*AA 598B Special Topics*

# Decision-Making & Control for Safe Interactive Autonomy

Instructor: Prof. Karen Leung

*Autumn 2024*

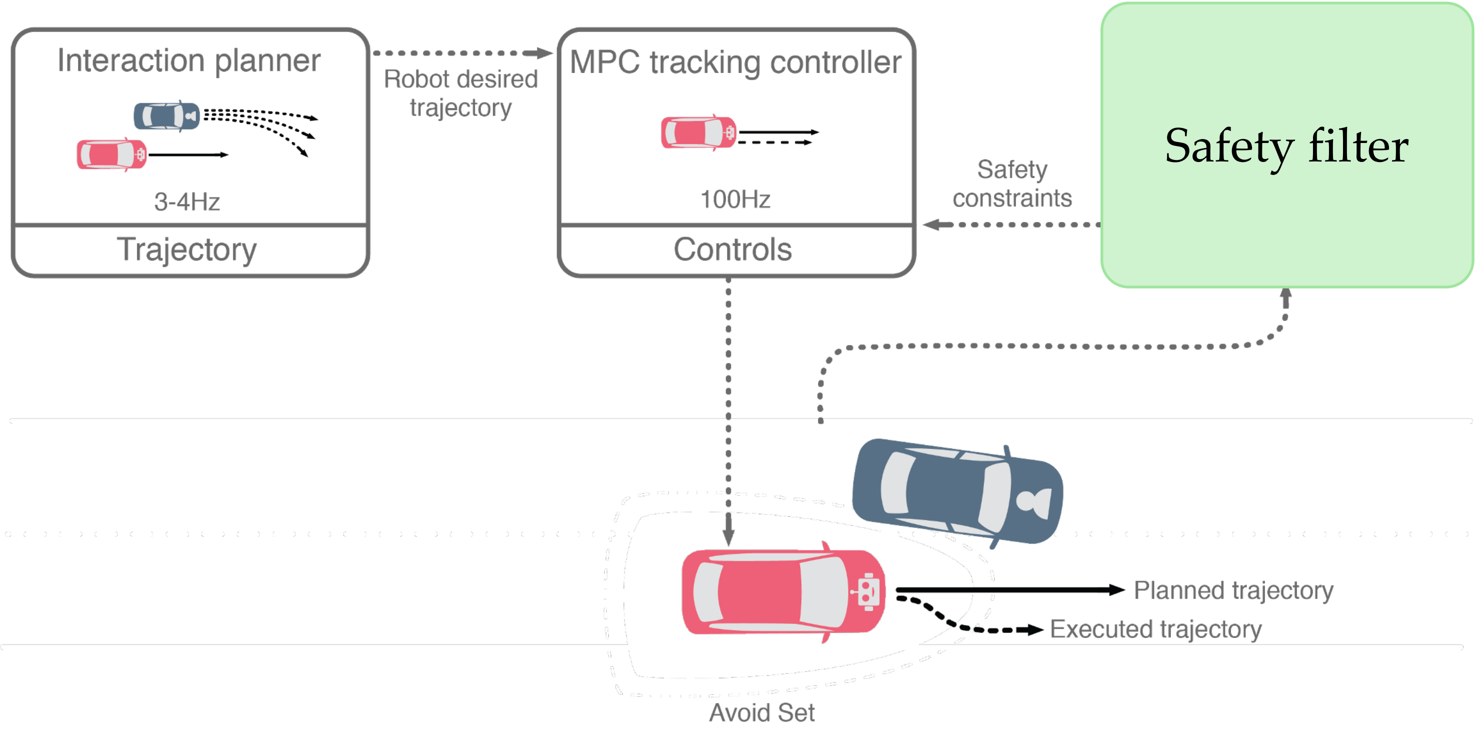
https://faculty.washington.edu/kymleung/aa598/

# Announcements

- No lecture Wednesday
- Sign up for a presentation time slot!
  - See course website
- Guest lecture record uploaded
- OH after class

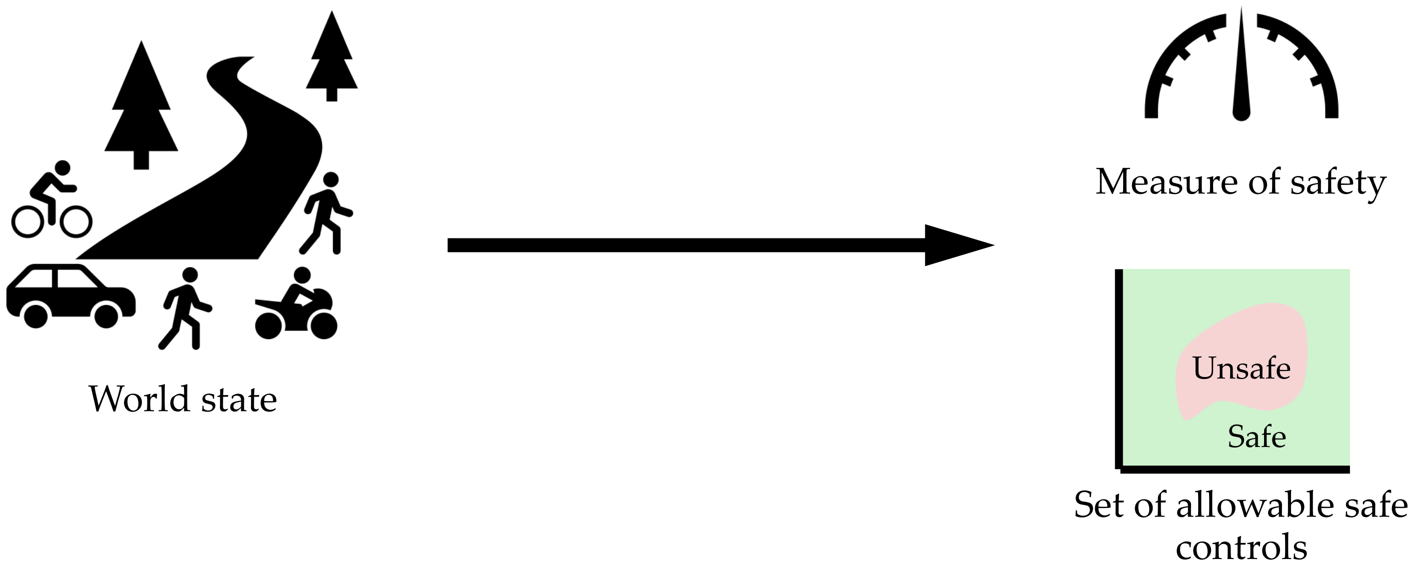
# Last time

- Introduce the concept of a "safety filter"



# Last time

- Introduce the concept of a "safety filter"
- Introduce the idea of a "safety concept"



# Last time

- Introduce the concept of a "safety filter"
- Introduce the idea of a "safety concept"
- Introduce HJ reachability as a way to define safe/unsafe sets
  - Frame as an optimal control problem.

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

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

# Today

- Continue HJ reachability discussion
- Control barrier functions
- Data-driven methods

# Ingredients of a HJ reachability problem

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

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

- Dynamics
- Control bounds
- Disturbance bounds
- Initial value function, aka, collision set
- State domain, aka grid size and limits
- Reach/avoid setup

# Can apply to *relative dynamics*

**Hamilton-Jacobi-Isaacs partial differential equation**

$$\frac{\partial V(x,t)}{\partial t} + \min\left\{0, \max_{u\in U} \min_{d\in D} \frac{\partial V(x,t)}{\partial x} \cdot f(x,u,d)\right\} = 0$$

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

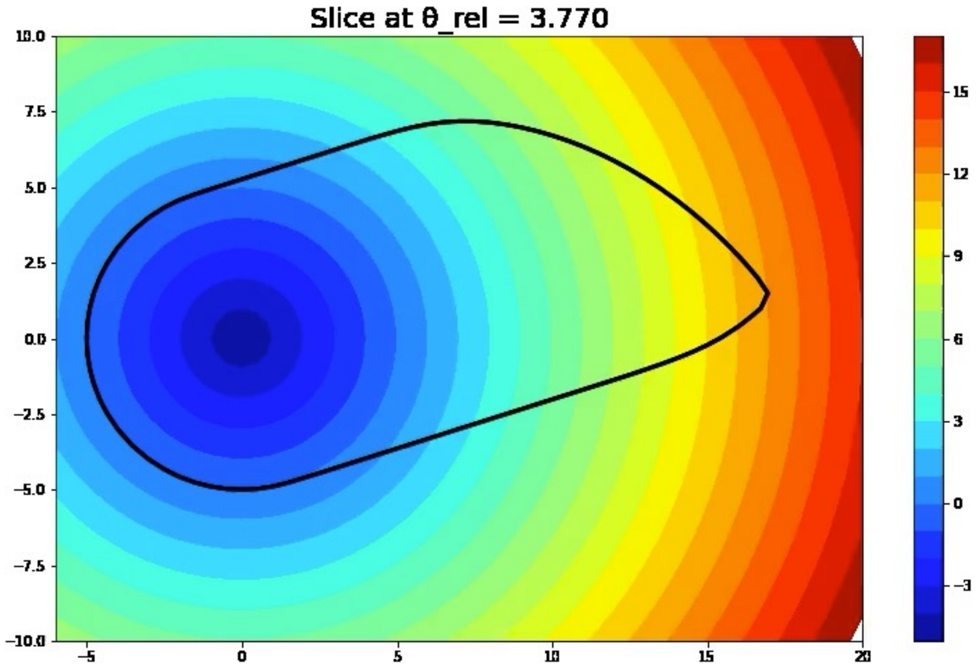
Measure of safety

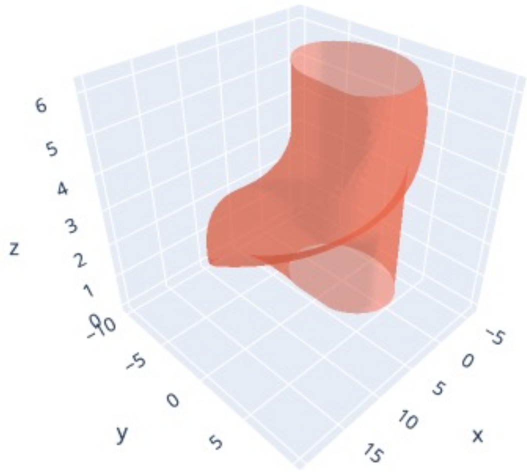
Robot avoids collision

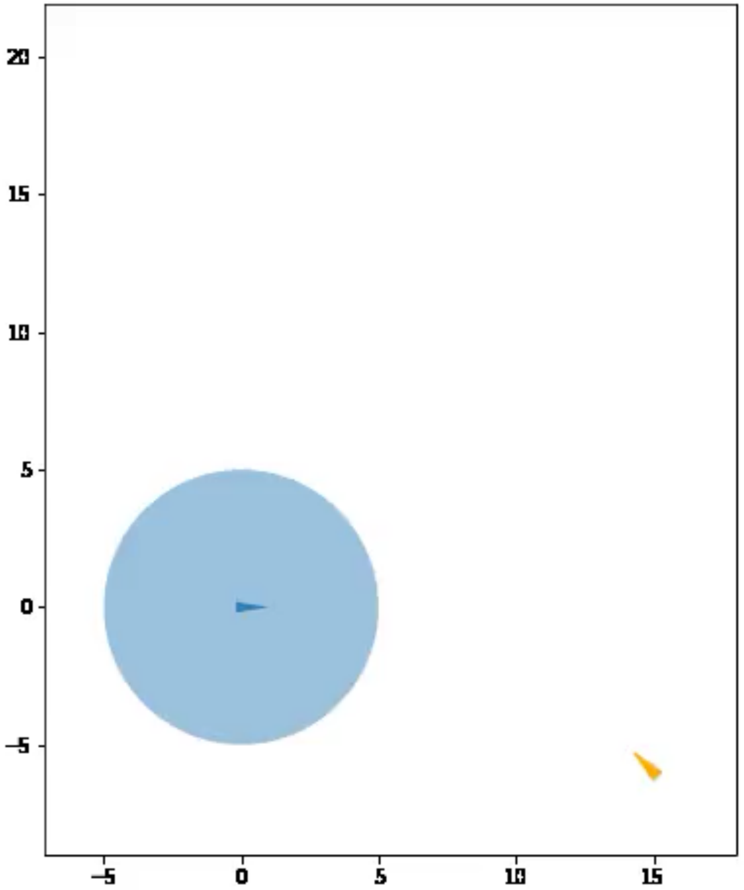
Human seeks collision

# Aircraft collision avoidance

$$\begin{bmatrix} \dot{x}_{\mathrm{rel}} \\ \dot{y}_{\mathrm{rel}} \\ \dot{\theta}_{\mathrm{rel}} \end{bmatrix} = \begin{bmatrix} -v_{\mathrm{a}} + v_{\mathrm{b}} \cos\theta_{\mathrm{rel}} + y_{\mathrm{rel}} u_{\mathrm{a}} \\ v_{\mathrm{b}} \sin\theta_{\mathrm{rel}} - x_{\mathrm{rel}} u_{\mathrm{a}} \\ u_{\mathrm{b}} - u_{\mathrm{a}} \end{bmatrix}$$



# We can construct different types of safety concepts

- Constant velocity assumption (VO)

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

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

- Hard braking assumption

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

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

- Forward reachable sets
  - Where can I definitely reach even under worst-case disturbances

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

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

# We can get creative

- **State dependent control sets**
  - Also coupled constraints, but harder
- More **“interesting” initial value**
  - Velocity-aware to encode collision severity
  - Region-specific collisions
  - Learned value function
- Change **agent assumptions**. Reach-reach, Reach-avoid, avoid-avoid
- **State augmentation** to include parameters
  - Parameterized formulation
- **Reaction-time**
  - Can string together multiple reachability problems. Solution of one is the initial condition for other

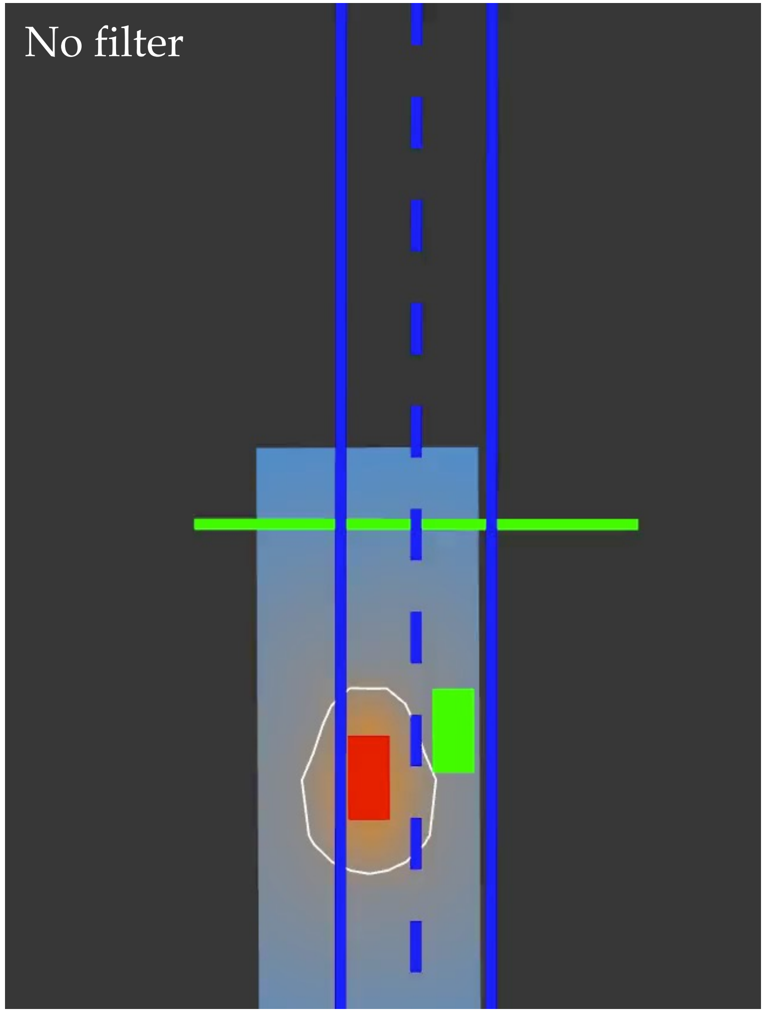
# Types of safety filter logic

Least restrictive

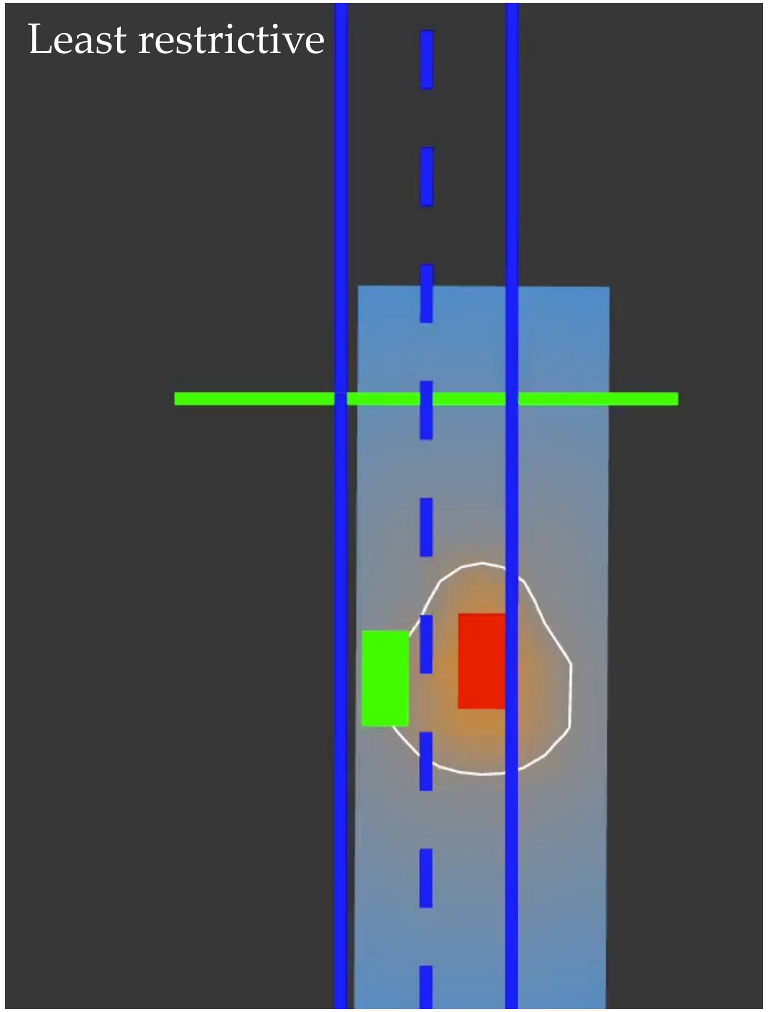
# Types of safety filter logic

## Minimally interventional

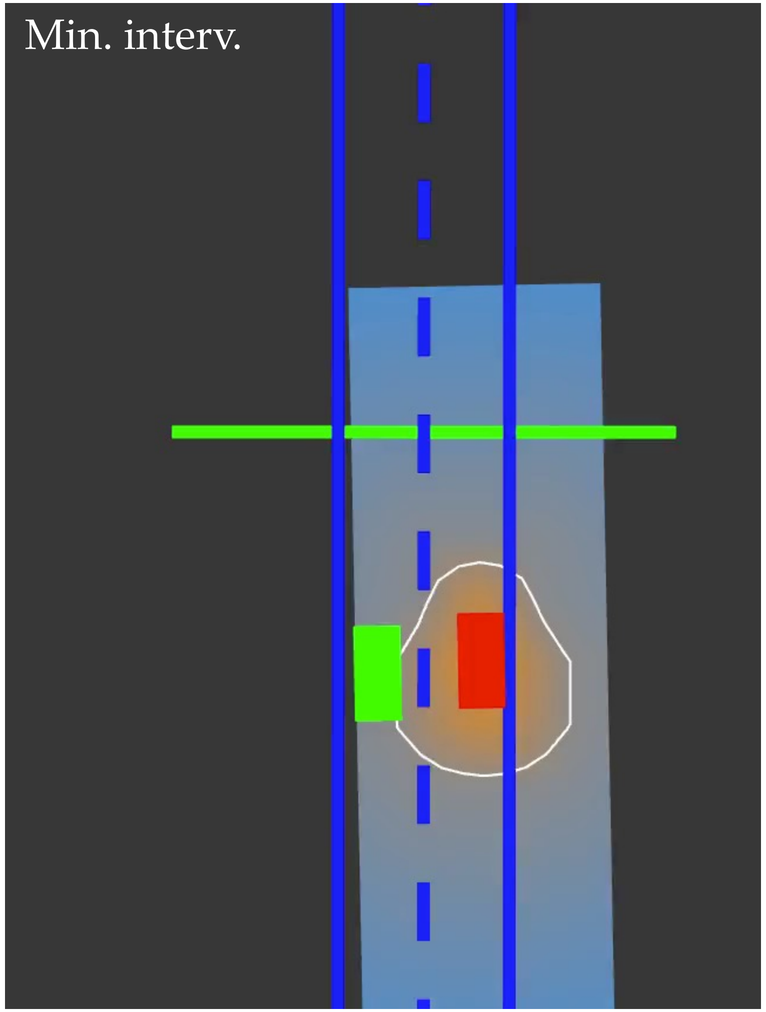
# Using HJ BRTs as safety filters




https://youtube.com/shorts/ydhvZ29HHBo?feature=share




https://youtube.com/shorts/5uyLyYURqwM?feature=share

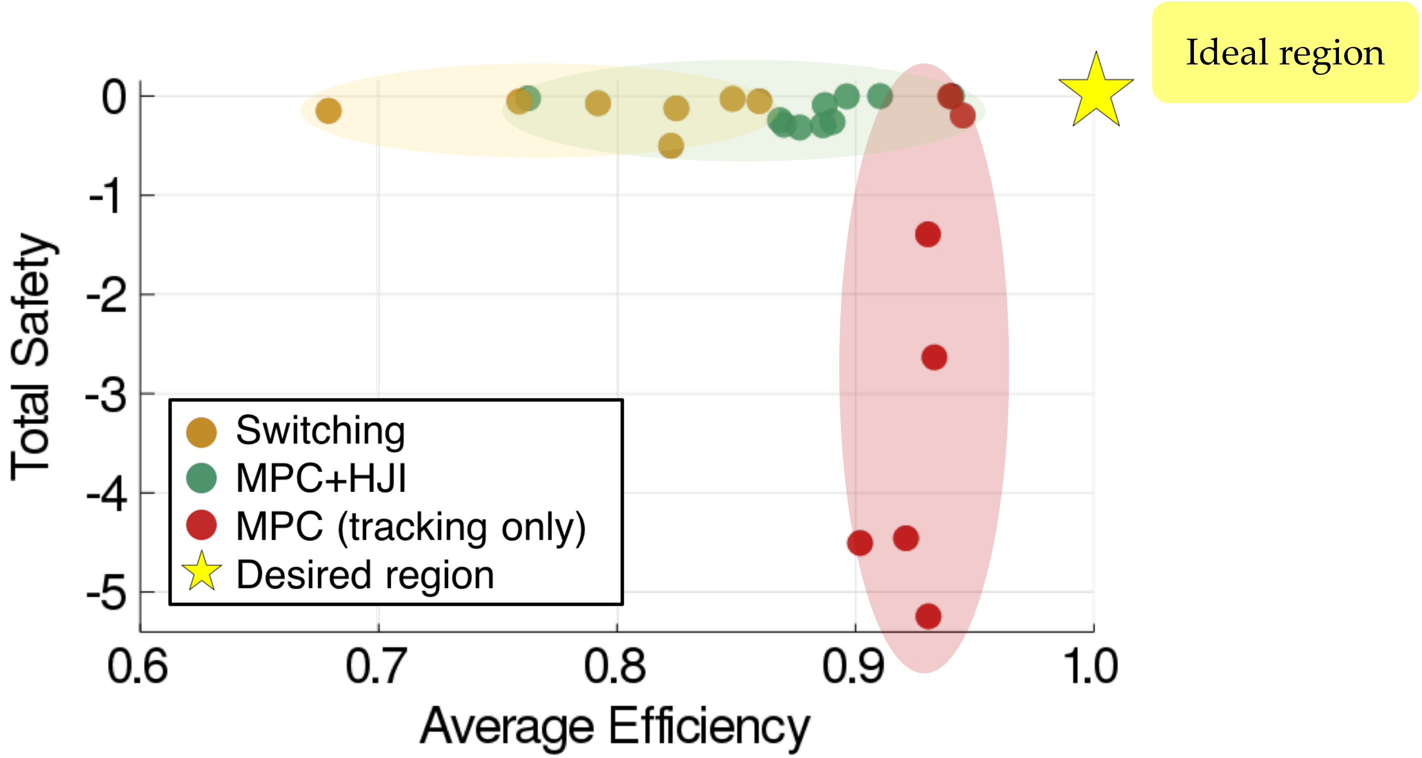



https://youtube.com/shorts/3Mq6-QPljbg?feature=share

# Safety – performance tradeoff

$$\int_t V(z(t),t)\mathbf{1}[V(z(t),t) < 0]dt$$

Integral when V<0



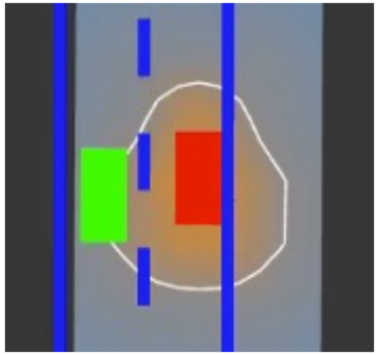

1 - g-force

# What are "reasonable" assumptions about how other agents behave?

**Hamilton-Jacobi-Isaacs partial differential equation**

$$\frac{\partial V(x,t)}{\partial t} + \min\left\{0, \max_{u \in U} \min_{d \in D} \frac{\partial V(x,t)}{\partial x} \cdot f(x,u,d)\right\} = 0$$

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



Overly-conservative assumptions lead to impractical safety concepts!

## How to pick "reasonable" choices for $U$ and $D$?

Leung, K., Bajcsy, A., Schmerling, E., and Pavone, M., **Towards data-driven synthesis of autonomous vehicle safety concepts,** https://arxiv.org/abs/2107.14412, 2022

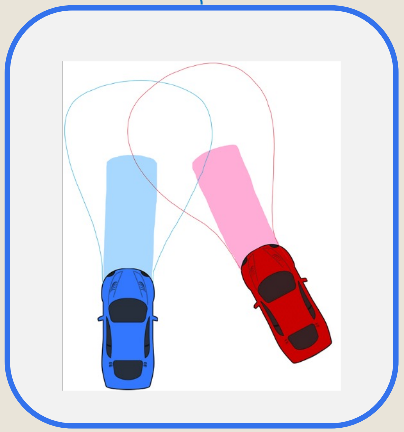
# Unification of Safety Concepts via Optimal Control Theory

Depends on the assumptions you make about other agents when evaluating safety

## *Hamilton-Jacobi Reachability*

**Open-loop**
**"non-reactive" policies**

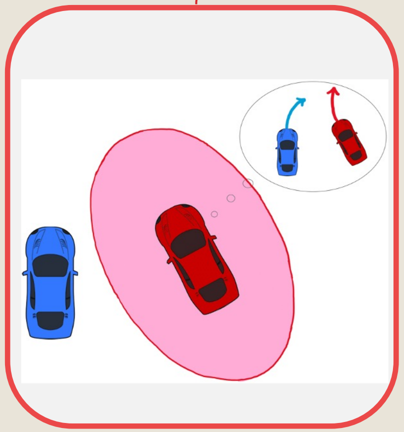
. . .



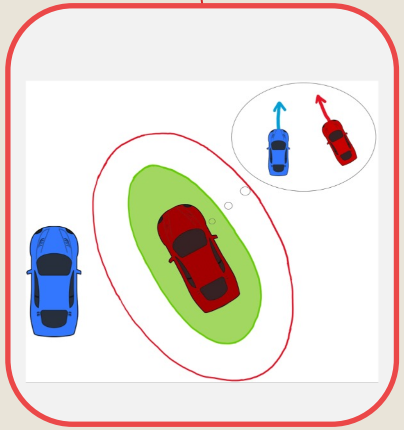
Consider *all* possible behaviors
*Full forward reachable set*

Consider only a *subset* of possible behaviors
*e.g., hard-braking **(SFF)***

**Closed-loop**
**"reactive" policies**



. . .

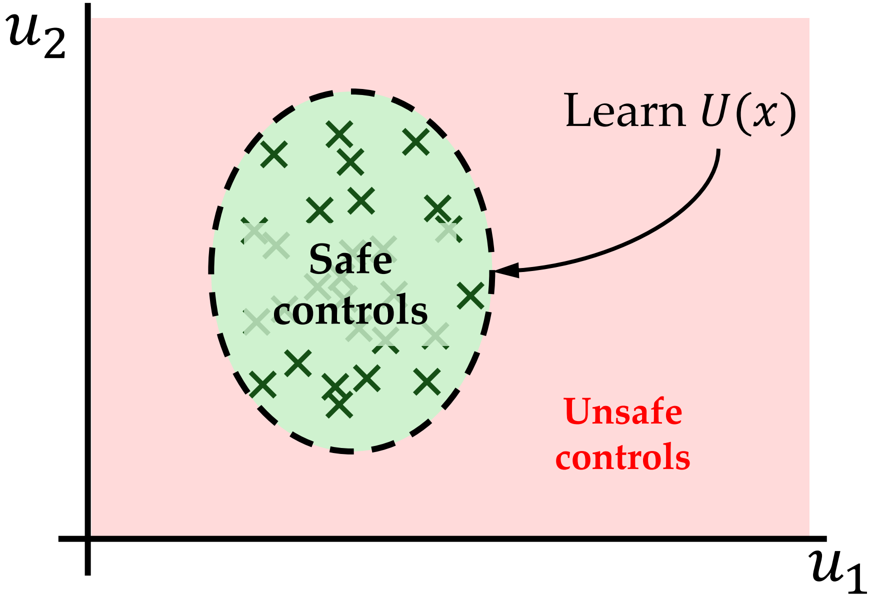


Guard against *all* possible policies
*Including worst-case outcomes*

Guard against a *subset* of possible policies
*Assumptions on other agent's behaviors*

# We can learn "reasonable controls" from human-human interaction data

Given a dataset of states and controls: $(x^{(1)}, u^{(1)}), (x^{(2)}, u^{(2)}), \ldots, (x^{(N)}, u^{(N)})$ we want to learn $U(x)$




**Key insight**: Humans take controls that keep them safe. Taking controls outside the boundary will lead to an undesirable outcome.

Data lives inside a control invariant set.

**Need to learn a control invariant set!**

Leung, K., Veer, S., Schmerling, E., and Pavone, M.,
**Learning Autonomous Vehicle Safety Concepts from Demonstrations**
*American Control Conference*, 2023, https://arxiv.org/abs/2210.02761

# Control barrier functions describe control invariant sets

**Control invariant set**

# Control barrier functions describe control invariant sets

**Control barrier function**

Choi et al 2021 https://arxiv.org/abs/2104.02808

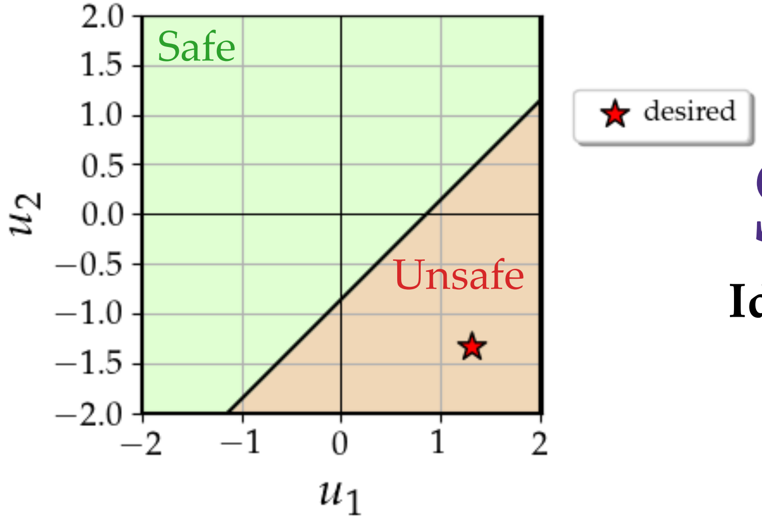
# Control barrier functions describe control invariant sets

**Control barrier function**

Choi et al 2021 https://arxiv.org/abs/2104.02808

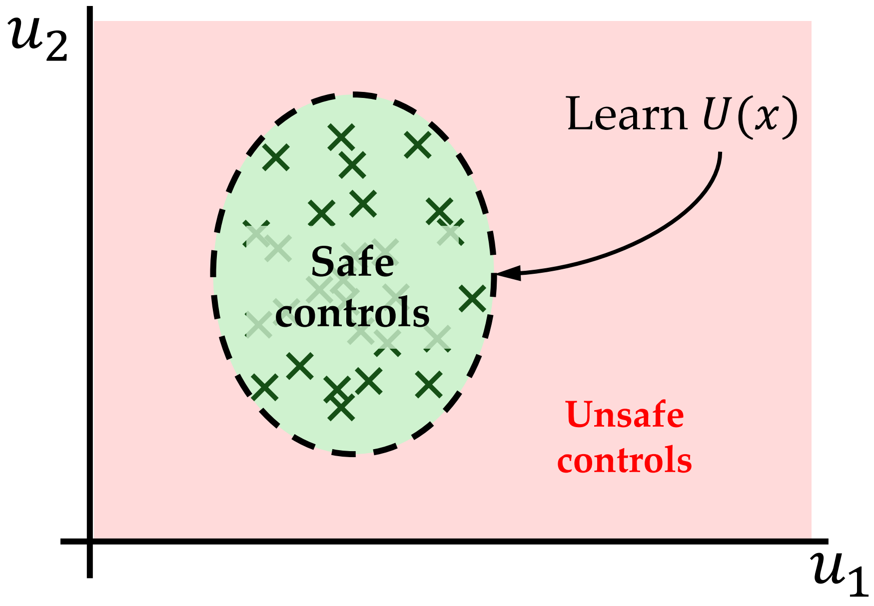
# Safety filter as a QP

**Idea: Project control into feasible safe control set**



# Learning CBFs from demonstrations

Given a dataset of states and controls: $(x^{(1)}, u^{(1)}), (x^{(2)}, u^{(2)}), \ldots, (x^{(N)}, u^{(N)})$ we want to learn $U(x)$




- Use data to learn parameters of CBF!

Robey et al 2020 https://arxiv.org/abs/2004.03315

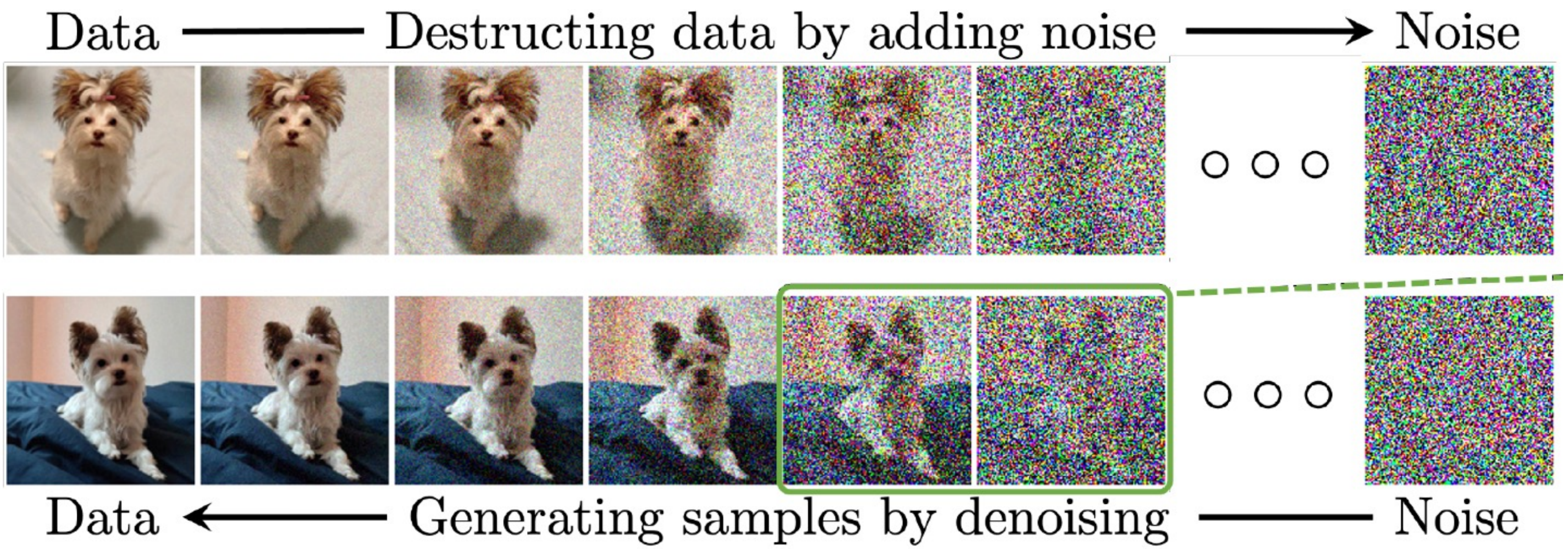
# So…is a safety filter all we need for safe autonomy?

- Safety needs to be considered in *all* aspects of the autonomy stack
  - Perception: not all perception errors are equal!
  - Prediction: not all prediction errors are equal!
  - Planning: avoid situations that require safety safety filter to activate!
    - Reason about potential risks
    - Contingency planning
    - Graceful degradation
  - Control: need a careful balance between safety and practicality
    - Cannot be overly conservative but still encompass reasonable assumptions of others

# CBFs in generative models

- A safety & dynamics-aware cost function for guided diffusion models

$$p_\theta(u^{i-1}|u^i) \approx N(u^{i-1}; \mu_\theta(u^i, i), \Sigma^i)$$ **Neural Network**




Mizuta, K. and Leung, K., *CoBL-Diffusion: Diffusion-Based Conditional Robot Planning in Dynamic Environments Using Control Barrier and Lyapunov Functions, In IEEE/RSJ International Conference on Intelligent Robots & Systems*, 2024 (https://arxiv.org/abs/2406.05309)

# CBFs in generative models

- A safety & dynamics-aware cost function for guided diffusion models

$$p_\theta(u^{i-1}|u^i) \approx N(u^{i-1}; \mu_\theta(u^i, i), \Sigma^i)$$ **Neural Network**

**Reverse Process w/ Guidance**

$$p_\theta(u^{i-1}|u^i) \approx N(u^{i-1}; \mu_\theta(u^i, i) + g, V\Sigma^i)$$

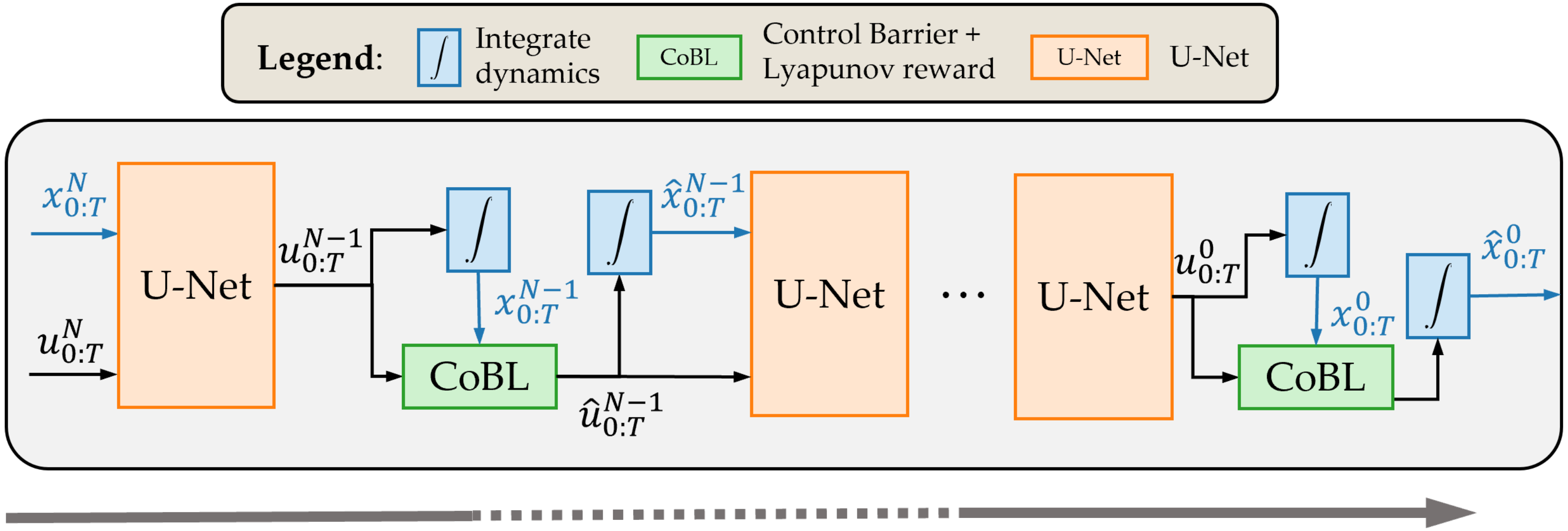
**CBF & CLF guidance**

$$g = \sum_{k=1}^{K} \sum_{t=0}^{T} \nabla_{u_t} \boldsymbol{W}_{\boldsymbol{k}}(\boldsymbol{x}_{\boldsymbol{t}}, \boldsymbol{u}_{\boldsymbol{t}})$$

- CBF Reward for **safety**
- CLF Reward for **goal-reaching**

Mizuta, K. and Leung, K., *CoBL-Diffusion: Diffusion-Based Conditional Robot Planning in Dynamic Environments Using Control Barrier and Lyapunov Functions, In IEEE/RSJ International Conference on Intelligent Robots & Systems*, 2024 (https://arxiv.org/abs/2406.05309)

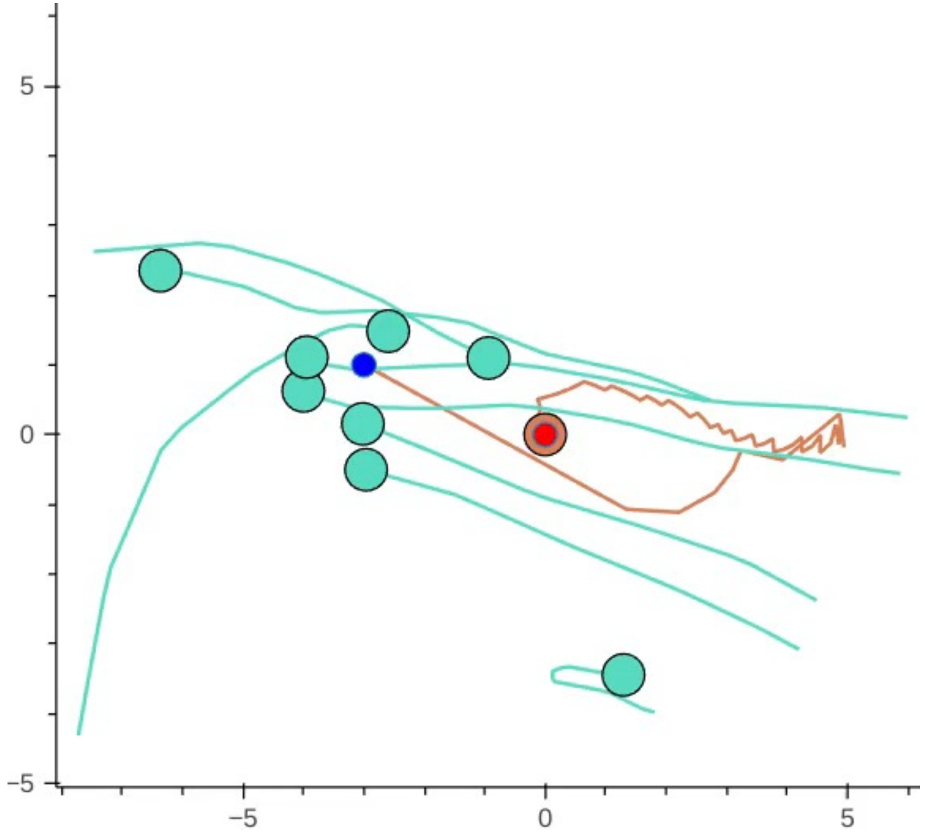
# **CoBL-Diffusion**: Guiding trajectory diffusion models with dynamics-aware safety

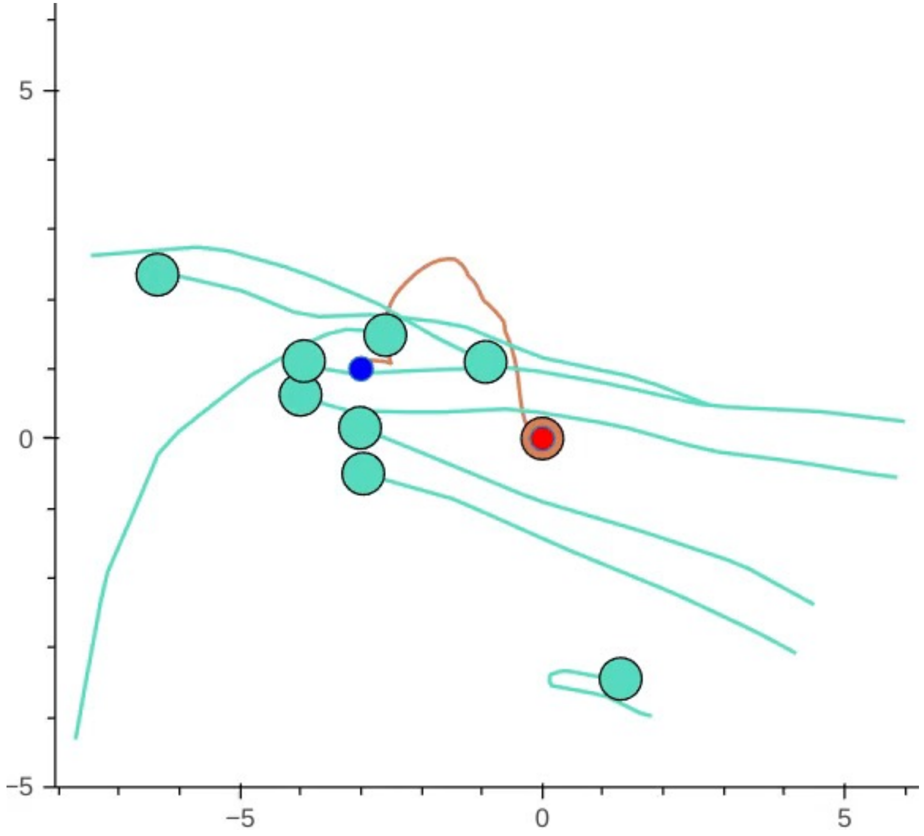



- Dynamically-feasible trajectory generation
- Use **<u>Co</u>ntrol <u>B</u>arrier & <u>L</u>yapunov Functions** as *guidance functions*

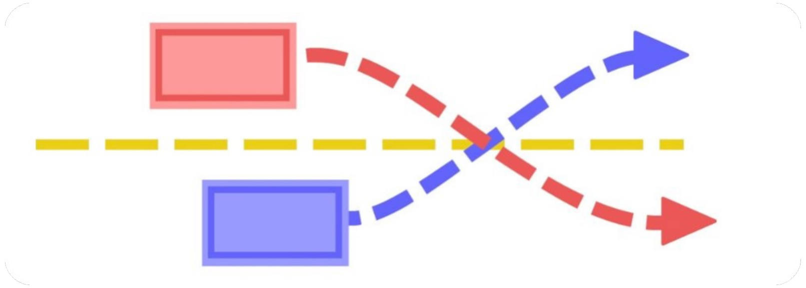
Mizuta, K. and Leung, K., *CoBL-Diffusion: Diffusion-Based Conditional Robot Planning in Dynamic Environments Using Control Barrier and Lyapunov Functions, In IEEE/RSJ International Conference on Intelligent Robots & Systems*, 2024 (https://arxiv.org/abs/2406.05309)

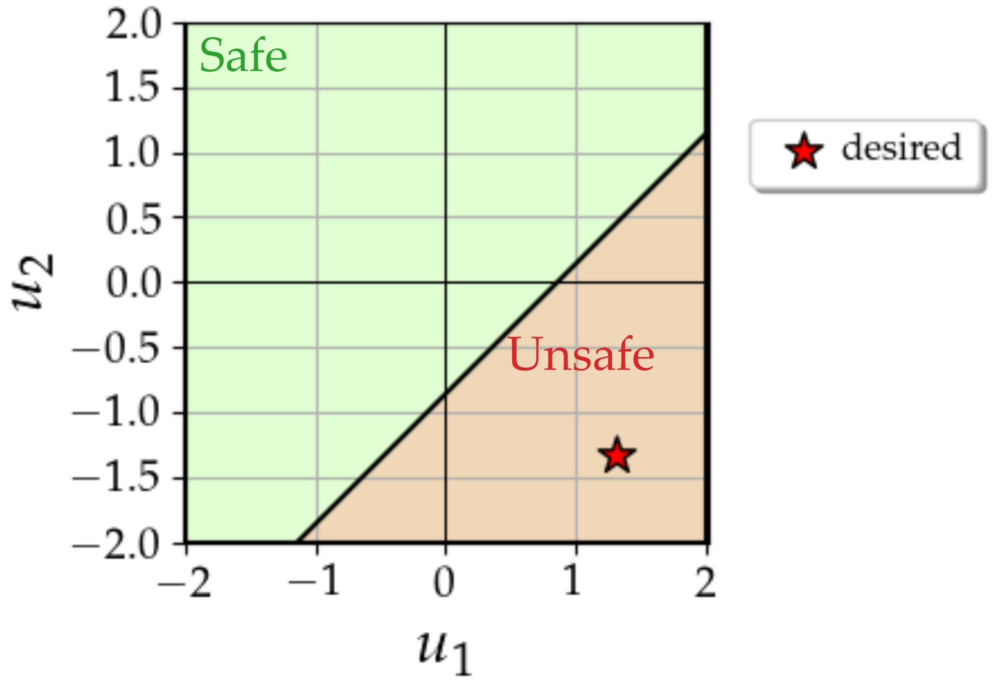
# Safety is accounted for *during* trajectory generation, rather than after-the-fact



# CBFs for explaining responsibility





# Recall, the CBF safety filter

**Single agent**

$$u^{\text{exec}} = \operatorname*{argmin}_{u \in \mathcal{U}} \|u - u^{\text{des}}\|_2^2 \text{ s.t. } \underbrace{\nabla b(x)^T f(x,u) + \alpha(b(x)) \geq 0.}_{\text{CBF safety constraint}}$$

**Multiple agents**

$$u_{1:N}^{\text{exec}} = \operatorname*{argmin}_{u_1,\ldots,u_N} \sum_{i=1}^{N} \|u_i - u_i^{\text{des}}\|_2^2$$

$$\text{s.t. } \nabla b(\mathbf{x})^T \left[ \tilde{f}(\mathbf{x}) + \sum_{i=1}^{N} g_i(\mathbf{x}) u_i \right] + \alpha(b(\mathbf{x})) \geq 0$$

$$u_1 \in \mathcal{U}_1, \ldots, u_N \in \mathcal{U}_N.$$

**Issues?**

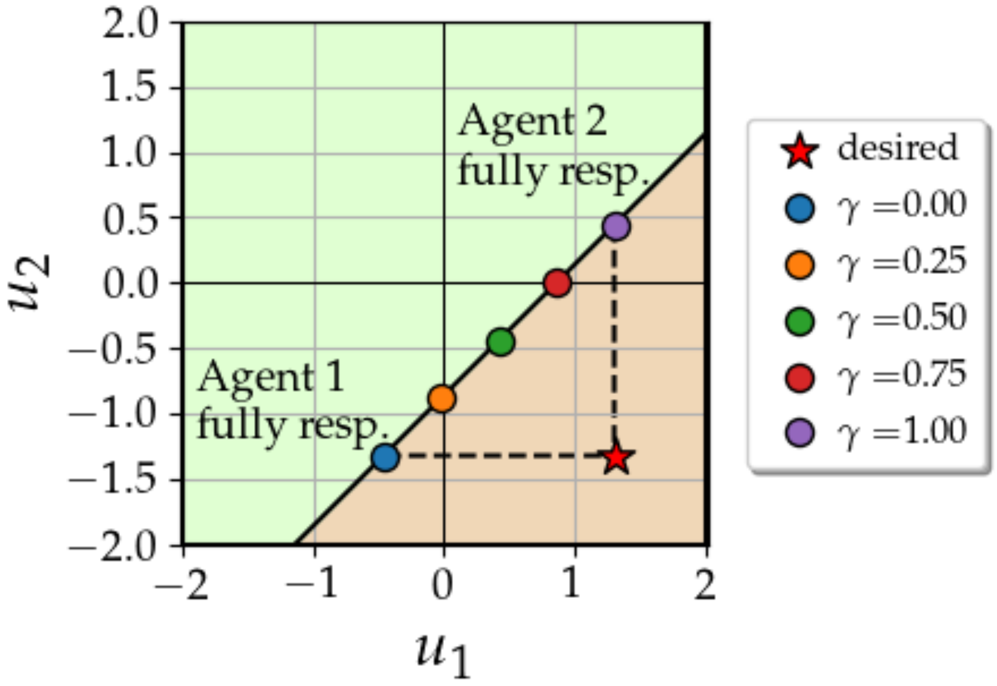
# A responsibility-aware safety filter

$$\underset{u_1,\ldots,u_N,\epsilon}{\text{argmin}} \sum_{i=1}^{N} \left(\gamma_i \|u_i - u_i^{\text{des}}\|_2^2 + \beta_1 \|u_i\|_2^2\right) + \beta_2 \epsilon^2$$

$$\text{s.t.} \quad \nabla b(\mathbf{x})^T \left[\tilde{f}(\mathbf{x}) + \sum_{i=1}^{N} g_i(\mathbf{x}) u_i\right] + \alpha(b(\mathbf{x})) \geq -\epsilon$$

$$u_1 \in \mathcal{U}_1, \ldots, u_N \in \mathcal{U}_N$$

$$\epsilon \geq 0.$$




$0 \leq \gamma_i \leq 1, \boldsymbol{\gamma}^T \mathbf{1} = 1$

- **Responsibility allocation**: each agent's incentive to deviate
- Regularization (if $\gamma_i = 0$)
- Slack variable

**How to select $\gamma_i$'s?**

Remy, I., Fridovich-Keil, D., and Leung, K.,
**Learning responsibility allocations for multi-agent interactions: A differentiable optimization approach with control barrier functions**

# Learn responsibility allocation from data

$$\min_{\boldsymbol{\gamma}} \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x}^i, u^i_{1:N}) \in \mathcal{D}} \Delta(u^i_{1:N}, \tilde{u}^i_{1:N}(\boldsymbol{\gamma}))$$

Outer problem

$$\begin{aligned}
\text{s.t.} \quad \tilde{u}^i_{1:N}(\boldsymbol{\gamma}) = \underset{u_1,\ldots,u_N,\epsilon}{\operatorname{argmin}} & \sum_{i=1}^{N} \left(\gamma_i \|u_i - u_i^{\text{des}}\|_2^2 + \beta_1 \|u_i\|_2^2\right) + \beta_2 \epsilon^2 \\
\text{s.t.} \quad & \nabla b(\mathbf{x})^T \left[\tilde{f}(\mathbf{x}) + \sum_{i=1}^{N} g_i(\mathbf{x}) u_i\right] + \alpha(b(\mathbf{x})) \geq -\epsilon \\
& u_1 \in \mathcal{U}_1, \ldots, u_N \in \mathcal{U}_N \\
& \epsilon \geq 0.
\end{aligned}$$

Inner problem

**Need to differentiate inner problem w.r.t. $\gamma$**

**Differentiable quadratic program**

https://github.com/kevin-tracy/qpax

# HJ reachability for feasible scenario generation

- Generating *realistic* yet *challenging* scenarios is a crucial piece in V&V of an autonomous system

Add a HJ value function term in their adversarial policy learning algorithm

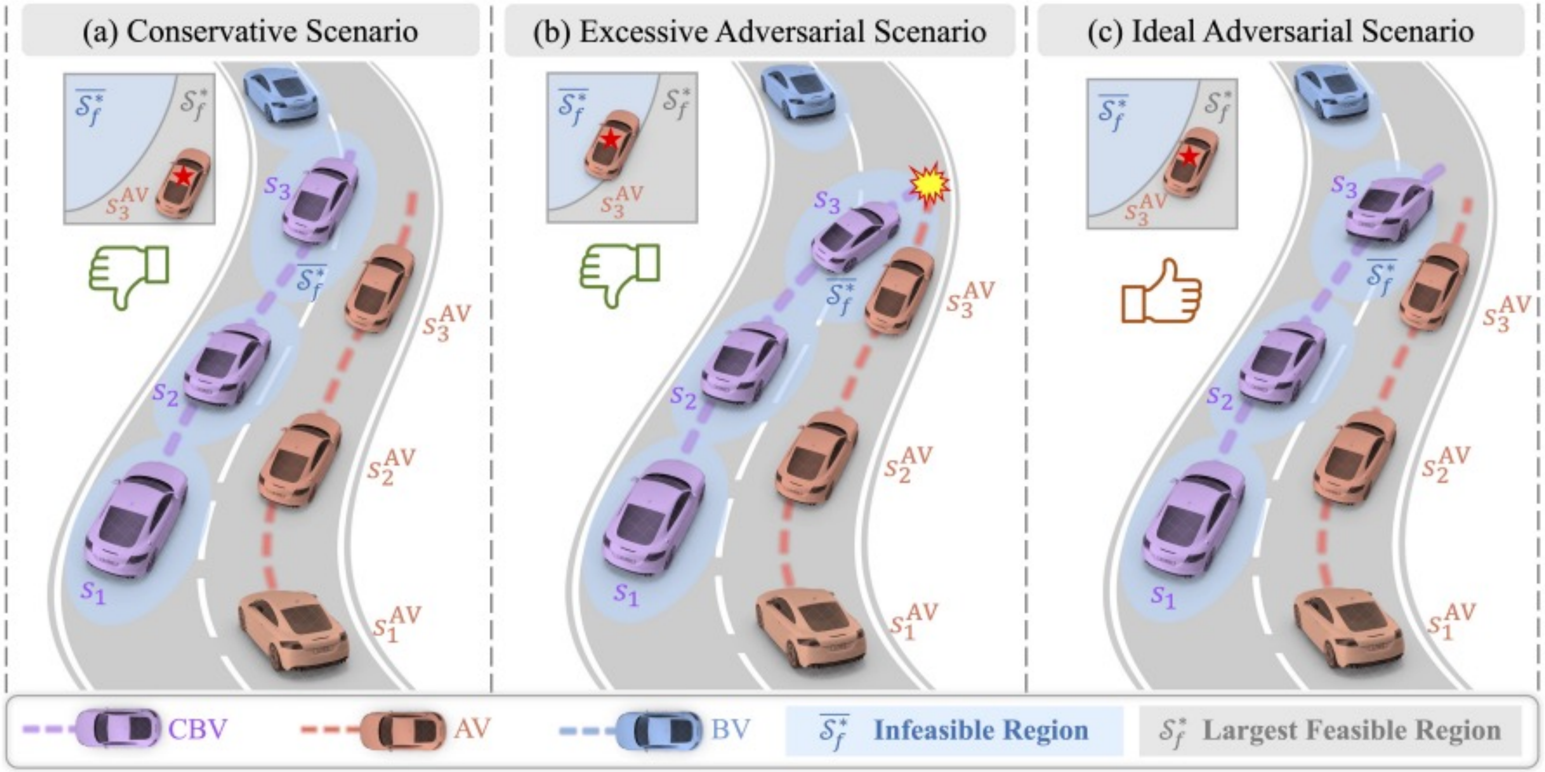



Figure 1: Illustration of adversarial yet AV-feasible scenarios in a two-lane traffic setting. The CBV employs three distinct policies, resulting in different scenarios: (a) Conservative scenario, where the policy is less adversarial; (b) Excessive adversarial scenario, resulting in an unavoidable collision; and (c) Ideal adversarial scenario, effectively balancing adversarial and AV's feasibility.

Chen at al 2024 https://arxiv.org/abs/2406.02983

# DeepReach

- Use a neural network to approximate the HJ value function

$$
\begin{aligned}
h(x_i, t_i; \theta) &= h_1(x_i, t_i; \theta) + \lambda h_2(x_i, t_i; \theta), \\
h_1(x_i, t_i; \theta) &= \|V_\theta(x_i, t_i) - l(x_i)\| \mathbb{1}(t_i = T), \\
h_2(x_i, t_i; \theta) &= \| \min \Big\{ D_t V_\theta(x_i, t_i) + H(x_i, t_i), \\
&\qquad l(x_i) - V_\theta(x_i, t_i) \Big\} \|.
\end{aligned}
$$

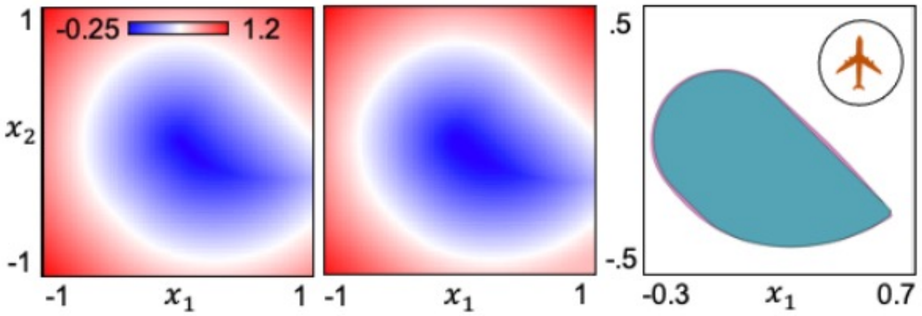



Fig. 1: The slices of the value functions and BRTs for $x_3 = \pi/2$. DeepReach recovers a value function (middle) that corresponds closely to the ground truth value function (left), computed using a principled HJI VI solver [7]. The two BRTs also align closely (right) – they overlap in the green region, the pink region is the error between the two BRTs.

Bansal and Tomlin 2020 https://arxiv.org/abs/2011.02082

# HJ reachability + Reinforcement learning

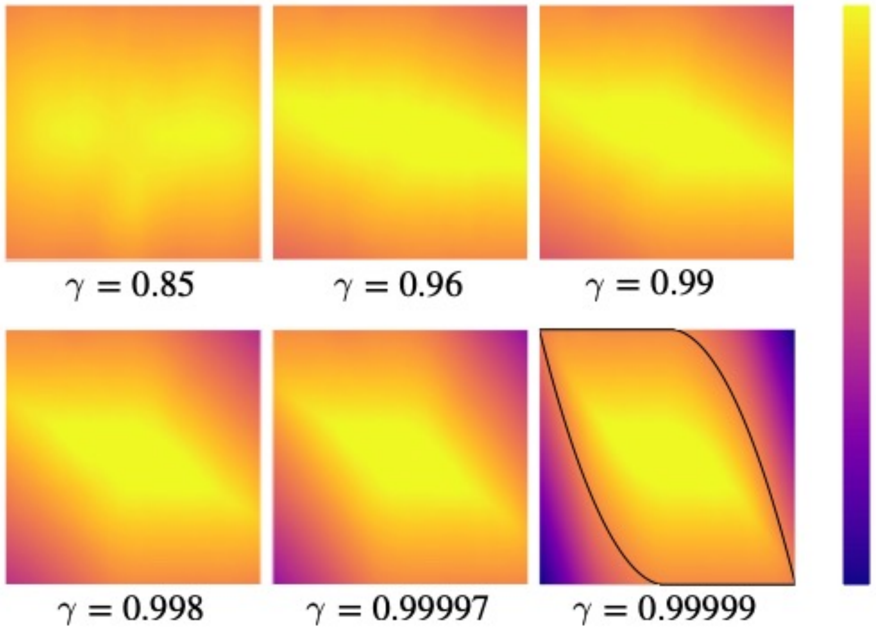



Fig. 1: Multiple snapshots of the neural network output of our safety Q-learning algorithm for a double-integrator system. As we anneal the discount factor $\gamma \to 1$ during Q-learning, our learned discounted safety value function asymptotically approaches the undiscounted value, allowing us to recover the safe set and optimal safety policy with very high accuracy.

Our central contribution is a modified form of the dynamic programming safety backup (5) which induces a contraction mapping in the space of value functions and is therefore amenable to reinforcement learning methods based on temporal difference learning [10, 24, 25].

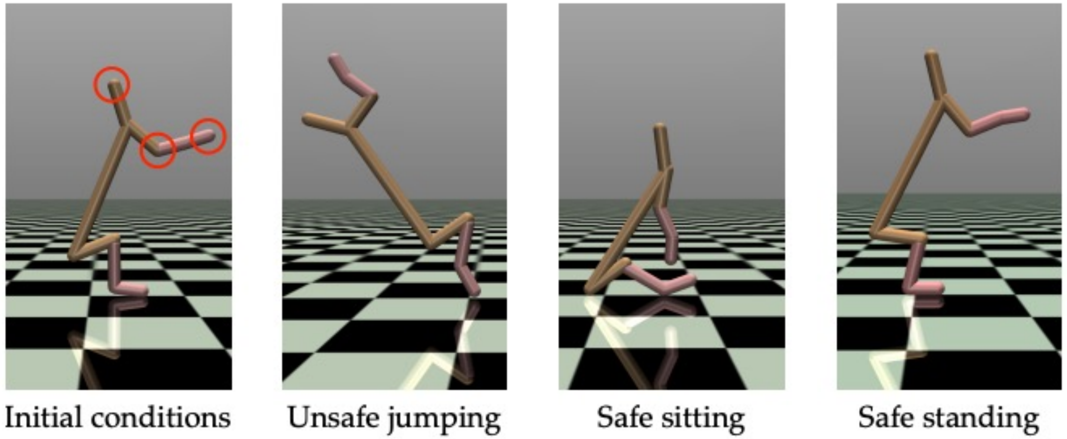



Fig. 6: Learned half-cheetah safety policies aimed to keep the head and front leg off the ground. *Left to right:* episode starting configuration; an unsafe jumping policy learned using a sum of discounted heights; a safe sitting policy learned using discounted safety or (less reliably) discounted sum of contact penalties; a safe standing policy learned using discounted safety.

Fisac et al 2019 https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8794107
https://scholar.google.com/citations?hl=en&user=HvjirogAAAAJ&view_op=list_works

# Takeaways

- Safety is hard.
- Control theoretic frameworks provide an interpretable inductive bias
- Active research in "data-driven safety"
  - Using data to inform parameters of safety models
  - Using safety models to inform data-driven solutions

# Thank you!

- Good luck on your project!

| | | | | |
|---|---|---|---|---|
| 9 | Mon, Nov 25 | Safe control II | | |
| 9 | Wed, Nov 27 | No lecture | | |
| 9 | Fri, Nov 29 | | Homework 3 | |
| 10 | Mon, Dec 2 | Final presentations I | | |
| 10 | Wed, Dec 4 | Final presentations II | | |
| Finals | Wed, Dec 11 | | Final report | |
| Finals | Fri, Dec 13 | | Homework & talk summaries | |

- Homework and talk summaries need to be submitted by Friday finals week
- Please do course evaluation – if more people enroll in the future, I can get a TA/grader ☺