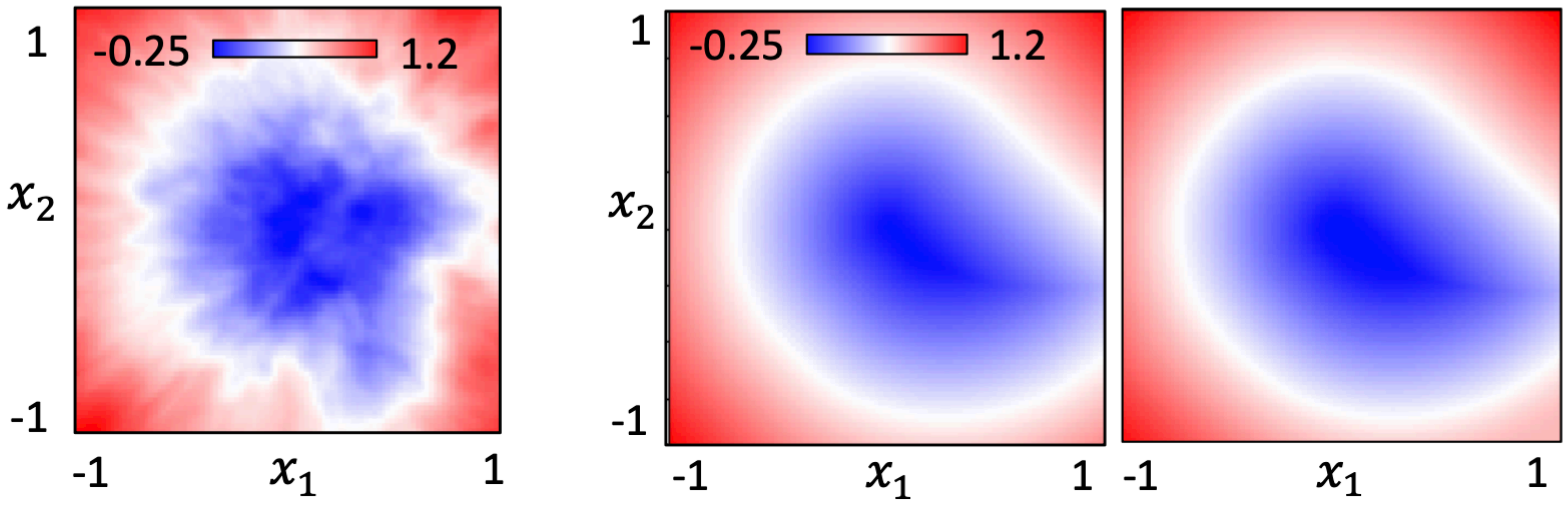
$$\theta \leftarrow \theta - \alpha \nabla L(\theta)$$

Update rule

Self-Supervised Value Learning: DeepReach

Sinusoidal networks: sinusoidal activation function works well in practice.

Value Function Comparison



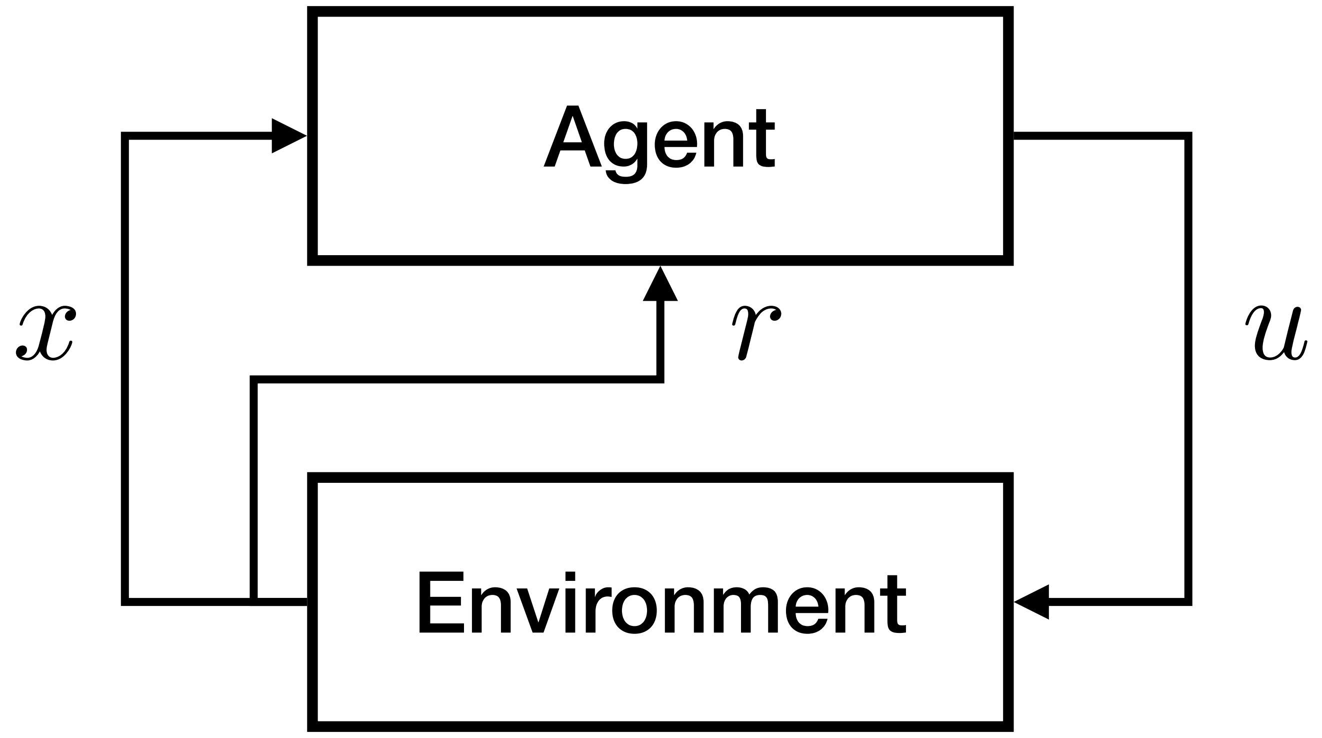
DeepReach + ReLU *Explicit PDE Solution* *DeepReach*

Comparison graphic by Somil Bansal (USC).

Other activation functions struggle to approximate the value function well

[Bansal and Tomlin. DeepReach: A Deep Learning Approach to High-Dimensional Reachability. ICRA 2021]

Reinforcement Learning: Beyond Rewards



Performance

$$J(\mathbf{x}) = \sum_{t=0}^{\infty} \gamma^t r\big(\mathbf{x}(t)\big)$$

$$V(x) = \max_{u \in \mathcal{U}} \; r(x,u) + \gamma V(x')$$

$$V(x) = \max_{u \in \mathcal{U}} \; (1-\gamma) r(x,u) + \gamma \Big(r(x,u) + V(x') \Big)$$

Contraction mapping

Safety

$$J(\mathbf{x}) = \inf_{t \geq 0} l\big(\mathbf{x}(t)\big)$$

$$V(x) = \min \Big\{ l(x), \max_{u \in \mathcal{U}} V(x') \Big\}$$

Not a contraction mapping

How do we discount a minimum?

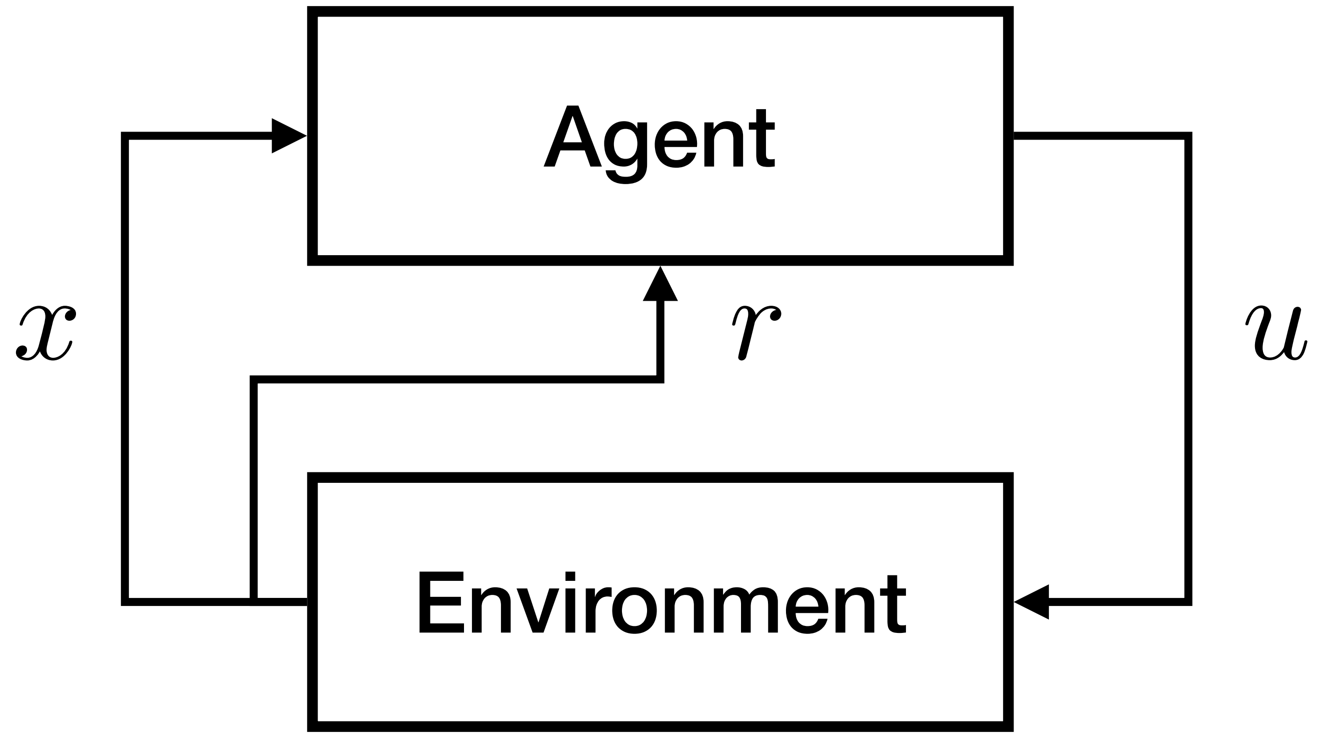
$$V(x) = \min\left\{ l(x), \max_{u \in \mathcal{U}} V(x') \right\}$$

$$V(x) = (1-\gamma)l(x) + \gamma \min\left\{ l(x), \max_{u \in \mathcal{U}} V(x') \right\}$$

$$\gamma \in [0, 1)$$

[Fisac*, Lugovoy*, et al. Bridging Hamilton-Jacobi safety analysis and reinforcement learning. ICRA 2019]

Reinforcement Learning: Beyond Rewards



Performance

$$J(\mathbf{x}) = \sum_{t=0}^{\infty} \gamma^t r\big(\mathbf{x}(t)\big)$$

$$V(x) = \max_{u \in \mathcal{U}} \; r(x,u) + \gamma V(x')$$

$$V(x) = \max_{u \in \mathcal{U}} \; (1-\gamma) r(x,u) + \gamma \Big(r(x,u) + V(x') \Big)$$

Contraction mapping

Safety

$$J(\mathbf{x}) = \inf_{t \geq 0} l\big(\mathbf{x}(t)\big)$$

$$V(x) = \min \Big\{ l(x), \max_{u \in \mathcal{U}} V(x') \Big\}$$

$$V(x) = (1-\gamma) l(x) + \gamma \min \Big\{ l(x), \max_{u \in \mathcal{U}} V(x') \Big\}$$

Contraction mapping

Contraction Mapping Result

Theorem: The time-discounted safety Bellman equation

$$V(x) = (1-\gamma)l(x) + \gamma \min\left\{l(x), \max_{u\in\mathcal{U}} V(x')\right\}$$

induces a *contraction* in the space of value functions under the supremum norm.

Let $V, \tilde{V} : \mathcal{X} \to \mathbb{R}$ then there exists a constant $\kappa \in [0,1)$ such that

$$\|B[V] - B[\tilde{V}]\|_\infty \leq \kappa \|V - \tilde{V}\|_\infty .$$

Proof: For *all* states $x \in \mathcal{X}$, the following bounds hold:

$$|B[V](x) - B[\tilde{V}](x)| = \gamma |\min\{l(x), \max_{u\in\mathcal{U}} V\big(x + f(x,u)\Delta t\big)\} - \min\{l(x), \max_{\tilde{u}\in\mathcal{U}} \tilde{V}\big(x + f(x,\tilde{u})\Delta t\big)\}|$$

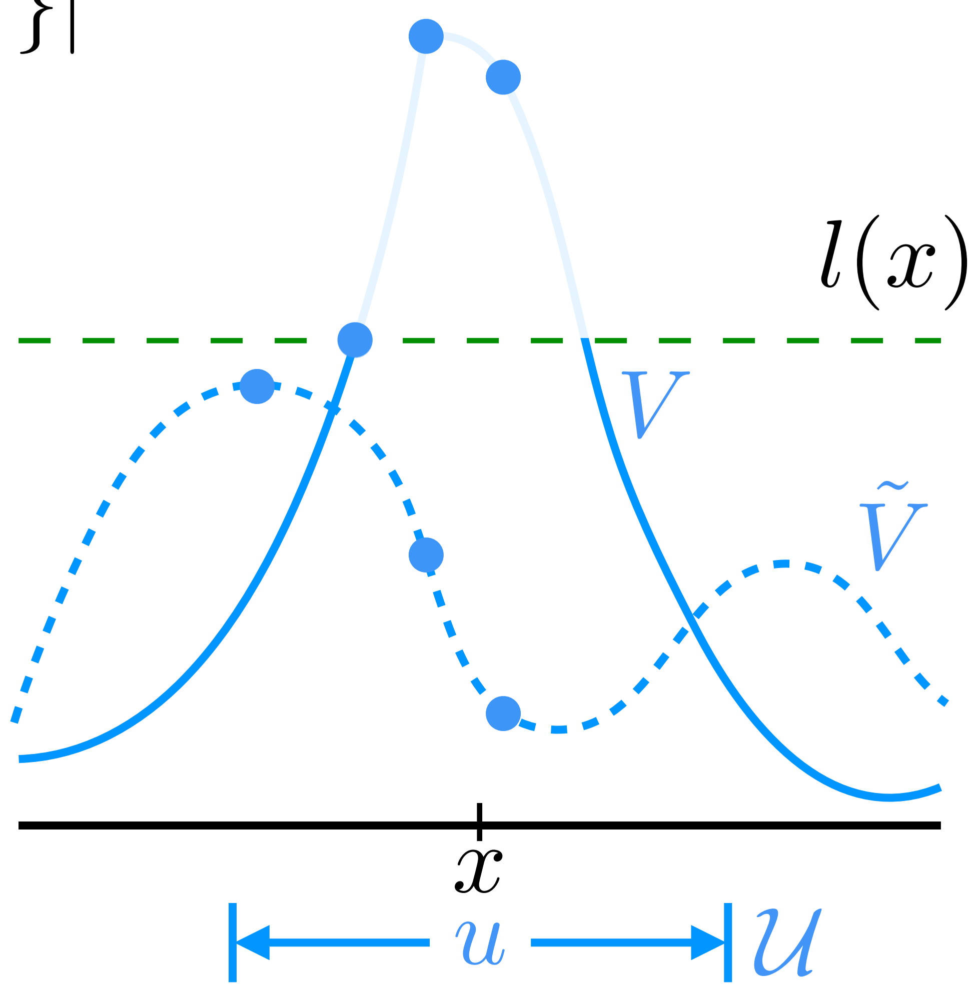
Let the first max be greater, and achieved by $u^* \in \mathcal{U}$.

$$\leq \gamma |\max_{u\in\mathcal{U}} V\big(x + f(x,u)\Delta t\big) - \max_{\tilde{u}\in\mathcal{U}} \tilde{V}\big(x + f(x,\tilde{u})\Delta t\big)|$$

$$\leq \gamma |V\big(x + f(x,u^*)\Delta t\big) - \tilde{V}\big(x + f(x,u^*)\Delta t\big)|$$

$$\leq \gamma \max_{u\in\mathcal{U}} |V\big(x + f(x,u)\Delta t\big) - \tilde{V}\big(x + f(x,u)\Delta t\big)|$$

$$\leq \gamma \sup_{\tilde{x}} |V(\tilde{x}) - \tilde{V}(\tilde{x})| = \gamma \|V - \tilde{V}\|_\infty \quad \square$$

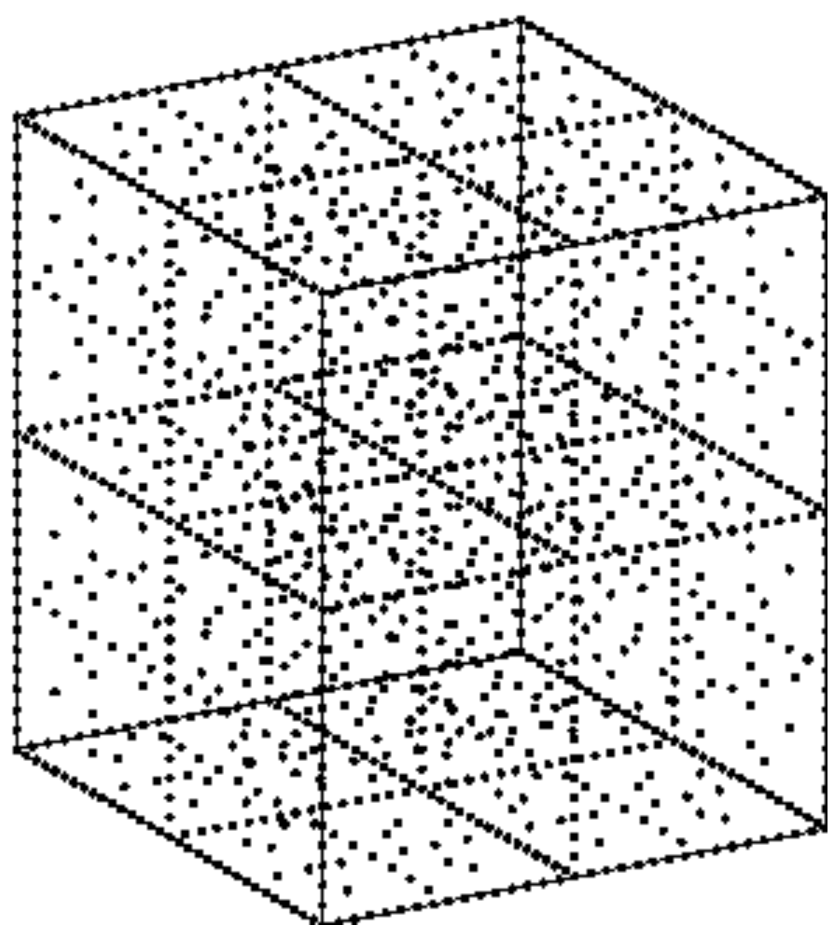


Safety Q-Learning

Time-discounted state-action Safety Bellman Equation

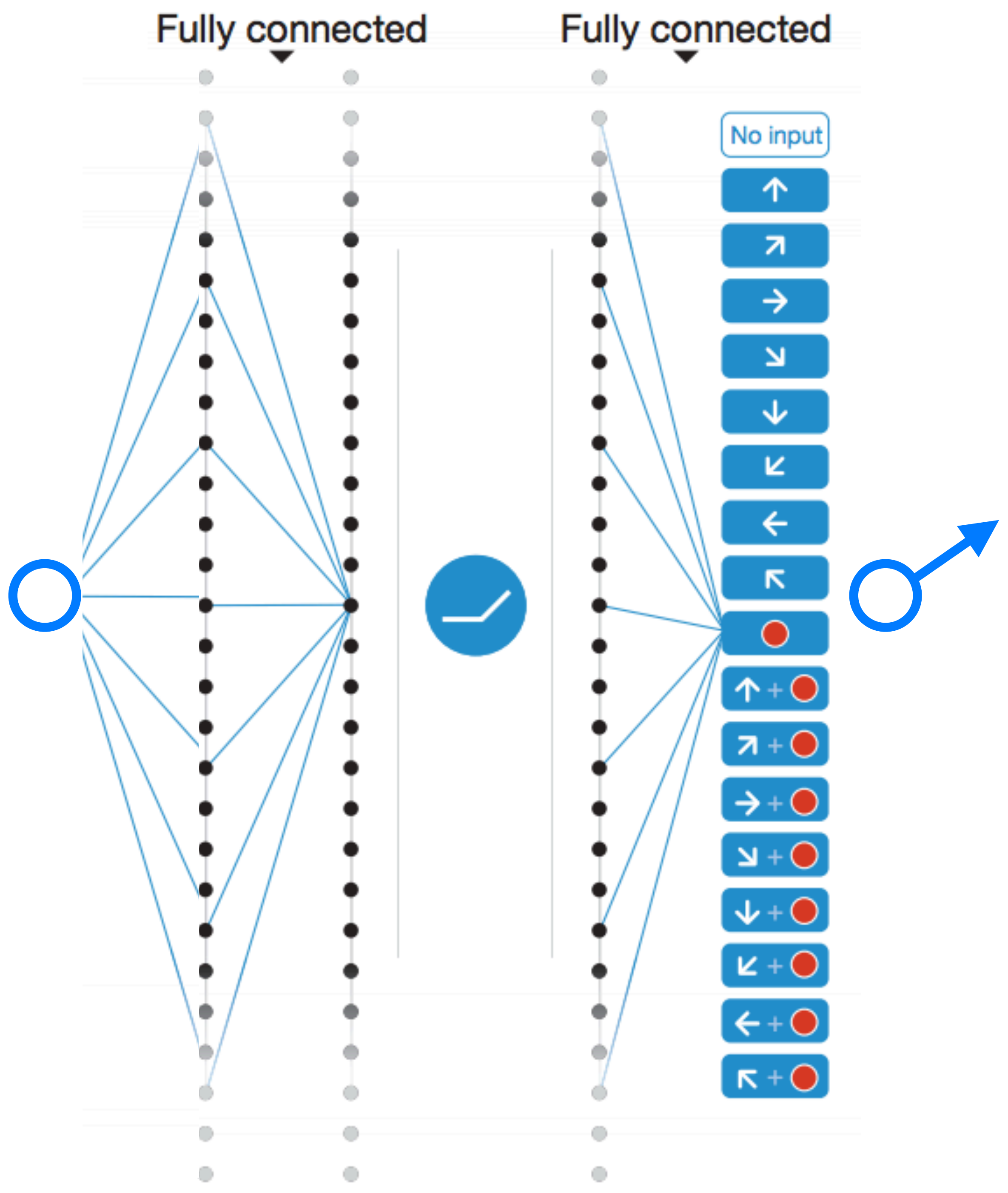
$$Q(x,u) \leftarrow (1-\alpha)Q(x,u) + \alpha\Big[(1-\gamma)l(x) + \gamma \min\big\{l(x), \max_{u' \in \mathcal{U}} Q(x',u')\big\}\Big]$$

Tabular Q-Learning



[Watkins and Dayan. 1992]

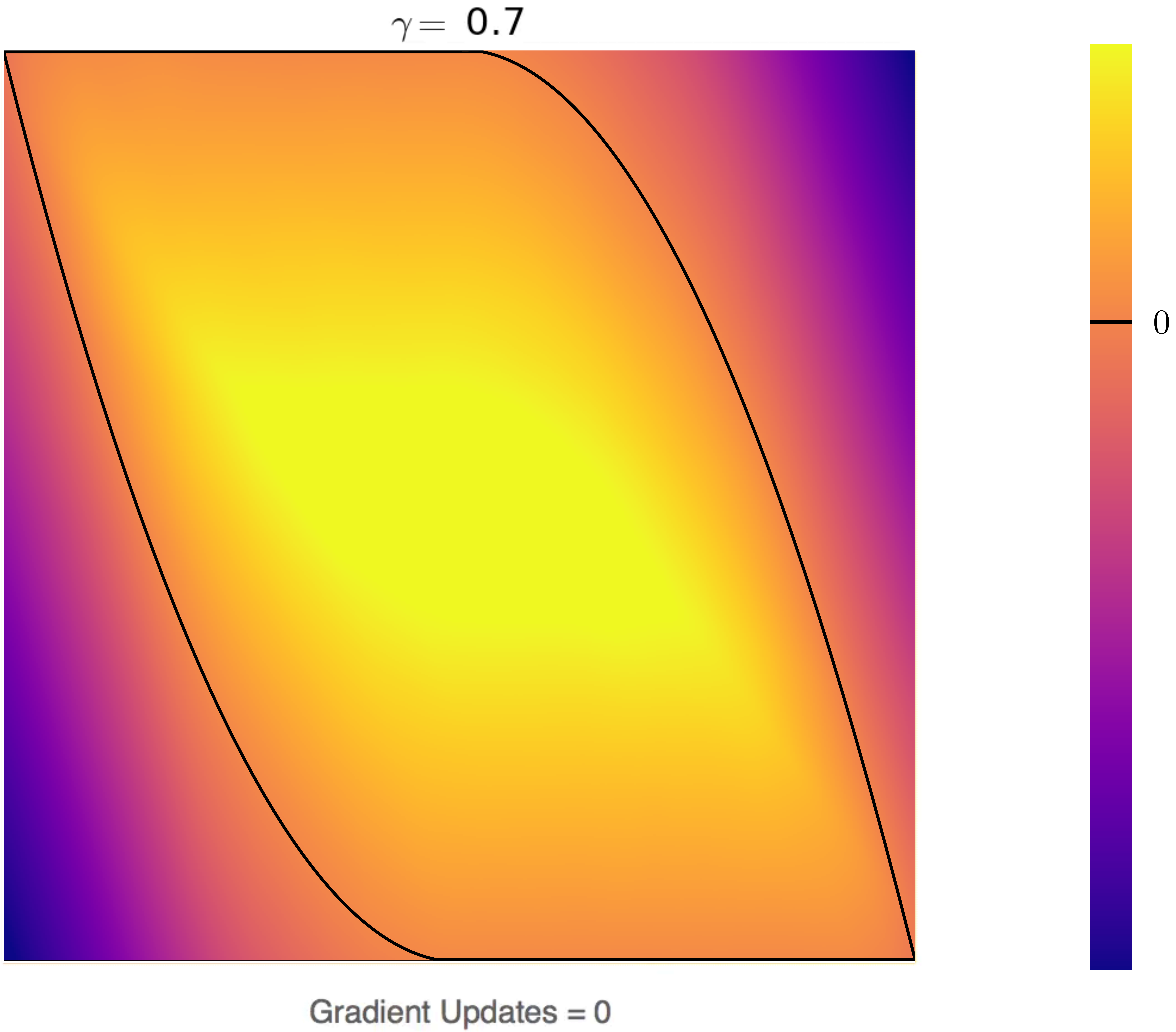
Deep Q-Learning



[Mnih et al. 2015]

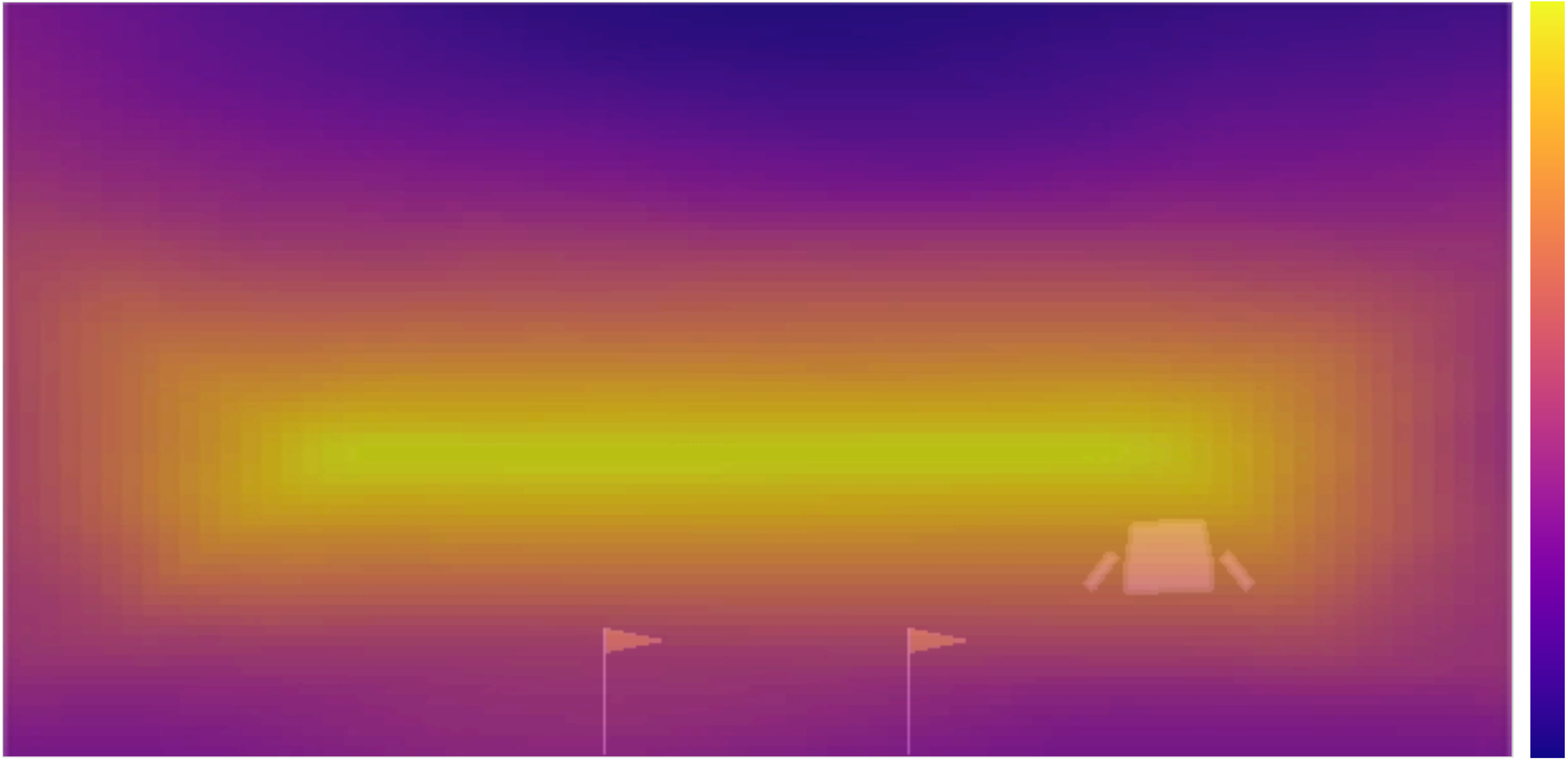
Deep Q-Learning

Double Integrator (2D)



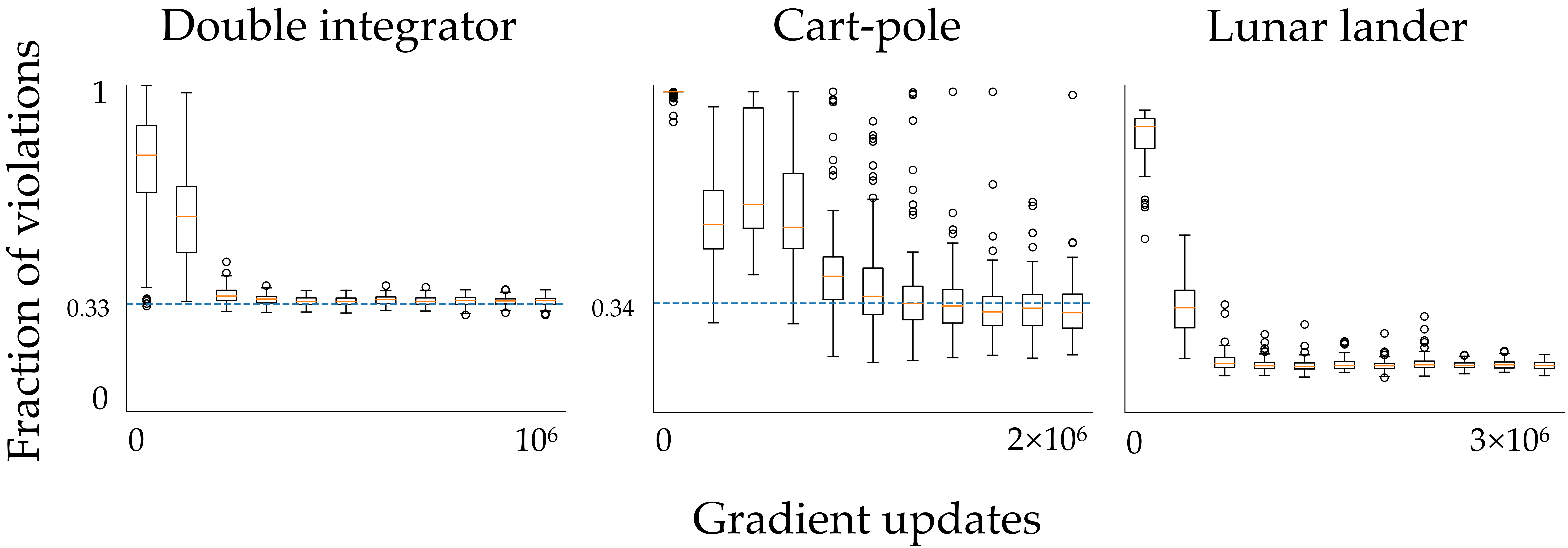
Deep Q-Learning

Lunar Lander (6D)



Learning to stay safe if staying safe is possible

Sample initial states uniformly throughout the state space and apply the learned safety policy.



Fraction of violations under the learned policy decreases towards volume of the true unsafe set.

Sum of rewards (Lagrange objective)

Discounted safety Bellman equation

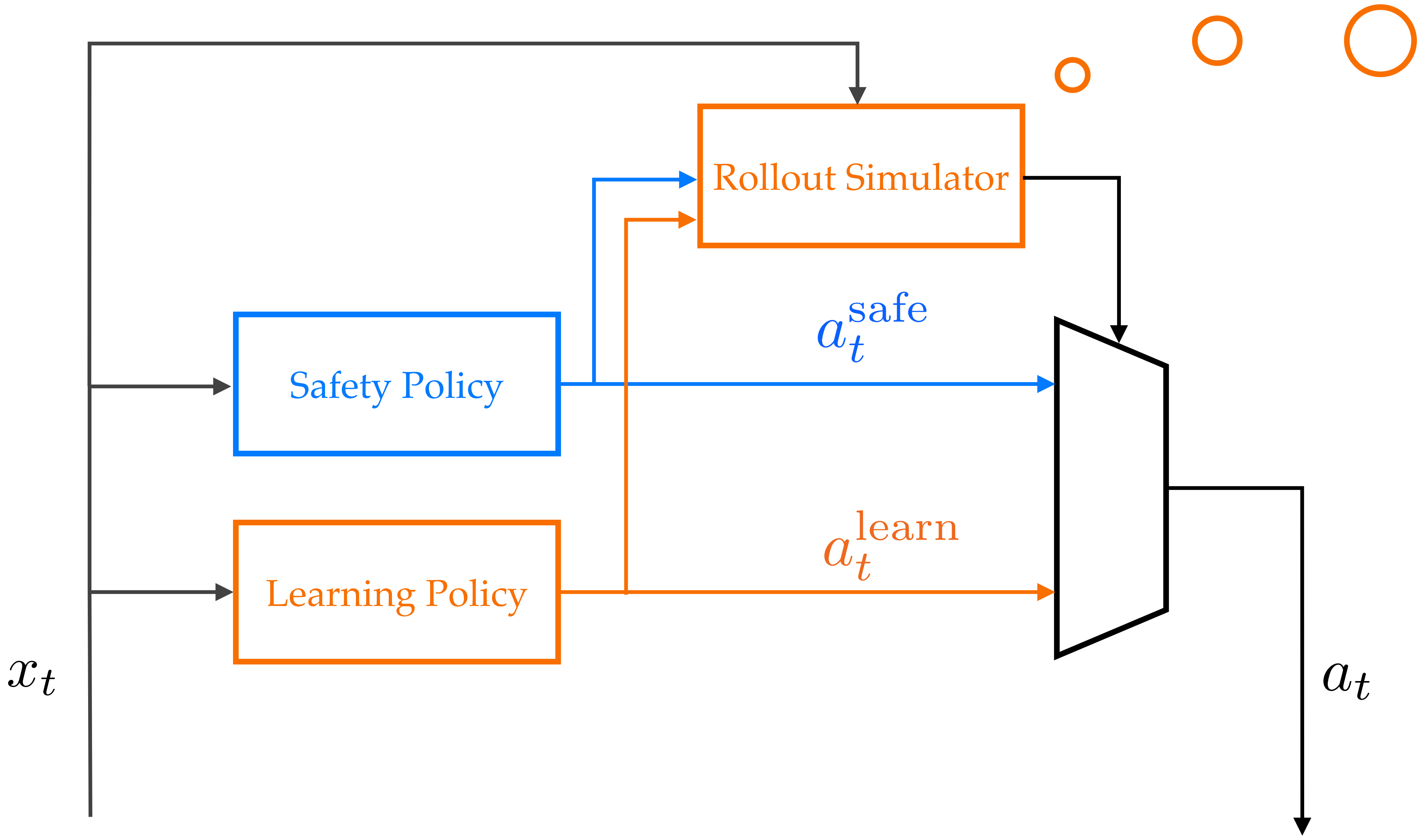
Discounted safety Bellman equation

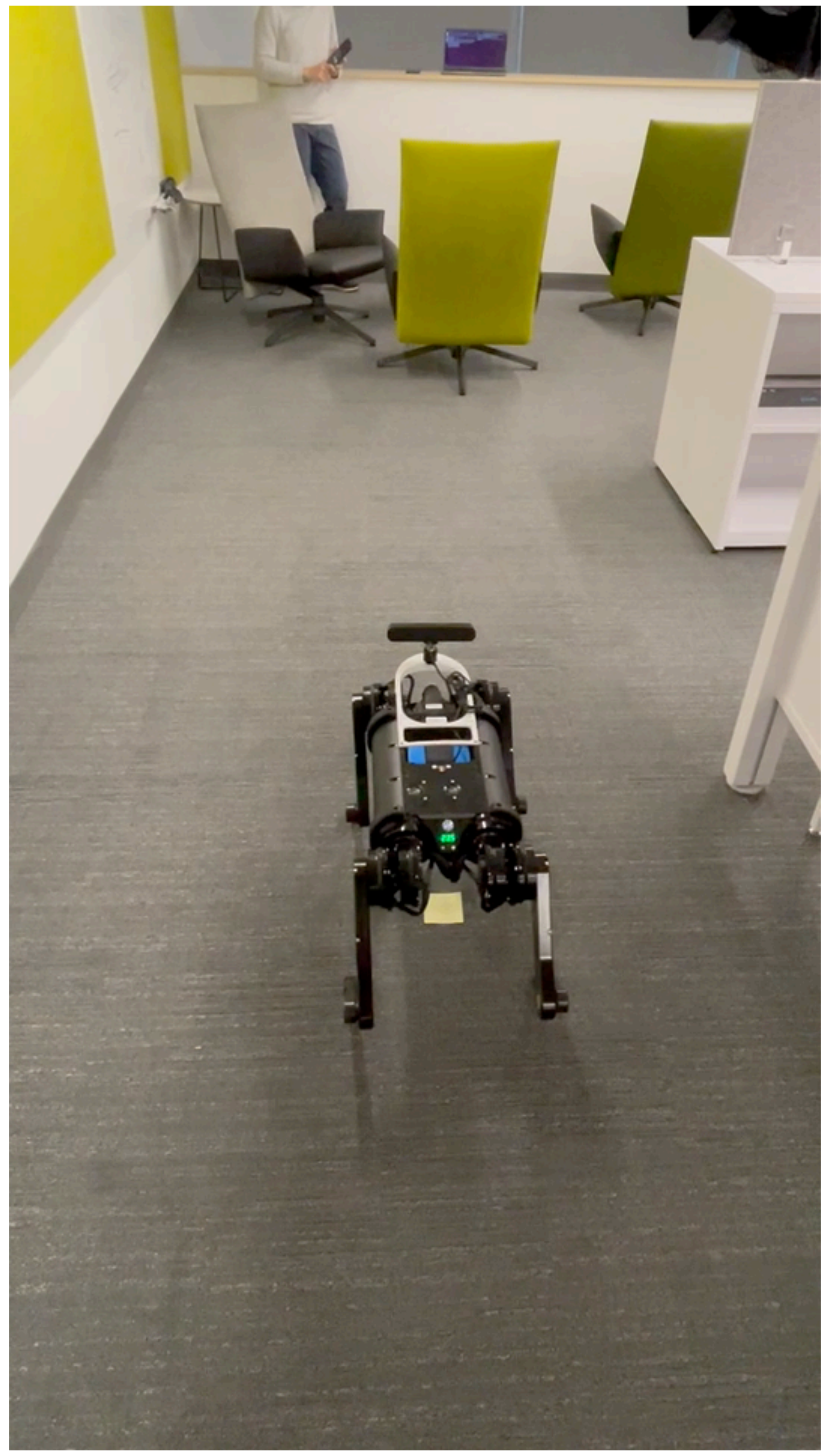
Discounted safety Bellman equation

Works well in practice, but what about guarantees?

Do we have safety filters that work with an arbitrary safety policy?

Safety RL Meets Shielding!





Navigation task

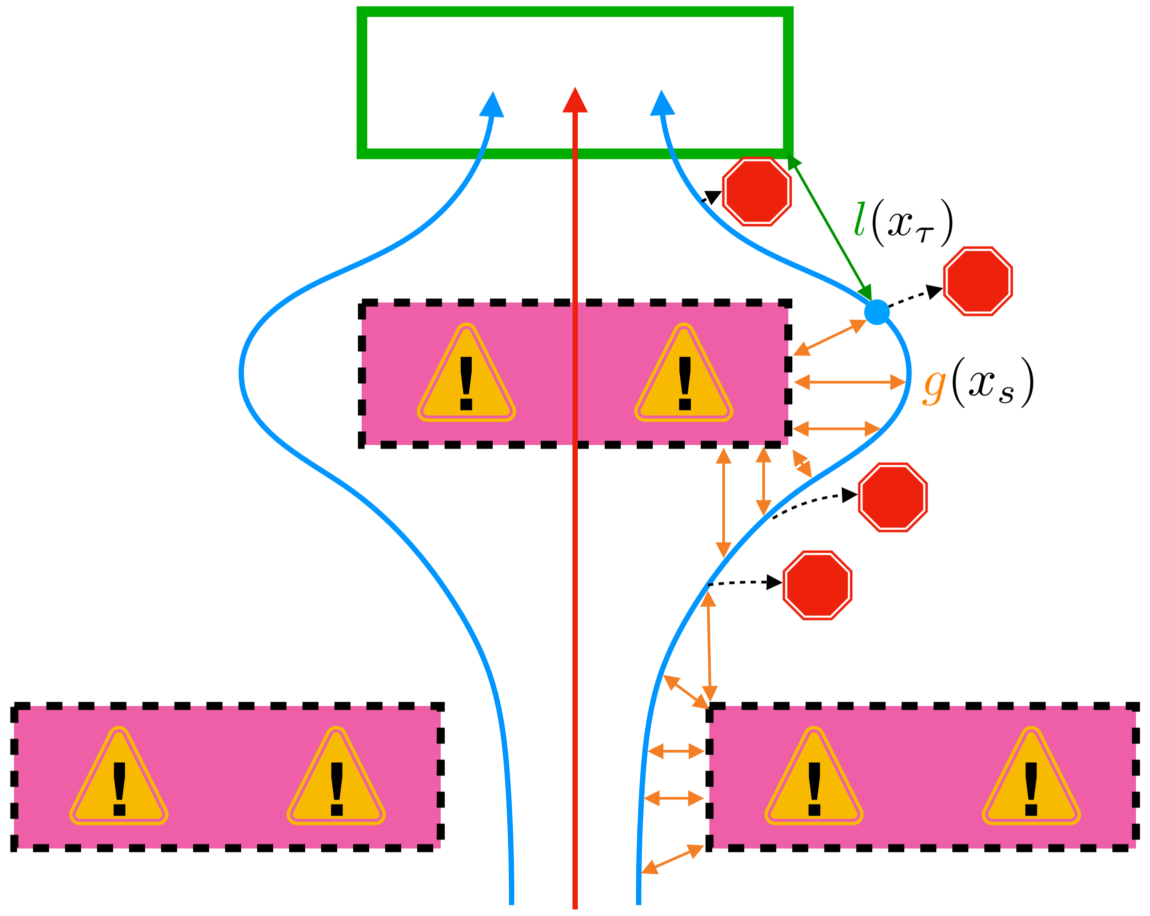
How do we certify safety with finite-horizon simulation rollouts?

Learn policies to safely reach a known controlled-invariant set.

Reach-Avoid Reinforcement Learning

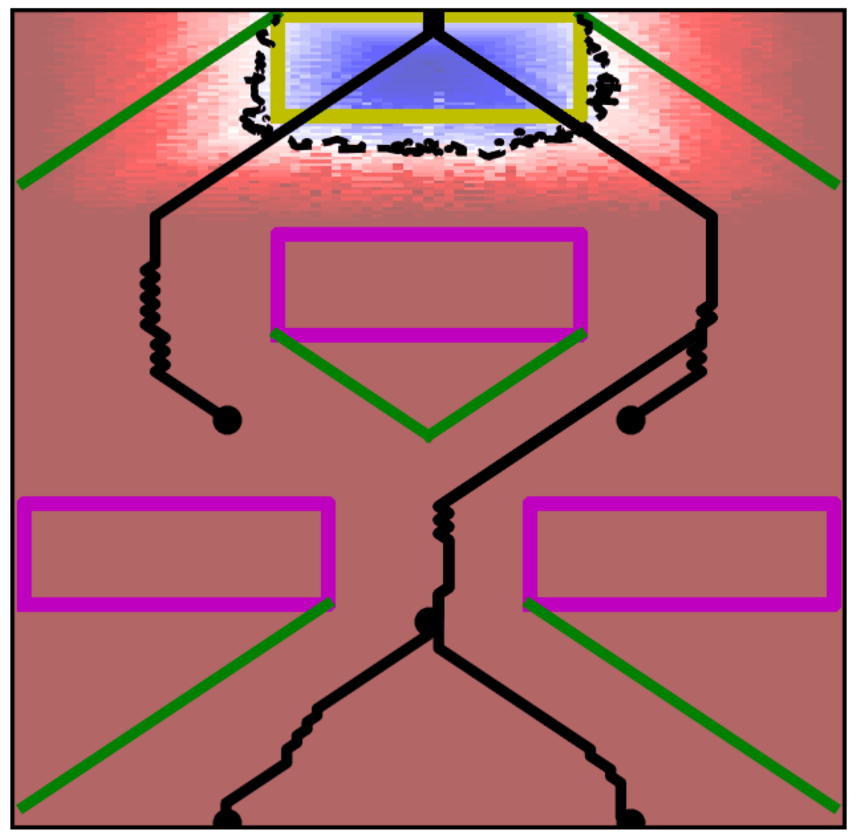
Time-discounted reach-avoid Bellman equation

$$V(s) = \gamma \max\left\{ \min\left\{ l(x), \min_{u \in \mathcal{U}} V(x') \right\}, g(x) \right\} + (1-\gamma) \max\left\{ l(x), g(x) \right\} \qquad \gamma \in [0,1)$$



Convergent under-approximation:

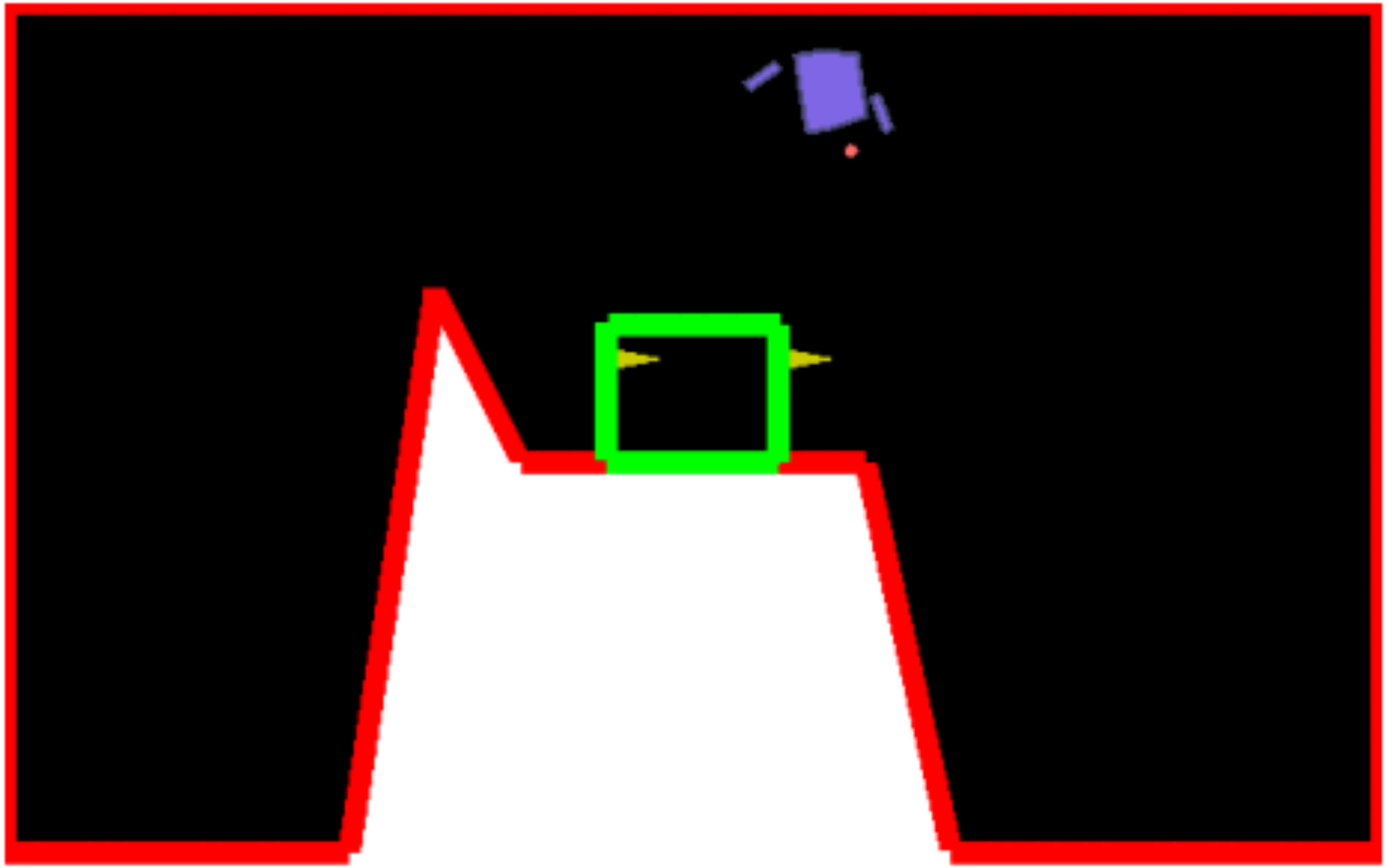
The sublevel set $\{x : V(x) < 0\}$ is a subset of $\overleftarrow{\mathcal{RA}}(\mathcal{T}, \mathcal{F})$ and converges to $\overleftarrow{\mathcal{RA}}(\mathcal{T}, \mathcal{F})$ as $\gamma \to 1$.



$\gamma = 0.900000$

[Hsu, Rubies-Royo et al. Safety and Liveness Guarantees through Reach-Avoid Reinforcement Learning. RSS, 2021]

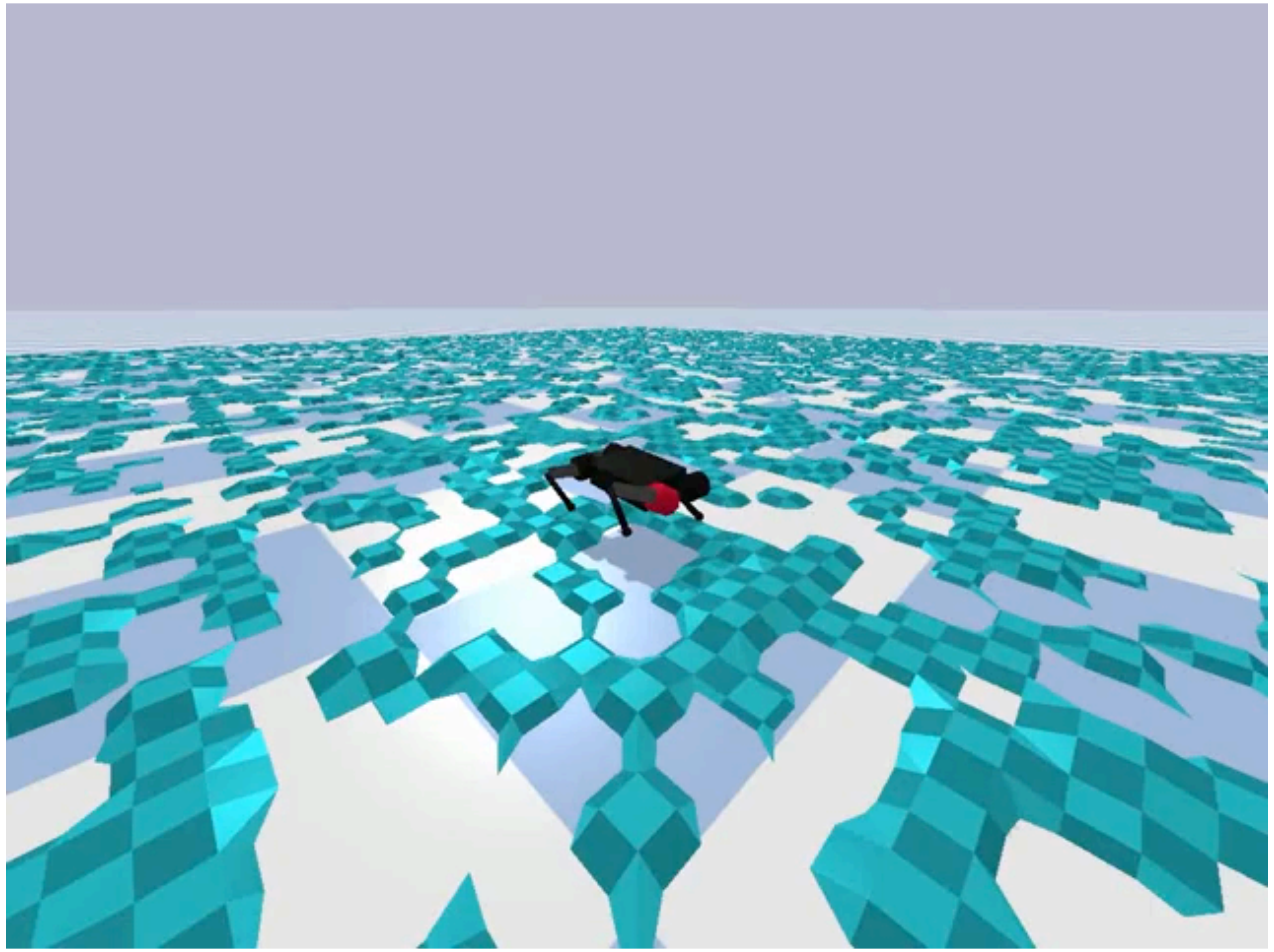
Reach-Avoid Reinforcement Learning



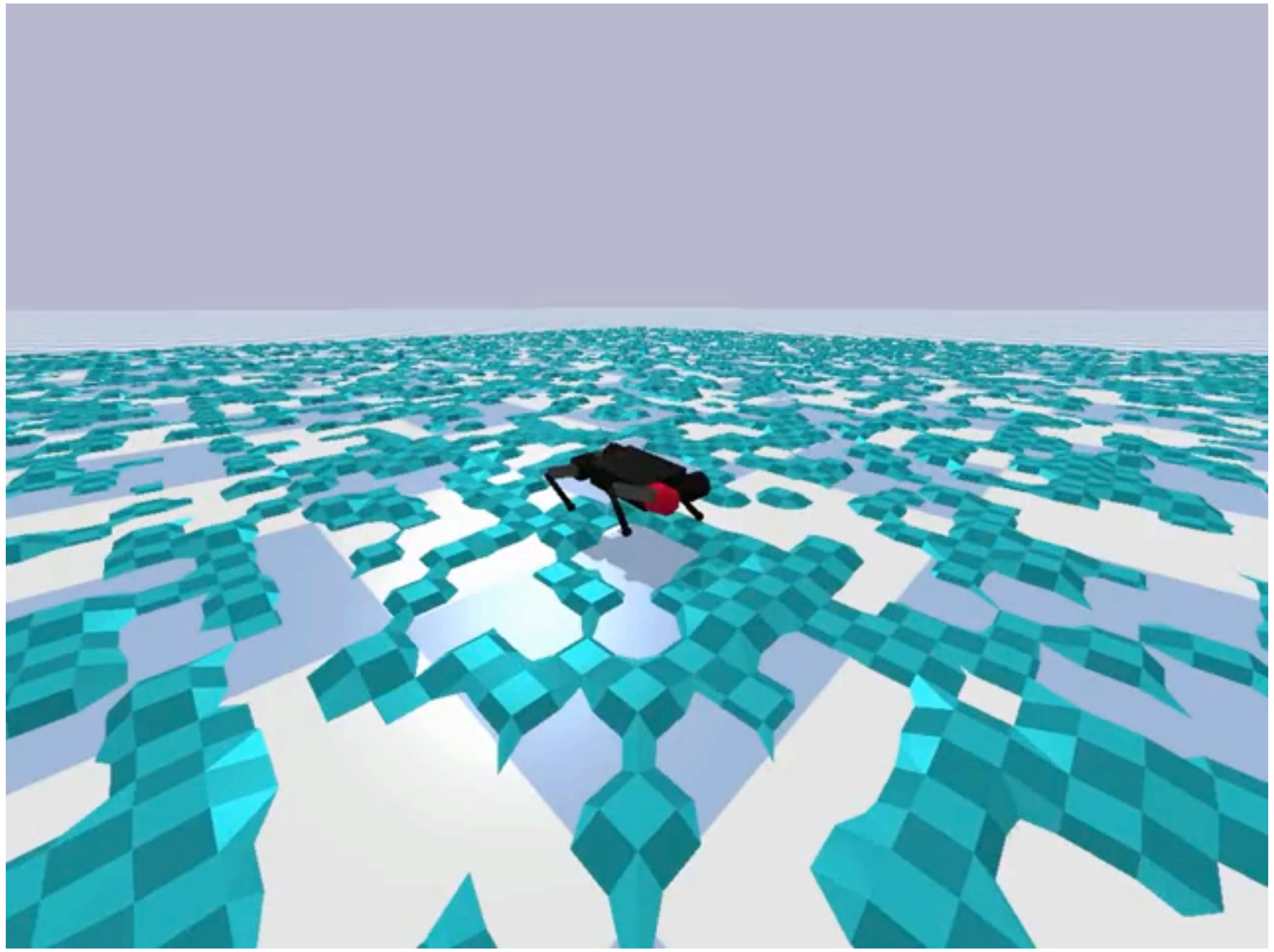
Simulation Stress Test

Gait control task

Previously unseen slippery terrain + randomly applied perturbation forces



Baseline policy



Safety policy

Generalization Guarantees

PAC-Bayes theory: probabilistic bounds on performance and safety *as long as* deployment environments have the same distribution as training environments.

Theorem 2 (PAC-Bayes Bound for Control Policies)**.** *For any $\delta \in (0,1)$, with probability at least $1-\delta$ over sampled environments $S \sim \mathcal{D}^N$, the following inequality holds:*

$$\underbrace{C_{\mathcal{D}}(P)}_{\textit{True expected cost}} \leq C_{PAC}(P) := \underbrace{C_S(P)}_{\textit{Training cost}} + \underbrace{\sqrt{\frac{\mathbb{D}(P\|P_0) + \log(\frac{2\sqrt{N}}{\delta})}{2N}}}_{\textit{“Regularizer”}}. \tag{14}$$

Algorithm 1 PAC-Bayes Policy Learning

Fix prior distribution $P_0 \in \mathcal{P}$ over policies
Inputs: $S = \{E_1, \dots, E_N\}$: Training environments, δ: Probability threshold
Outputs:
$P^\star_{\text{PAC}} = \underset{P \in P}{\text{argmin}}\ C_{\text{PAC}}(P) := \frac{1}{N}\sum_{E \in S} \underset{w \sim P}{\mathbb{E}}[C(r_w; E)] + \sqrt{\frac{\mathbb{D}(P\|P_0)+\log(\frac{2\sqrt{N}}{\delta})}{2N}}$
$C^\star_{\text{bound}} := \mathbb{D}^{-1}\left(C_S(P^\star_{\text{PAC}}) \| \frac{\mathbb{D}(P^\star_{\text{PAC}}\|P_0)+\log(\frac{2\sqrt{N}}{\delta})}{N}\right)$

[Majumdar et al. PAC-Bayes Control: Learning Policies that Provably Generalize to Novel Environments. CoRL, 2018]

Sim-to-Lab-to-Real

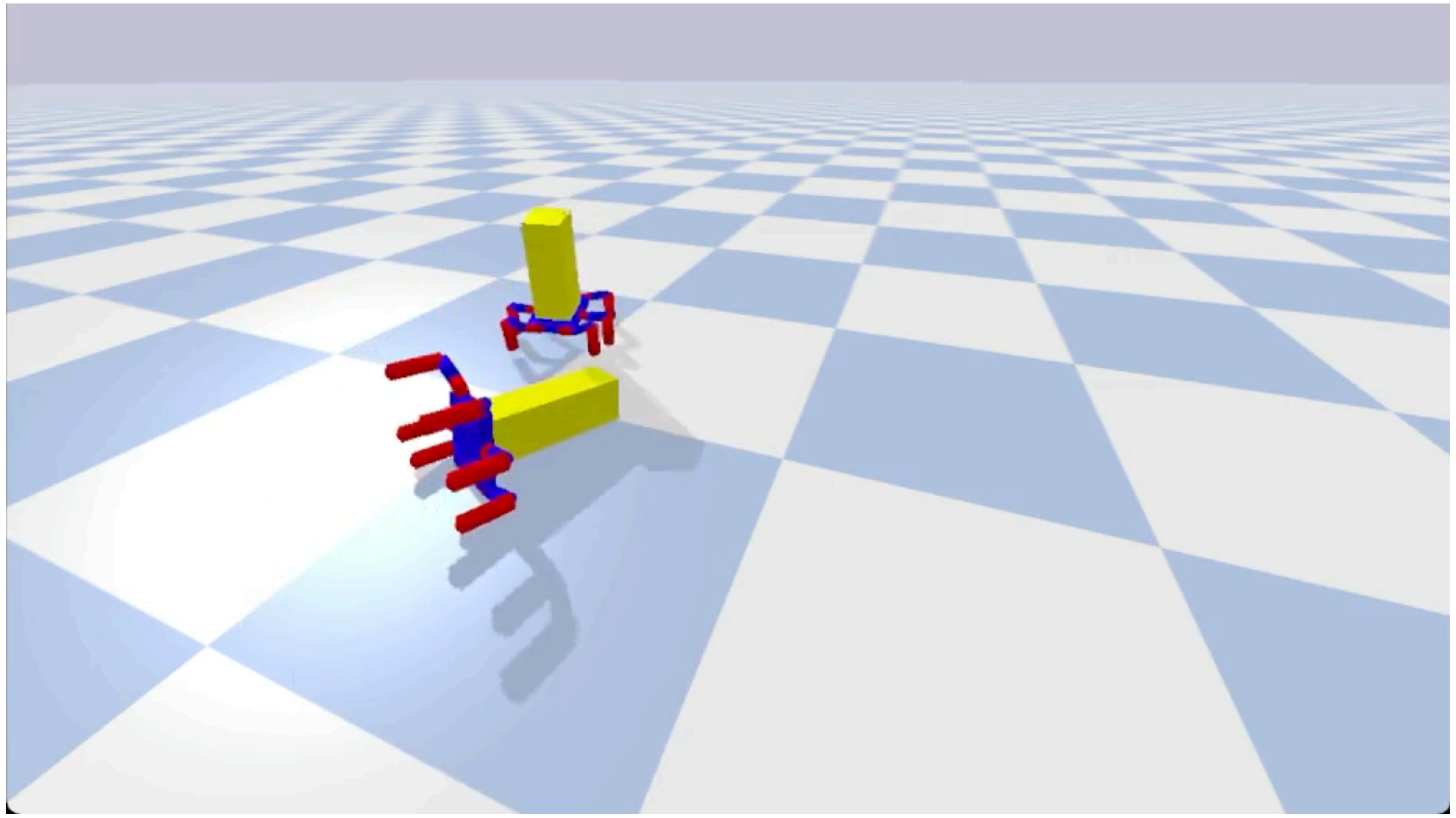
Fine-tune control policies in controlled environments before deployment.

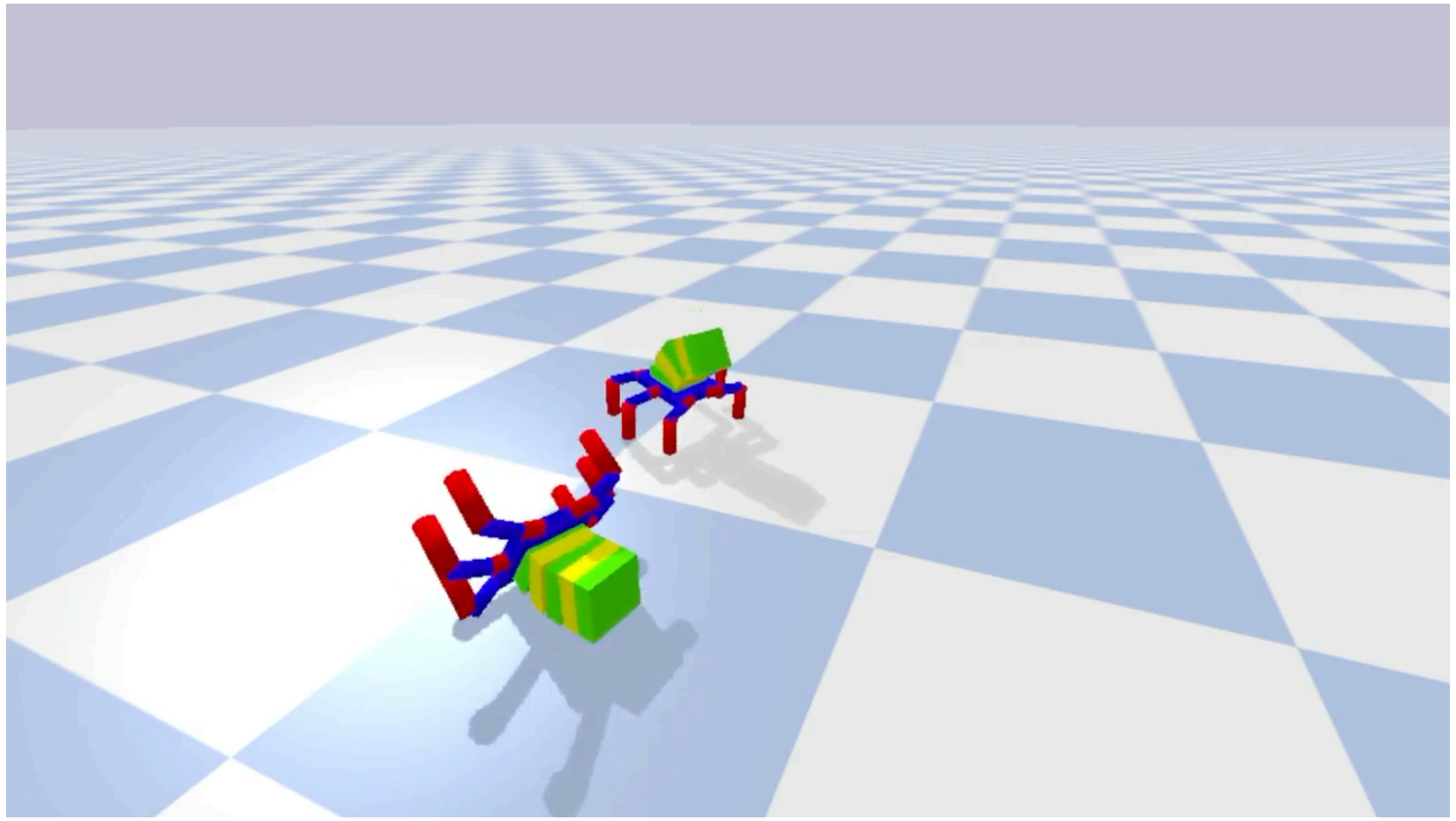
Co-training of performance and safety policies leads to stronger PAC-Bayes guarantees.

[Hsu, Ren, et al. Safe Reinforcement Learning with Shielding and Generalization Guarantees. *Artificial Intelligence*, 2022]

Robustness beyond Training Conditions

Soft Actor-Critic trained with random disturbance (domain randomization) and no payload





Adversarial Safety Reinforcement Learning?


amazon
PrimeAir

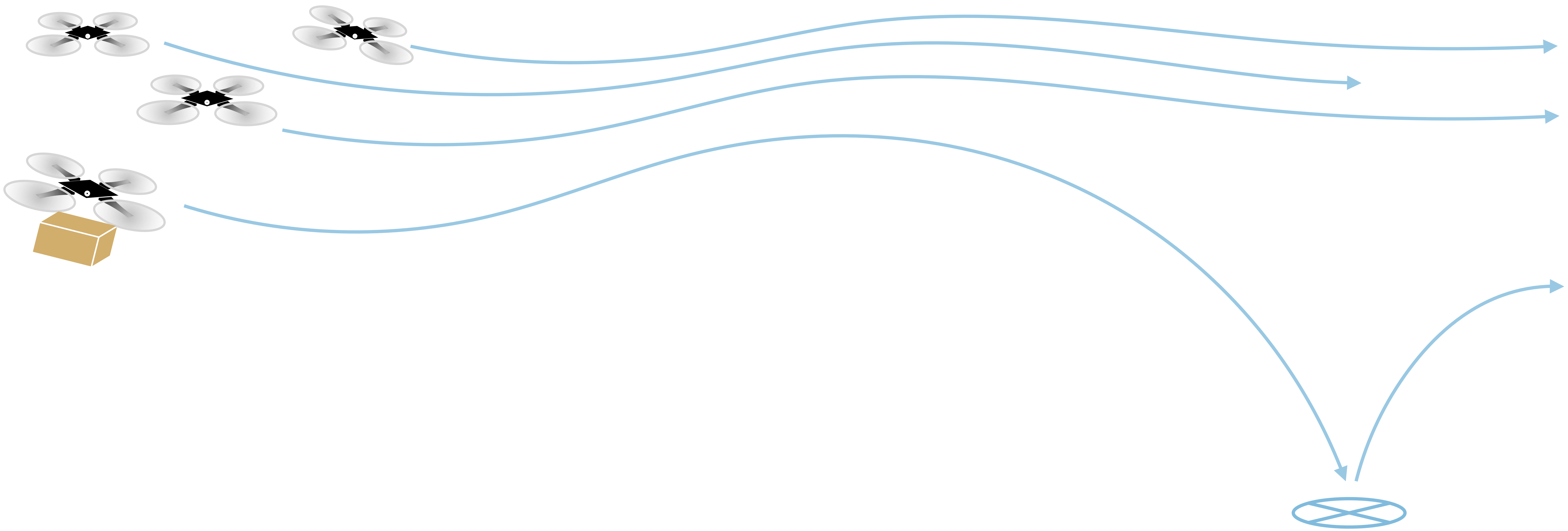
Some Takeaways

Safe Learning is critical for many real-world RL applications (can't learn if you break).

Safety Filters allow RL systems to explore and learn within a ***safety envelope***.

Learning-based Safety Analysis improves **scalability**, **flexibility** and **robustness** of Safe RL.

MIND THE GAP



jfisac@princeton.edu