From Neural Certificates to Certificate-based RL in Large-Scale Autonomy Design

Chuchu Fan
Assistant Professor of AeroAstro and LIDS
REALM Lab: REliable Autonomous systems Lab at MIT
chuchu@mit.edu

IFAC Workshop on Data-Driven Verification and Control of Cyber-Physical Systems

Massachusetts Institute of Technology

REALM

Challenges in autonomous safe navigation

A large fleet of agents

High-dimensional state and input spaces

Unknown or imprecise system and environment models

Interactions with other uncontrolled agents

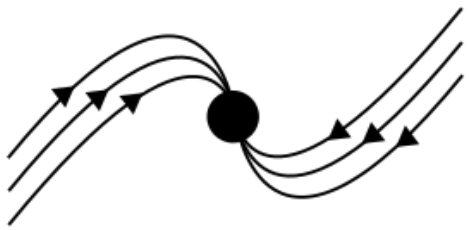
What can ML and control theory help with solving the above challenges?

Control certificate informed control

Certificate functions are from **control theory** to prove desired system properties.

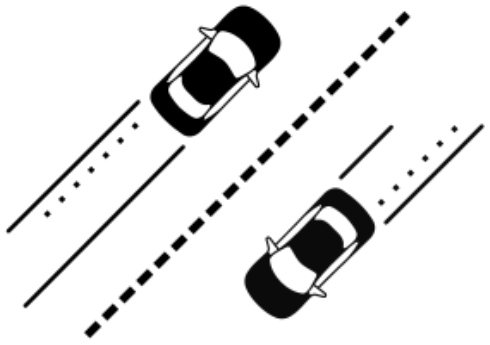
Lyapunov Function

Certifies stability of a fixed point



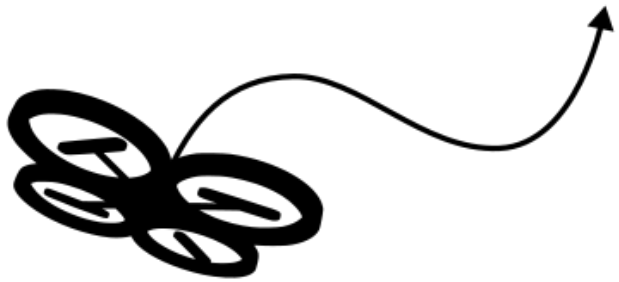
Barrier Function

Certifies invariance of a region



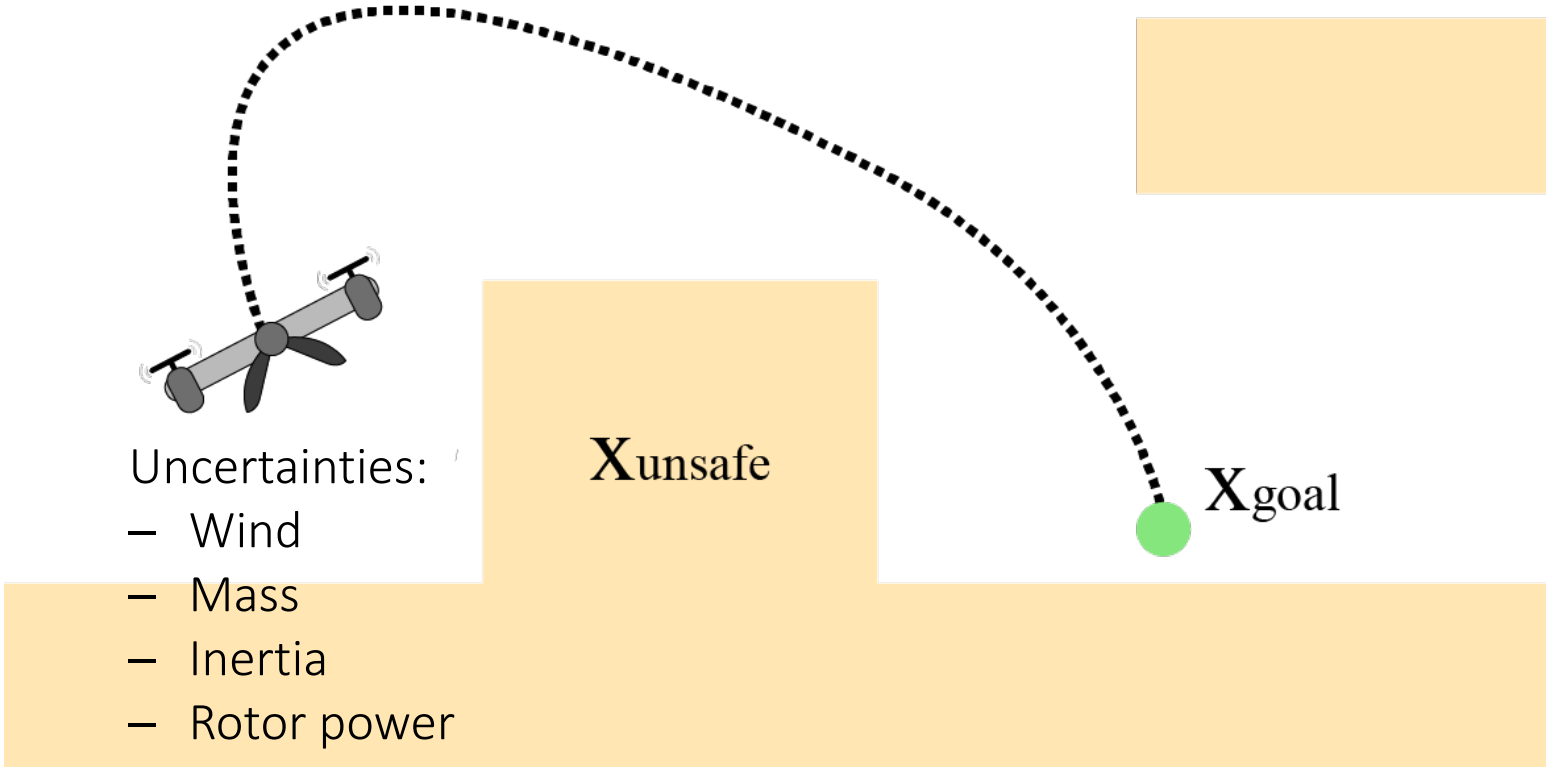
Contraction Metric

Certifies ability to track arbitrary trajectories



Control certificate informed control

Certificate functions are from **control theory** to prove desired system properties.



Uncertainties:

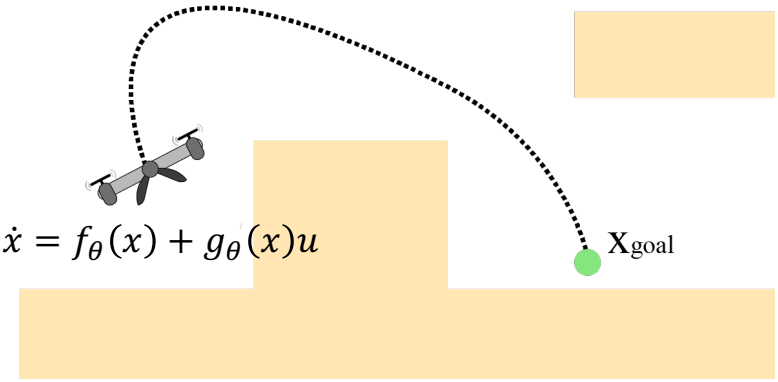
- Wind
- Mass
- Inertia
- Rotor power

…

How can we prove that the drone can reach the goal while avoiding the obstacles?

If we have a semi-perfect state estimation, many approaches can prove the success of a reach-avoid problem: MPC, HJ, reachability-based control etc.

Control certificate informed control



From a control certificate point of view, a robust Control Lyapunov Barrier function (rCLBF) can serve the purpose of certifying stabilize-avoid problems

Lyapunov

$$V(x_{\text{goal}}) = 0$$

$$V(x) > 0, \forall x \in \mathcal{X} \setminus x_{\text{goal}}$$

$$V(x) \leq c, \forall x \in \mathcal{X}_{\text{safe}}$$

$$V(x) > c, \forall x \in \mathcal{X}_{\text{unsafe}}$$

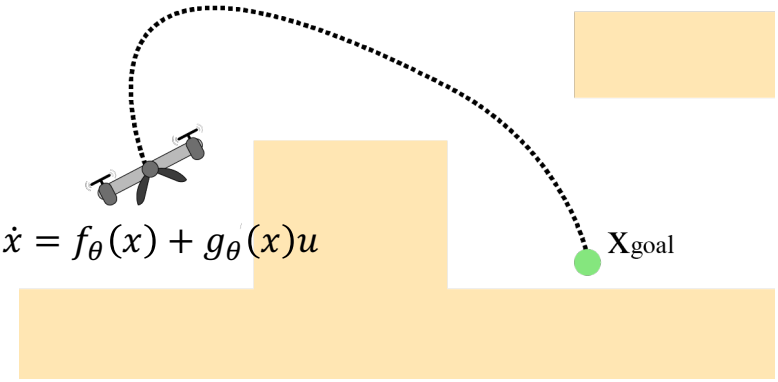
$$\inf_u L_{f_\theta} V + L_{g_\theta} V u + \lambda V(x) \leq 0, \ \forall x \in \mathcal{X} \setminus x_{\text{goal}}$$

Barrier

Theorem: If we can find such a V for a control policy u, then the closed-loop system is robustly safe and stable in terms of goal-reaching.

There are many approaches to find such a V, such as SoS, Simulation-guided synthesis. But the computational complexity has been a bottleneck so far.

Proposed approach: Neural certificate functions



From a control certificate point of view, a robust Control Lyapunov Barrier function (rCLBF) can serve the purpose of certifying stabilize-avoid problems

$$V(x_{\text{goal}}) = 0$$

$$V(x) > 0, \forall x \in \mathcal{X} \setminus x_{\text{goal}}$$

$$V(x) \leq c, \forall x \in \mathcal{X}_{\text{safe}}$$

$$V(x) > c, \forall x \in \mathcal{X}_{\text{unsafe}}$$

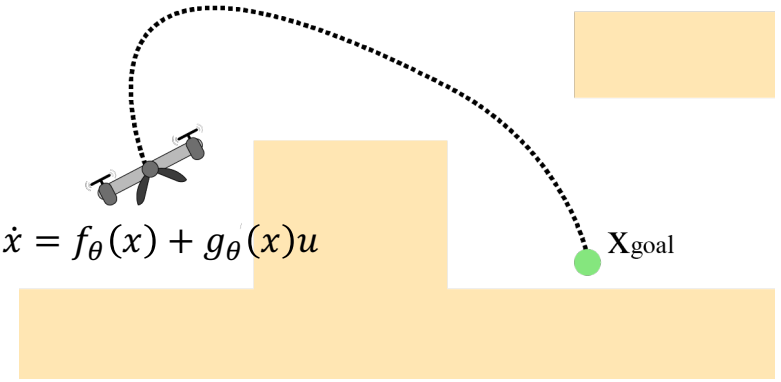
$$\inf_u L_{f_\theta} V + L_{g_\theta} V u + \lambda V(x) \leq 0, \ \forall x \in \mathcal{X} \setminus x_{\text{goal}}$$

[Dawson 21] u and V can be represented as NNs, with loss $\mathcal{L}$ being

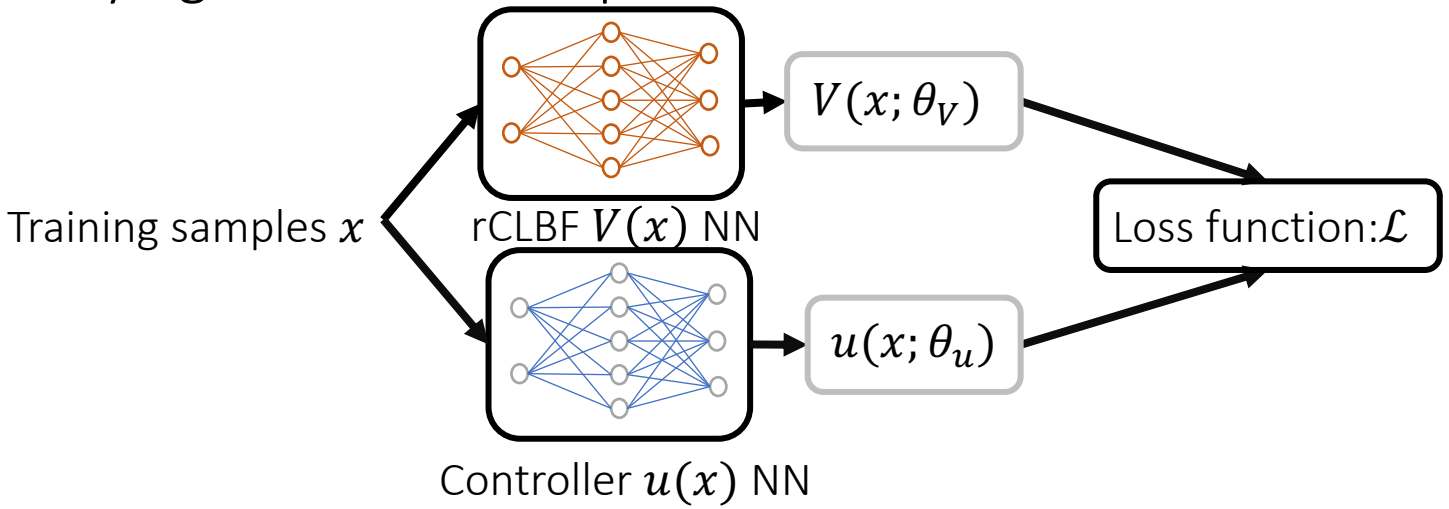
$$\inf_{V,u} \sup_{x \in \mathcal{X}} \Bigg(V^2(x_{\text{goal}}) + a_1 \frac{1}{N_{safe}} \sum_{x \in \mathcal{X}_{\text{safe}}} \max(V(x) - c, 0) + a_2 \frac{1}{N_{unsafe}} \sum_{x \in \mathcal{X}_{\text{safe}}} \max(V(x) - c, 0)$$

$$+ a_3 \frac{1}{N_{train}} \sum_{x \in \mathcal{X}} r(x) \sum_i \max(L_{f_{\theta_i}} V + L_{g_{\theta_i}} V u + \lambda V(x), 0) \Bigg)$$

Proposed approach: Neural certificate functions



From a control certificate point of view, a robust Control Lyapunov Barrier function (rCLBF) can serve the purpose of certifying stabilize-avoid problems

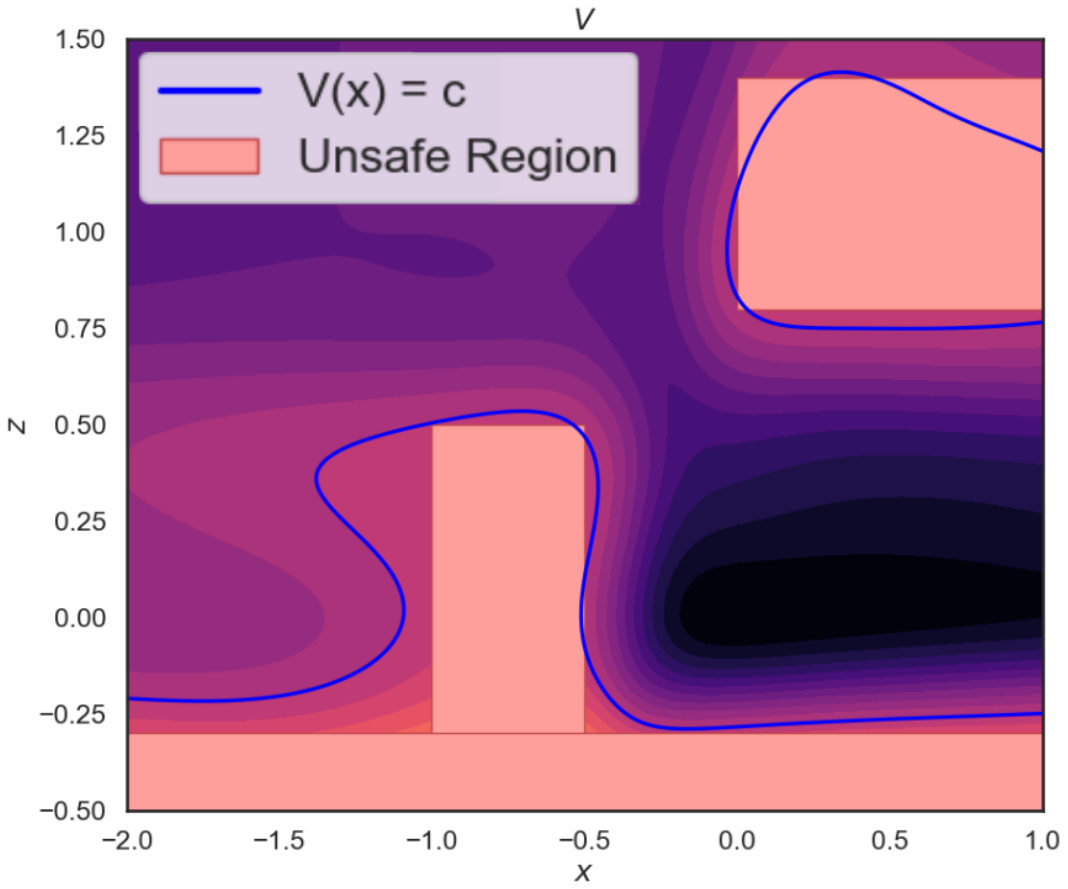


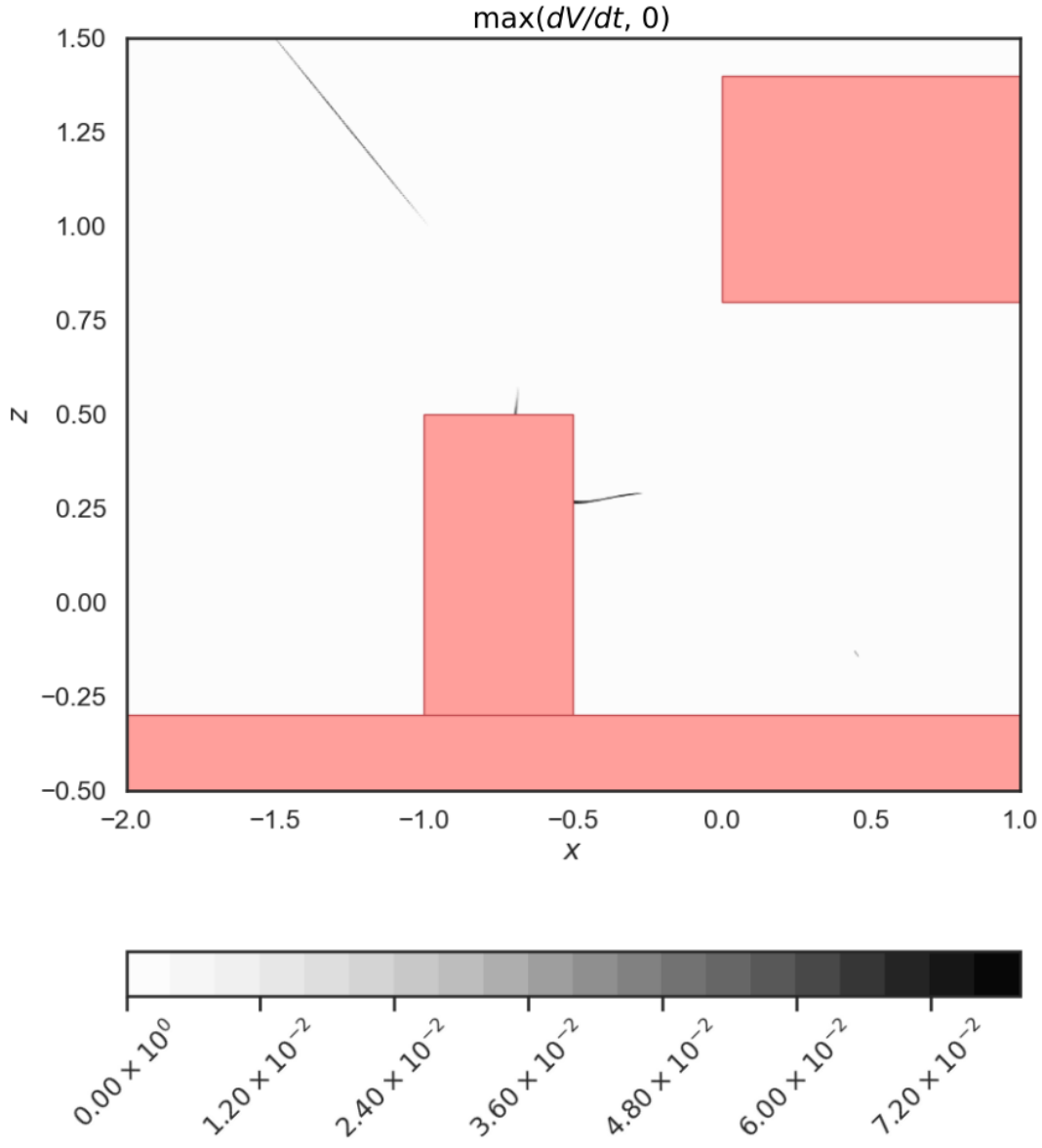
[Dawson 21] u and V can be represented as NNs, with loss $\mathcal{L}$ being

$$\inf_{V,u} \sup_{x\in\mathcal{X}} \Bigg(V^2(x_{\text{goal}}) + a_1 \frac{1}{N_{safe}} \sum_{x\in\mathcal{X}_{\text{safe}}} \max(V(x) - c, 0) + a_2 \frac{1}{N_{unsafe}} \sum_{x\in\mathcal{X}_{\text{safe}}} \max(V(x) - c, 0)$$

$$+ a_3 \frac{1}{N_{train}} \sum_{x\in\mathcal{X}} r(x) \sum_i \max(L_{f_{\theta_i}} V + L_{g_{\theta_i}} V u + \lambda V(x), 0) \Bigg)$$

Proposed approach: Neural certificate functions





Neural control certificates can help with

Single-agent certified learning-based control (robust, online)

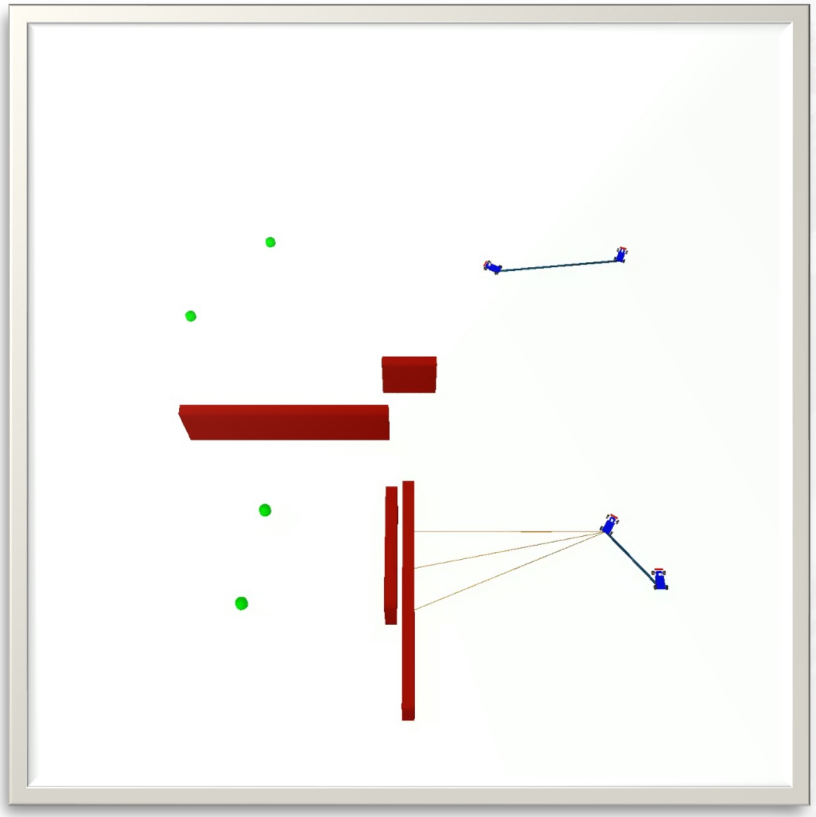
Generalization in a large fleet of agents

Handle high-dimensional state and input spaces

Combined with RL for unknown or imprecise models

Provide insights on other uncontrolled agents react to autonomous agents

Neural certificates for multi-agent systems



To handle multi-agent systems and generalize well to different settings, we must use decentralized certificates.

Example: decentralize barrier certificate

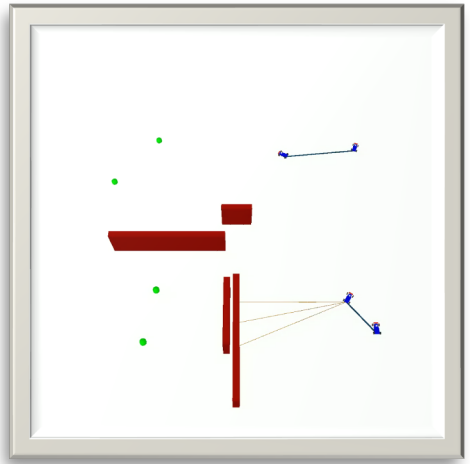
1. A decentralized barrier certificates with a decentralized controller is defined with observations o_i (e.g., LiDAR measurements, images).

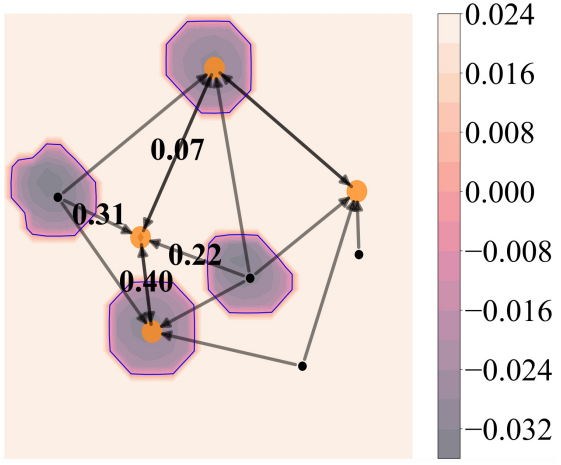
$$h_i(x_i, o_i) \leq 0 \text{ for } (x_i, o_i) \in \mathcal{X}_{\text{safe}}$$

$$h_i(x_i, o_i) > 0 \text{ for } (x_i, o_i) \in \mathcal{X}_{\text{unsafe}}$$

$$\inf_{u_i} \left[\nabla_{x_i} h_i \cdot \dot{x}_i + \nabla_{o_i} h_i \cdot \dot{o}_i + \alpha(h_i) \right] \leq 0$$

h_i can be parameterized using Graph Neural Networks with local information and communications.





Example: decentralize barrier certificate

1. A decentralized barrier certificates with a decentralized controller is defined with observations o_i (e.g., LiDAR measurements, images).

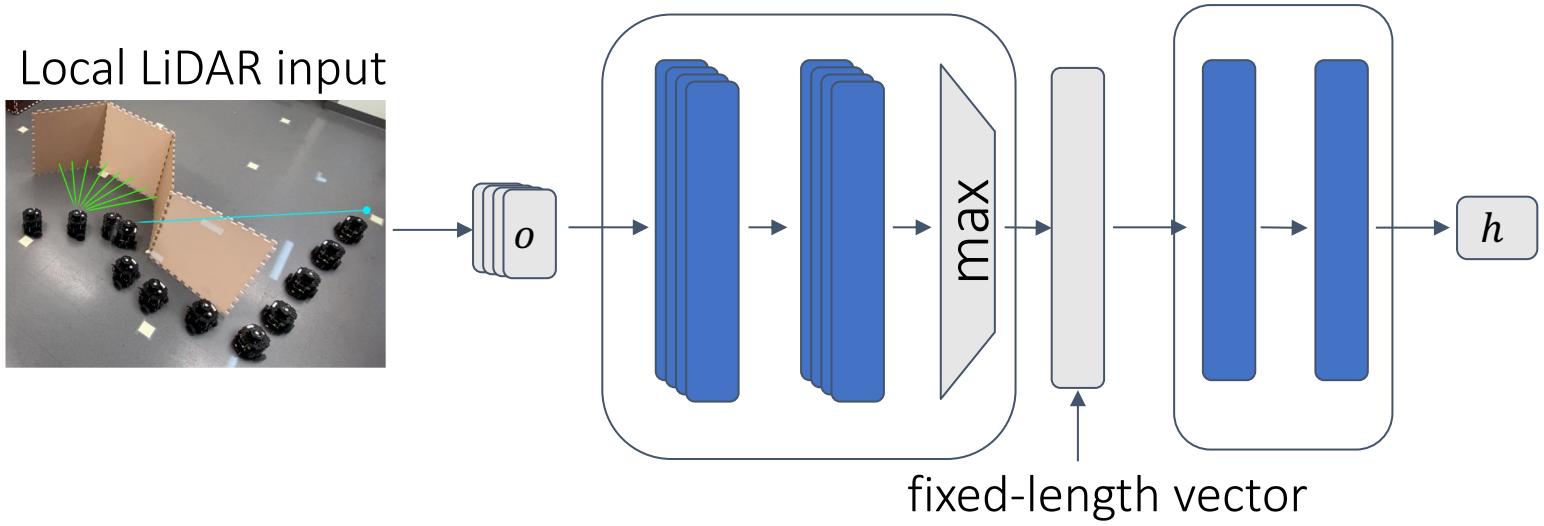
$$h_i(x_i, o_i) \leq 0 \text{ for } (x_i, o_i) \in \mathcal{X}_{\text{safe}}$$
$$h_i(x_i, o_i) > 0 \text{ for } (x_i, o_i) \in \mathcal{X}_{\text{unsafe}}$$
$$\inf_{u_i} \left[\nabla_{x_i} h_i \cdot \dot{x}_i + \nabla_{o_i} h_i \cdot \dot{o}_i + \alpha(h_i) \right] \leq 0$$

h_i can be parameterized using Graph Neural Networks with local information and communications.

In fact, $h = \max\{h_1, h_2, \cdots\}$ is a big global barrier certificate for safety.

Example: decentralize barrier certificate

1. A decentralized barrier certificates with a decentralized controller is defined with observations o_i (e.g., LiDAR measurements, images).
2. To handle the time-varying observation o_i, use a permutation-invariant embedding (e.g., PointNet).



Example: decentralize barrier certificate

1. A decentralized barrier certificates with a decentralized controller is defined with observations o_i (e.g., LiDAR measurements, images).

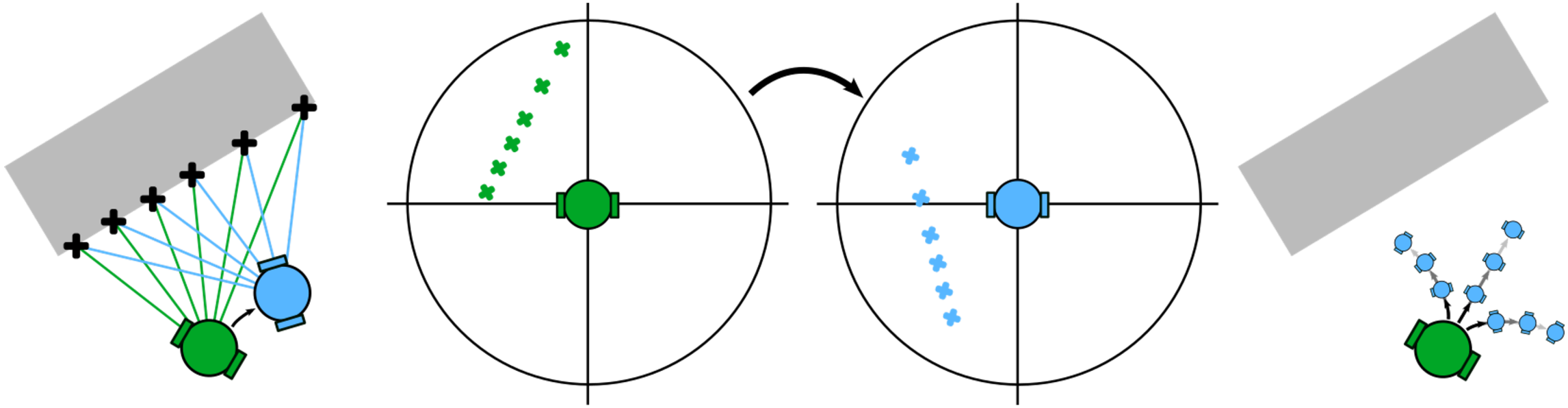
$$h_i(x_i, o_i) \leq 0 \text{ for } (x_i, o_i) \in \mathcal{X}_{\text{safe}}$$

$$h_i(x_i, o_i) > 0 \text{ for } (x_i, o_i) \in \mathcal{X}_{\text{unsafe}}$$

$$\inf_{u_i} \left[\nabla_{x_i} h_i \cdot \dot{x}_i + \nabla_{o_i} h_i \cdot \dot{o}_i + \alpha(h_i) \right] \leq 0$$

2. To handle the time-varying observation o_i, use a permutation-invariant embedding (e.g., PointNet).
3. Approximate lookahead for estimating $\dot{o}_i$.

Approximate lookahead for $\dot{o}_i$



control inputs
↓
workspace translation + rotation

workspace translation + rotation
↓
affine update to lidar observations

search over
control inputs

Generative models like NeRF can be used for camera image inputs [Tong ICRA 23].

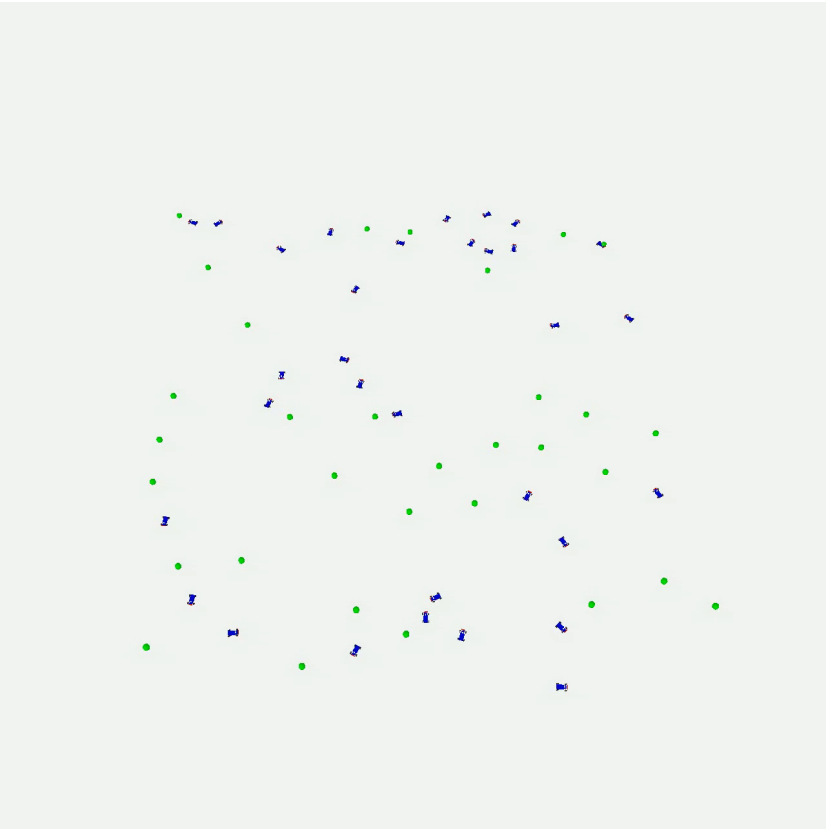
Example: decentralize barrier certificate

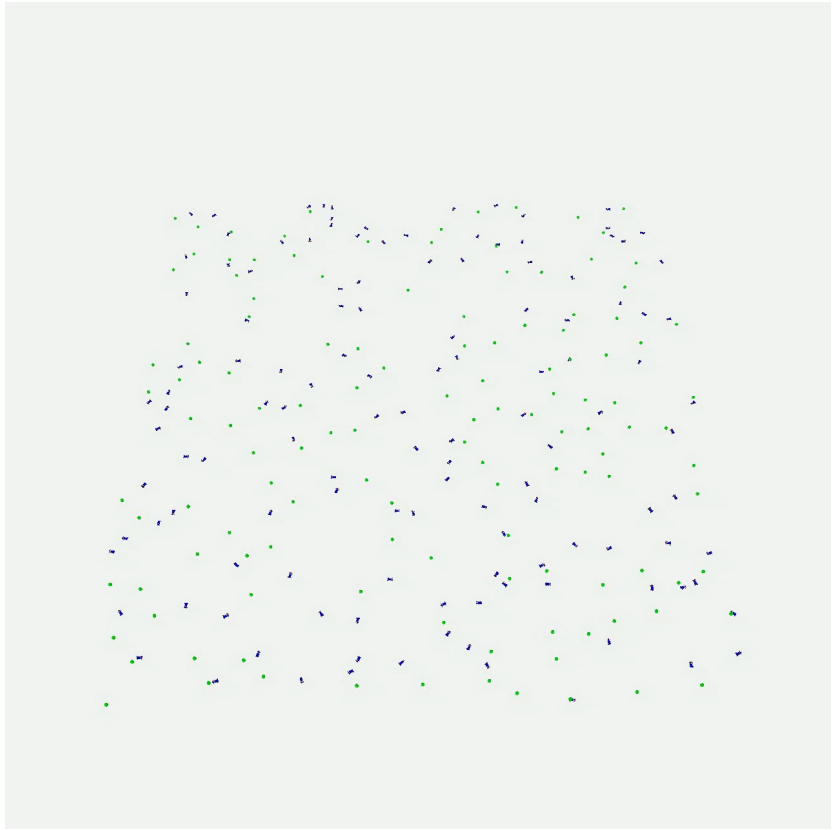
1. A decentralized barrier certificates with a decentralized controller is defined with observations o_i (e.g., LiDAR measurements, images).
2. To handle the time-varying observation o_i, use a permutation-invariant embedding (e.g., PointNet).
3. Approximate lookahead for estimating $\dot{o}_i$.

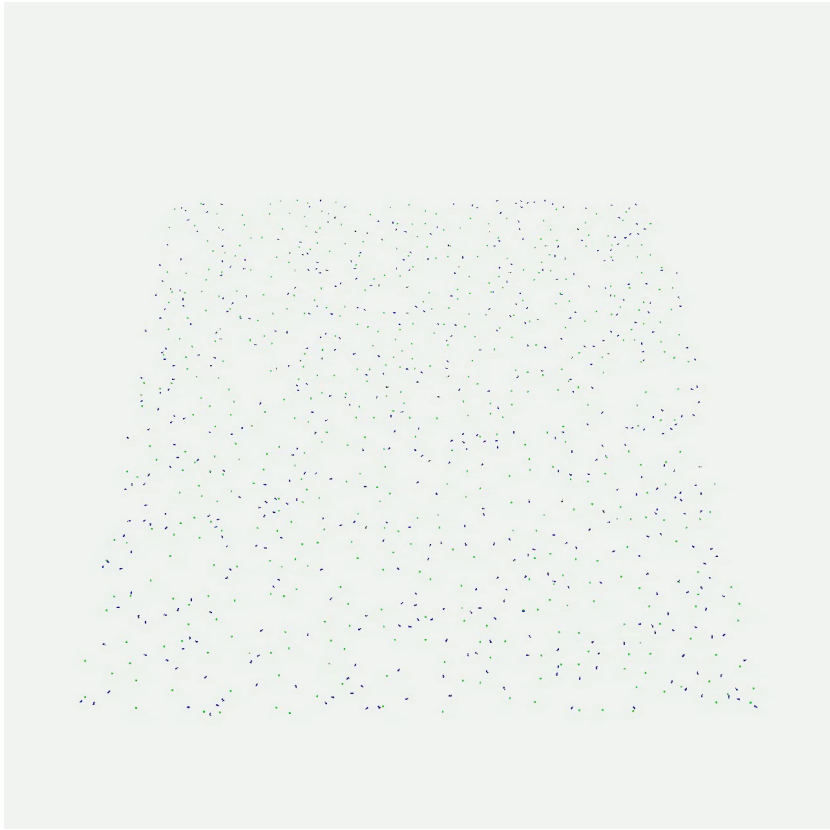
Each step can cause some loss on the guarantees that the learned candidate certificates can provide. Additional verification steps are needed. But we will see that empirically neural certificates can achieve great performance!

GNN-based neural CBF-based control

Trained with 16 Dubin cars, the resulting controller can be deployed on an arbitrary number of agents with the same dynamics.

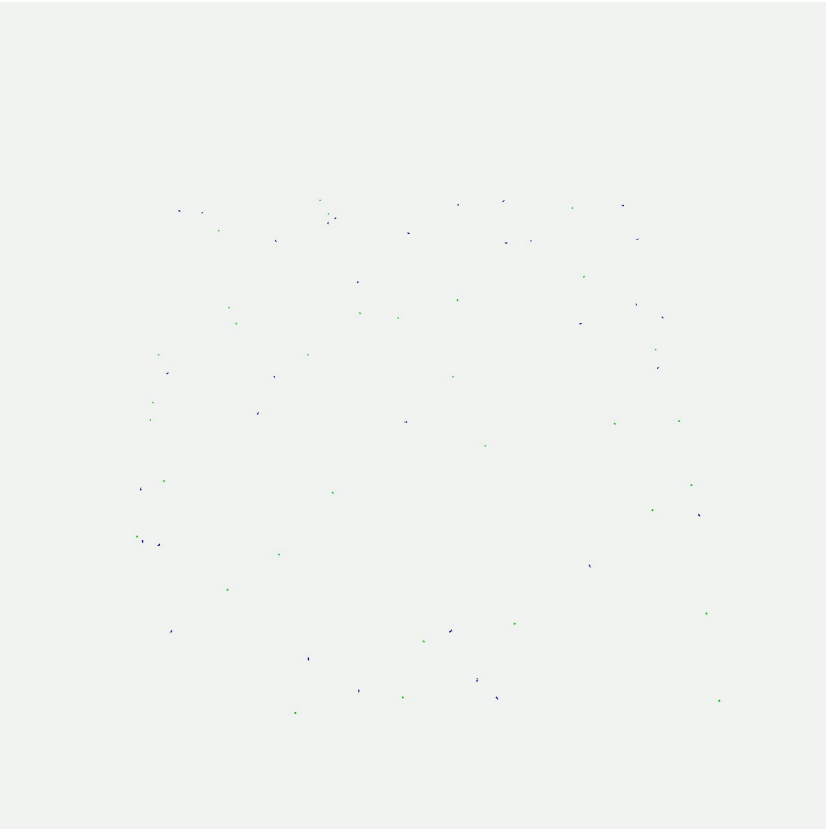
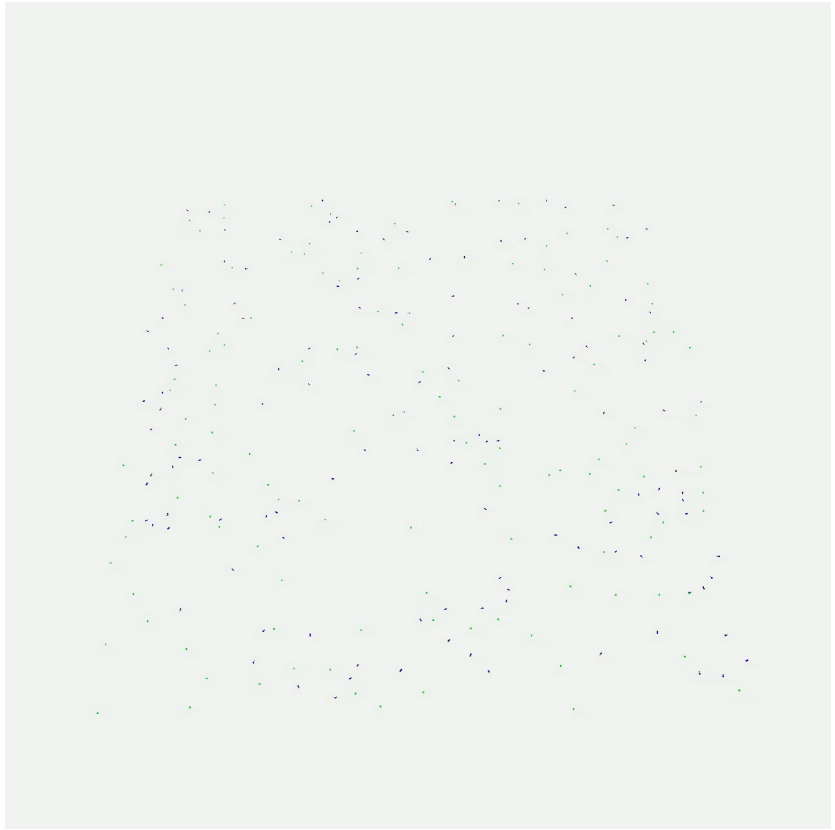
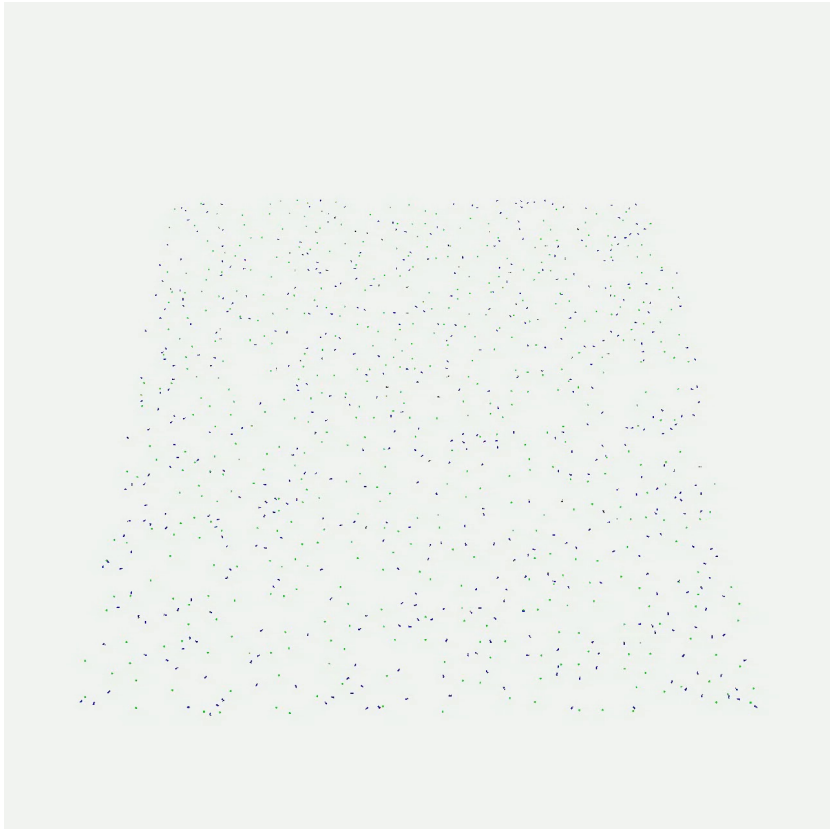






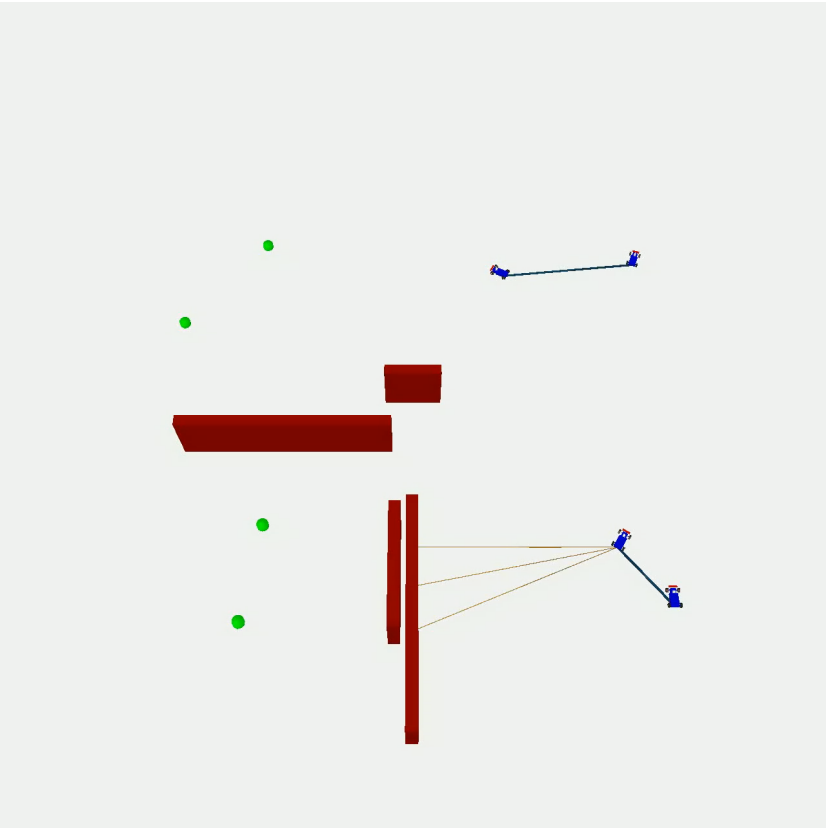
GNN-based neural CBF-based control

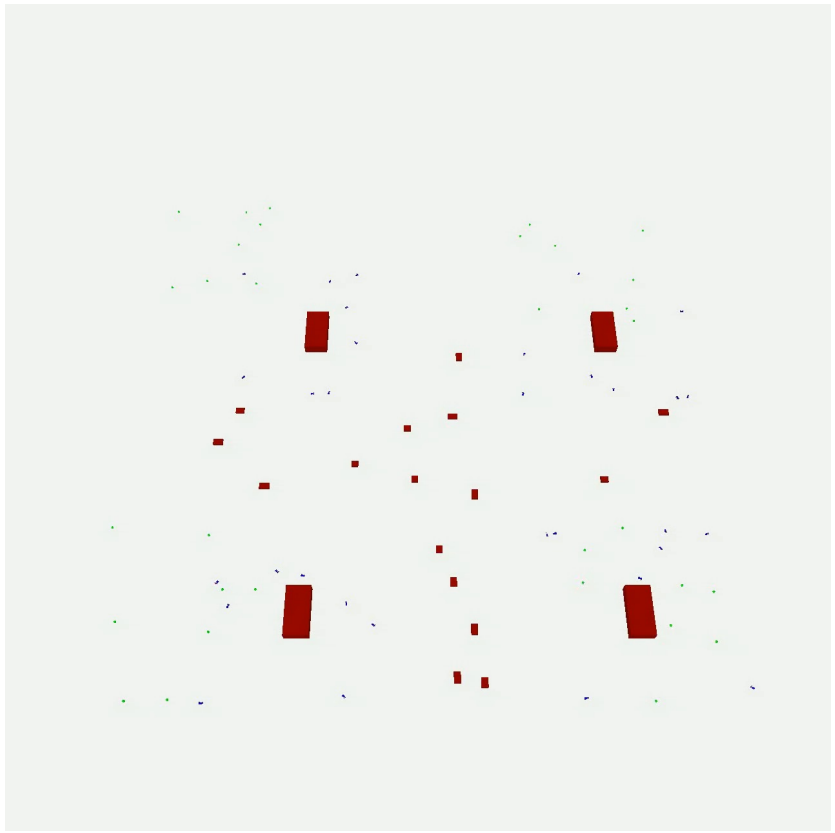
Trained with 16 Dubin cars, the resulting controller can be deployed on an arbitrary number of agents with the same dynamics, even with increased density.

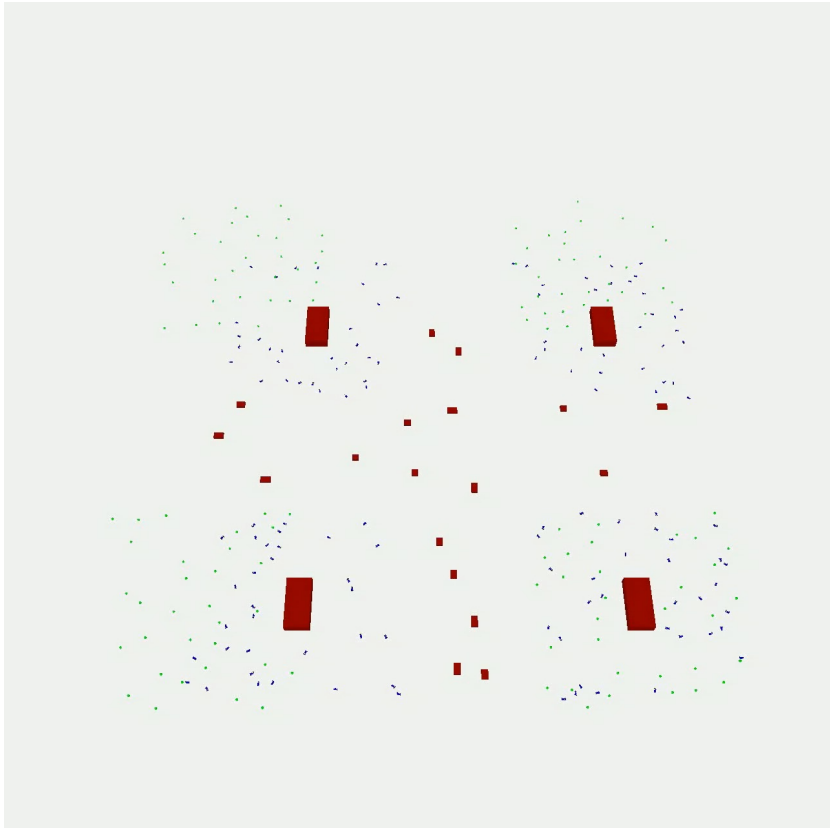




GNN-based neural CBF-based control

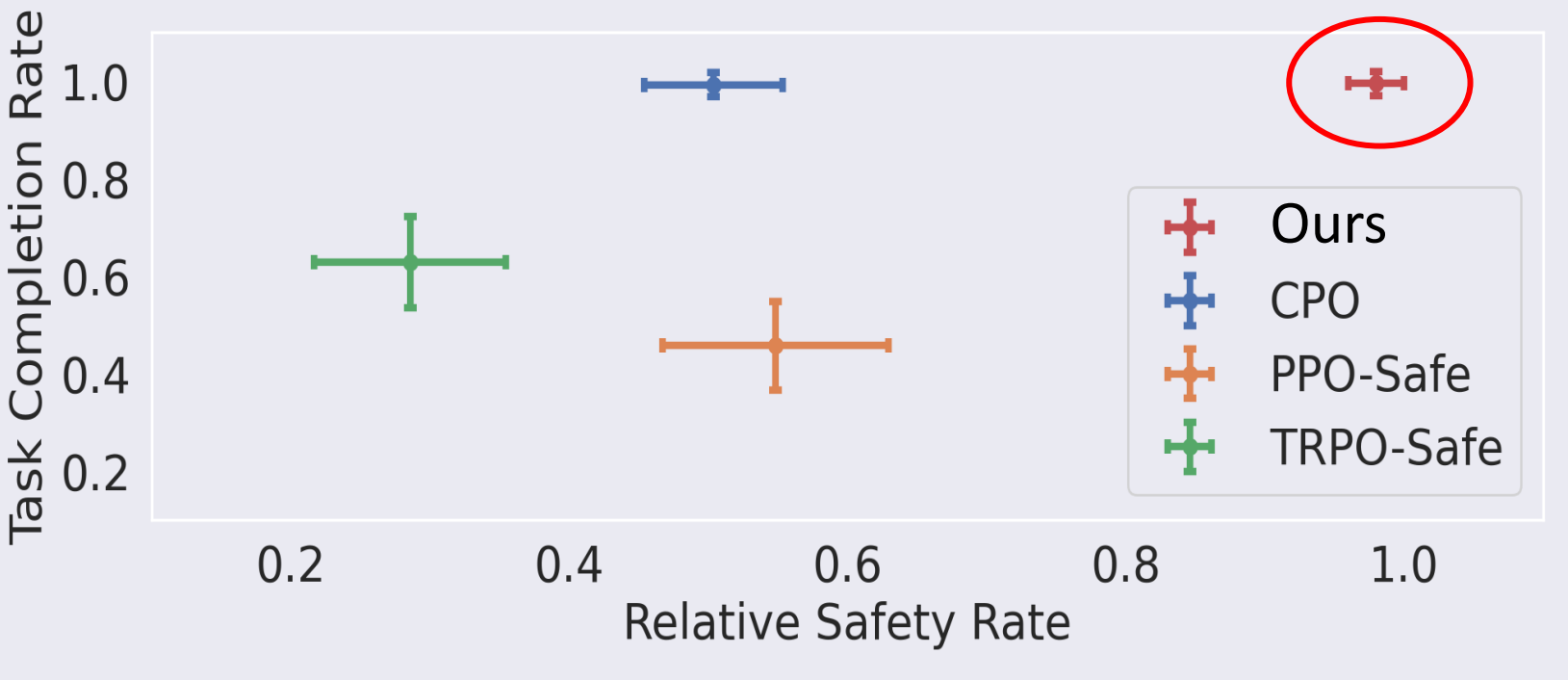
With LiDAR inputs, the decentralized controller can handle unseen (uncontrolled) obstacles of varying sizes and speed.







Comparison with state-of-the-art

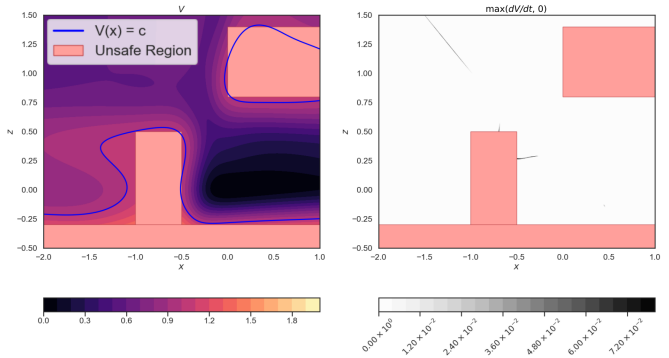


All environments contain 1024 agents. Our neural certified control policies can achieve **nearly 100% task completion rate and safety rate**, outperforming all leading safe RL methods.

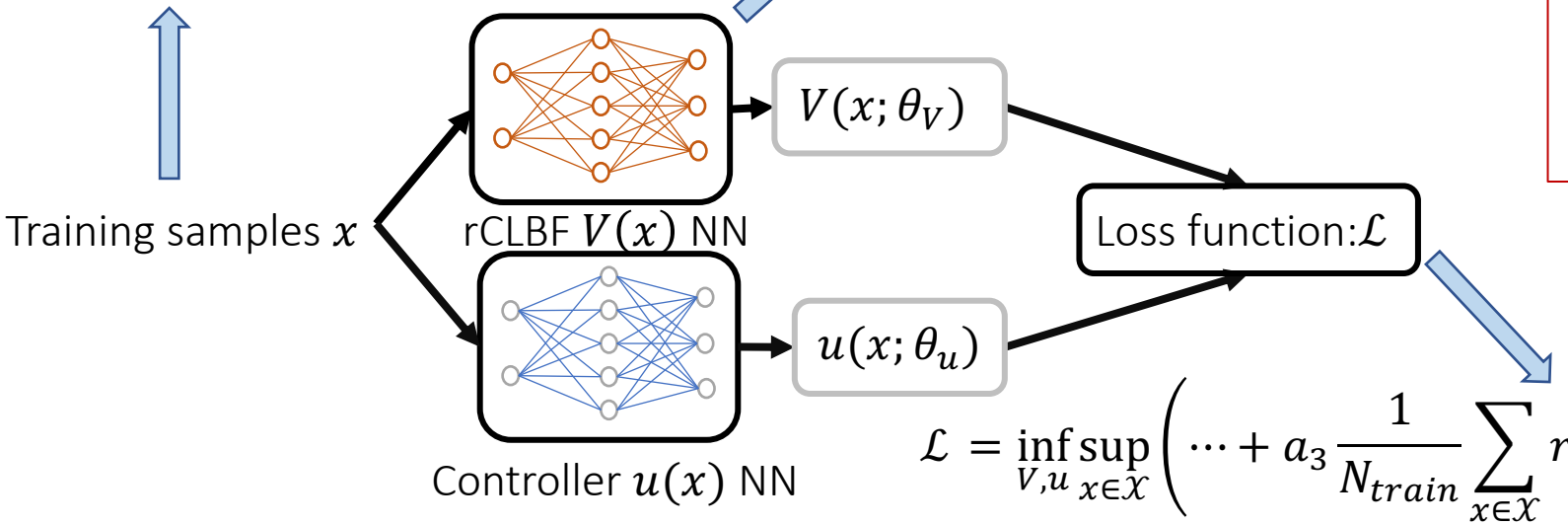
References: CPO [Achiam 2017] PPO-Safe [Schulman 2017], TRPO-Safe [Tessler 2019]

Limitations of current neural certificates

- Uniformly collecting samples in the state space is not feasible for high-dim systems.
- Collecting samples only along trajectories may not have good coverage.



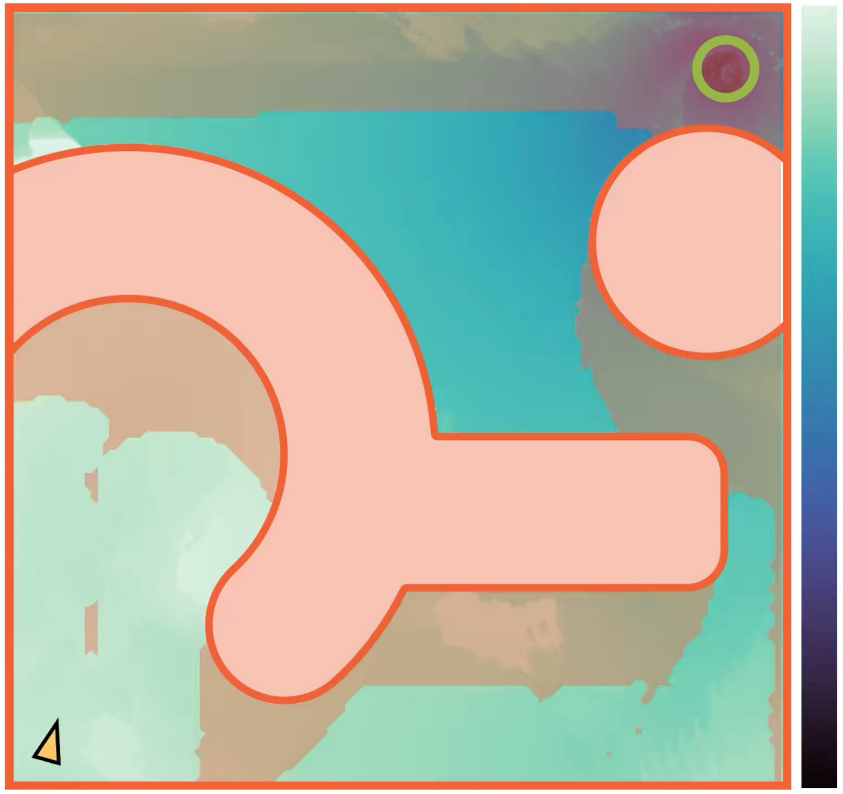
- We have to learn a new rCLBF if the environment changes.
- Separate CLF and CBF can lead to deadlocks (because CBF is myopic).



$$\mathcal{L} = \inf_{V,u} \sup_{x \in \mathcal{X}} \left(\cdots + a_3 \frac{1}{N_{train}} \sum_{x \in \mathcal{X}} r(x) \sum_i \max(L_{f_\theta} V + L_{g_\theta} V u + \lambda V(x), 0) \right)$$

- So far, neural certificates need dynamics information.
- But dynamics can be hard to accurately estimate.

Stabilize-avoid as constrained optimization problem



Let $\mathcal{X}_{\text{unsafe}} = \{x\colon h(x) > 0\}$

$$\min_{\pi} \quad J(\pi) = \sum_{k=0}^{\infty} l(x_k)$$

$$\text{s.t.} \quad h(x_k) \le 0, \qquad k \ge 0$$

$$x_{k+1} = f\big(x_k, \pi(x_k)\big)$$

$$l(x) = 0, \qquad x \in \text{Goal}$$
$$l(x) > 0, \qquad x \notin \text{Goal}$$

The policy value function

$V(x_k) = \min_u l(x_k, u) + V(x_{k+1})$

is a Lyapunov function

$h(x_k)$ is a barrier function for infinite horizon!

Standard approach: Lagrangian duality

Instead, we use the **epigraph form.**

Stabilize Reach and stay within the goal region

Avoid Avoid entering unsafe

Stabilize-avoid as constrained optimization problem

Standard form

$$\min_{x} \; J(x) \quad \text{s.t.} \; h(x) \le 0$$

$\equiv$

Epigraph form

$$\min_{x,z} \; z \quad \text{s.t.} \; h(x) \le 0, \; J(x) \le z$$

$\equiv$

$$\min_{z} \; z \quad \text{s.t.} \; \min_{x} \max\{h(x), J(x) - z\} \le 0$$

z is a "cost budget" (with units of J) for satisfying h

This is a **two-stage** optimization problem!

Stabilize-avoid constrained OCP

$$\min_{\pi} \quad J(\pi) = \sum_{k=0}^{\infty} l(x_k)$$

$$\text{s.t.} \quad h(x_k) \le 0, \quad k \ge 0$$

$$x_{k+1} = f\big(x_k, \pi(x_k)\big)$$

$\equiv$

Epigraph form Constrained OCP

$$\min_{z} \; z \quad \text{s.t.} \min_{\pi} \; \tilde{V}^{\pi}(x_0, z) \le 0$$

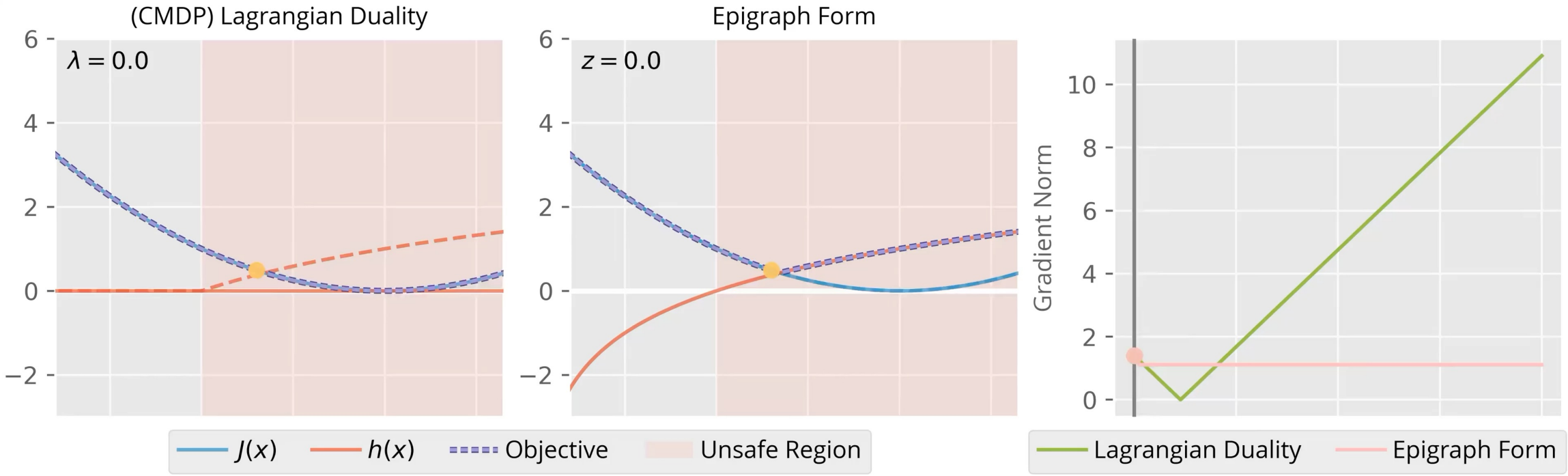
Where

$$\tilde{V}^{\pi}(x_0, z) = \max\left\{\max_{k} h(x_k), \sum_{k=0}^{\infty} l(x_k) - z\right\}$$

Why epigraph form?

The gradient for Lagrangian duality-based methods scales linearly with λ, which can cause optimization problems when λ grows large as states are unsafe.

The gradient for the epigraph form does not scale with z.



Varying λ and z on the same cost function and constraint function at a given point, the gradient norms (right) of the objective grow for Lagrangian duality but not for the epigraph form.

Solve epigraph form constrained OCP via DeepRL

Epigraph form constrained OCP (EF-COCP)

$$\min_{z} \quad z$$
$$\text{s.t.} \min_{\pi} \quad \tilde{V}^{\pi}(x_0, z) \leq 0$$

Where

$$\tilde{V}^{\pi}(x_0, z) = \max\left\{\max_{k} h(x_k), \sum_{k=0}^{\infty} l(x_k) - z\right\}$$

EFPPO for EF-COCP [So RSS23]

1. Solve inner problem over $[z_{\min}, z_{\max}]$ → $\tilde{V}^{\pi}(x, z)$ and $\pi(x, z)$ over the state space

2. Solve outer problem for each x_0 → $z^*(x_0)$

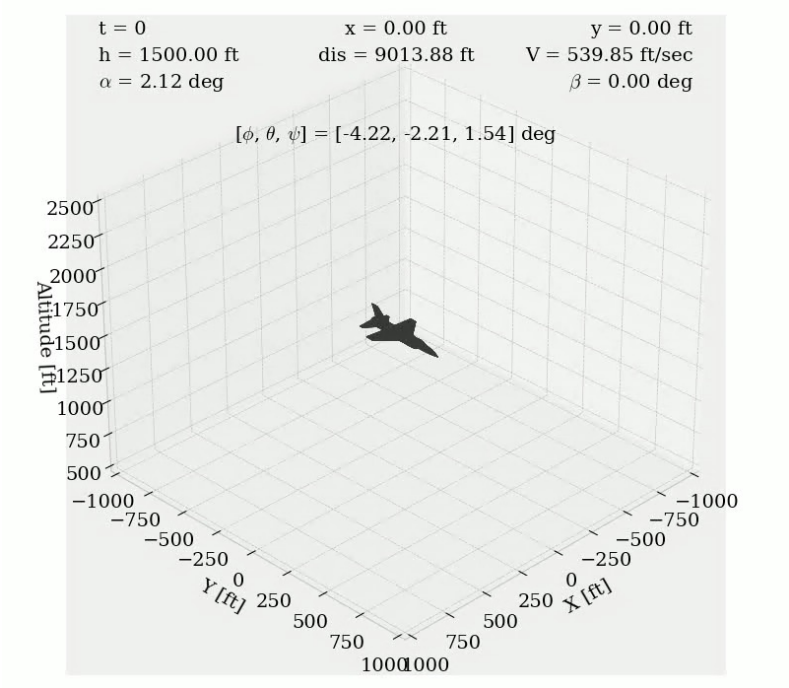
3. The final policy is then obtained as $\pi(x, z^*)$.

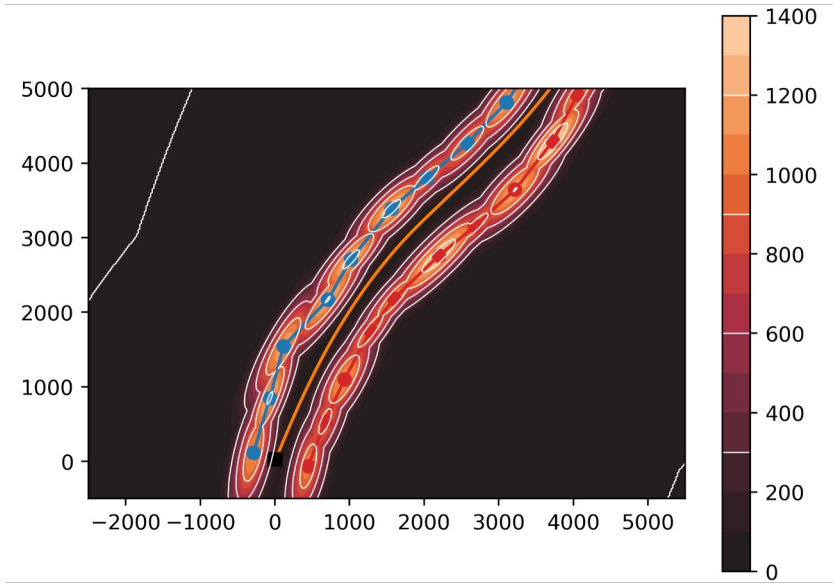
Hopper

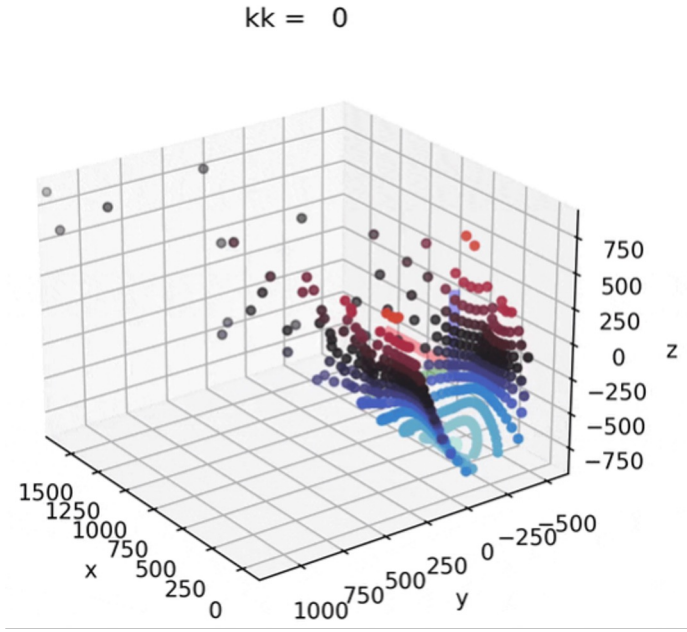
Average Example

Stabilize to the green region
Avoid torso dropping into red region
Avoid torso tilting too far

EFPPO for F-16

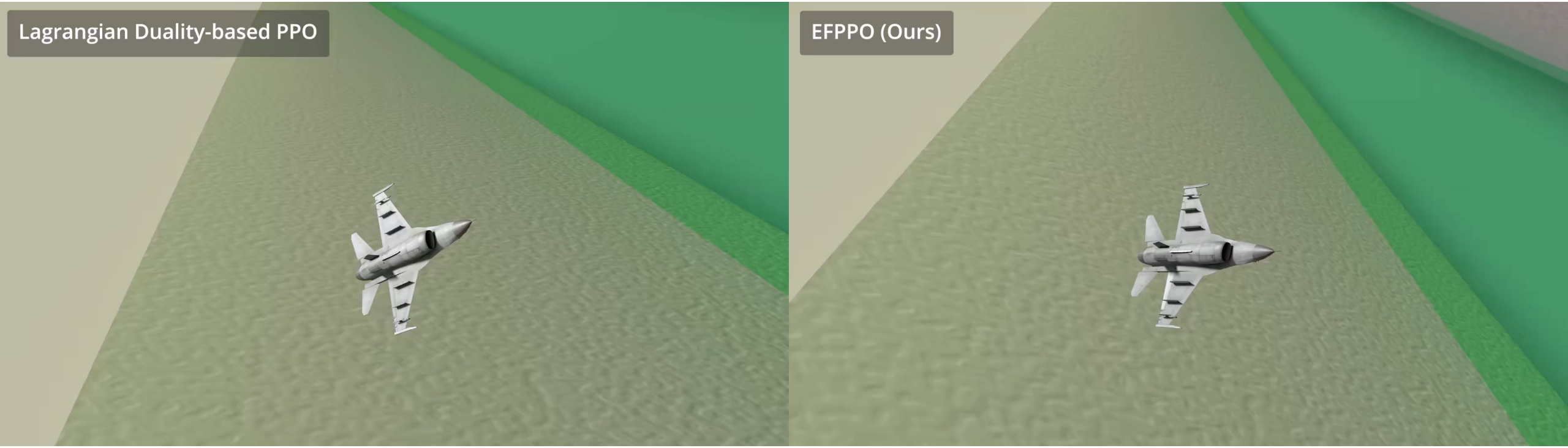




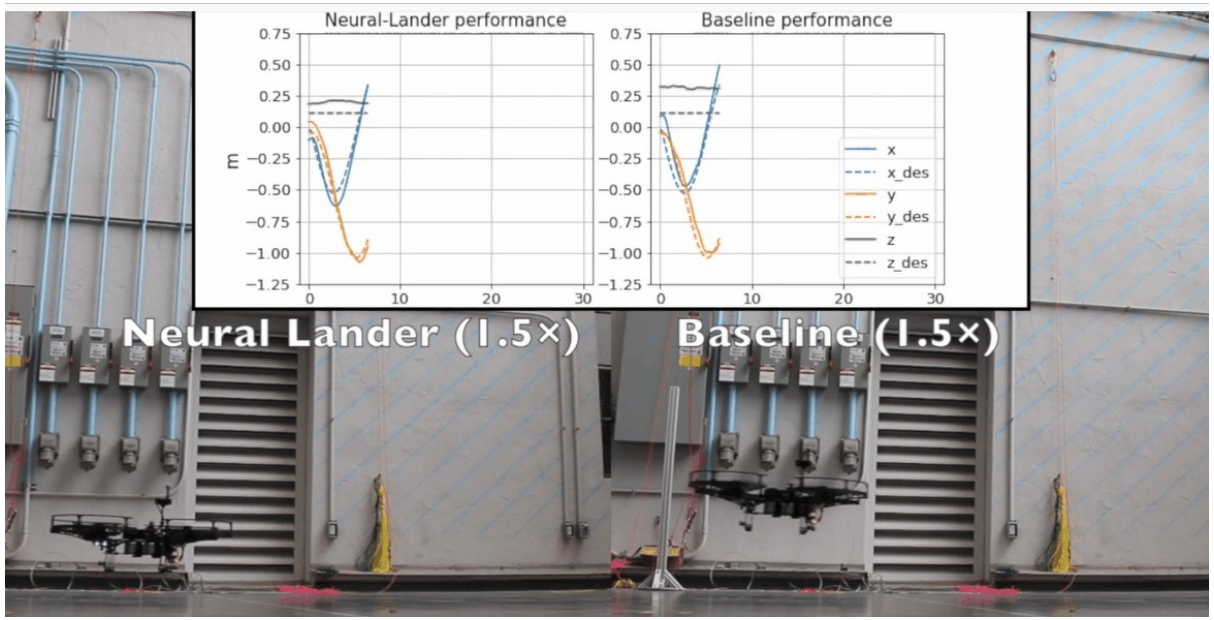


Model: An F-16 Simulink (black-box) model developed by AFRL with a 17-d state space and a 4-d input space [Heidlauf 18].

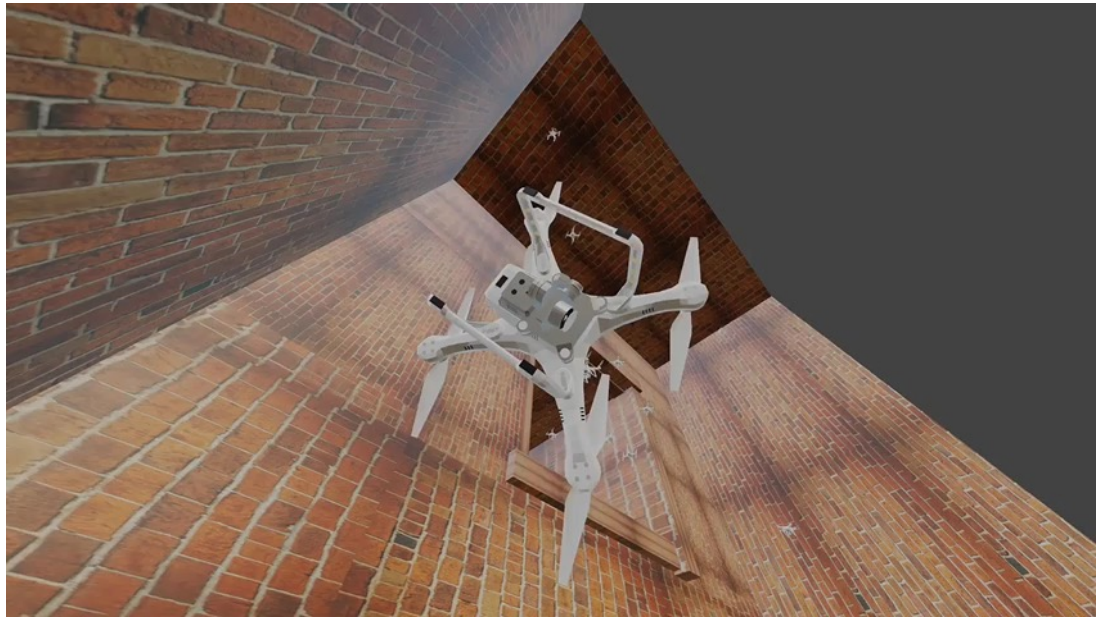
We further enhanced the model with a 3D-LiDAR for traveling through valleys (Top Gun!)


Lagrangian Duality-based PPO
EFPPO (Ours)

Other applications of neural certificates

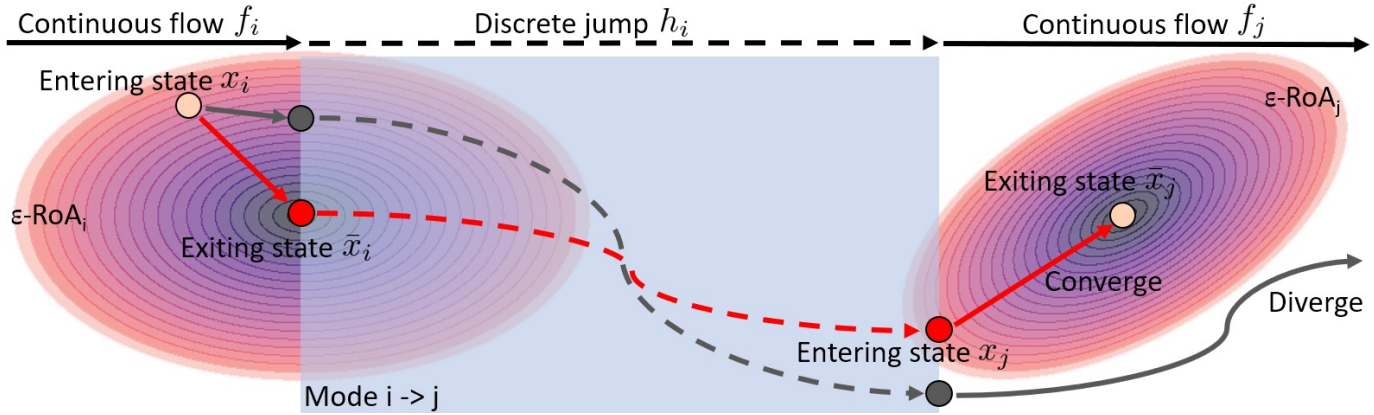


[Caltech Neural Lander] Video by Caltech CAST

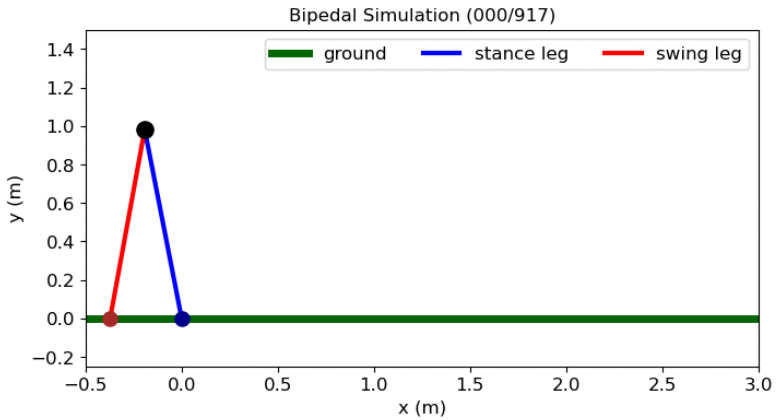


Precise drone control against prop wash using neural control contraction metrics. [Sun CoRL20]

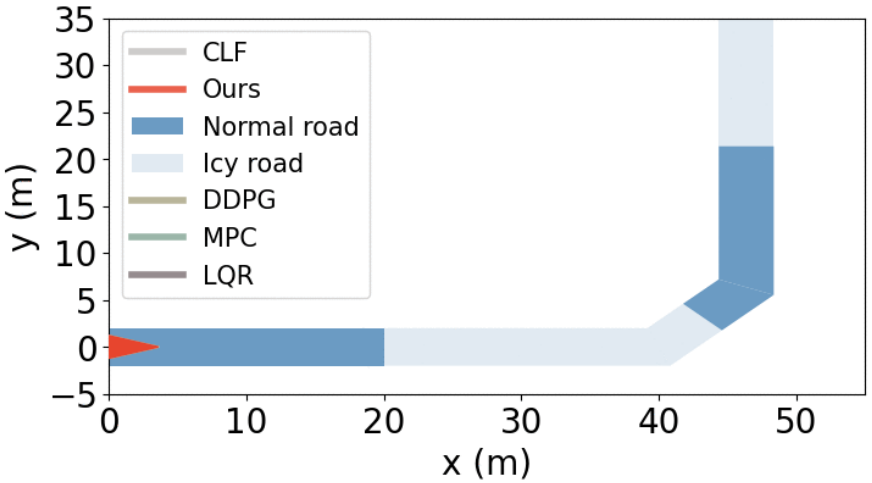
Other applications of neural certificates



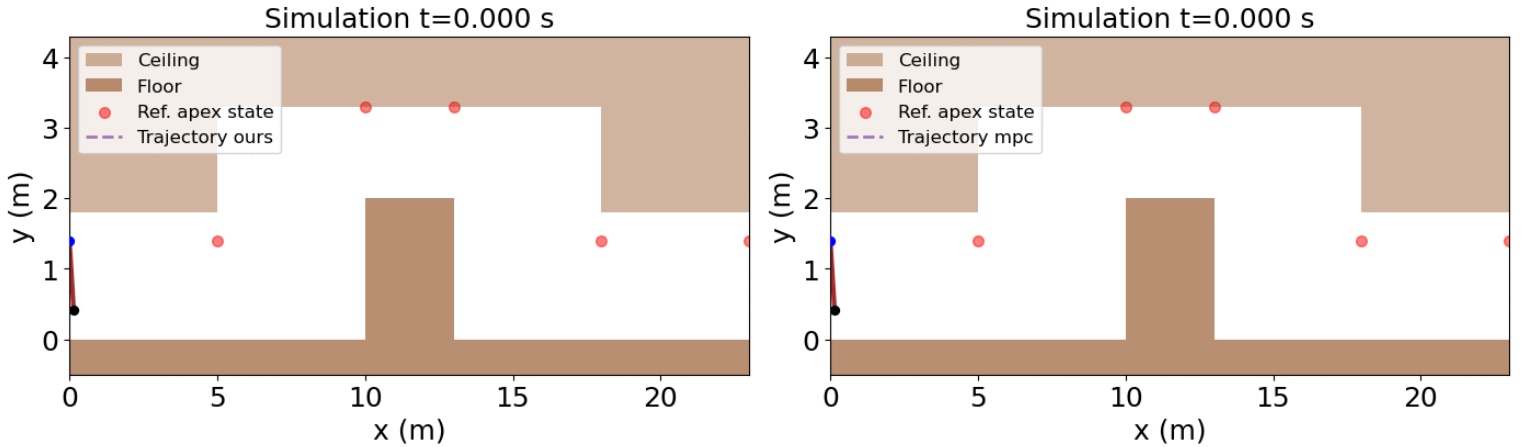
Hybrid system certificates with region-of-attraction planner [Meng L4DC23]



Control a bipedal robot



Drive on a partially slippery road



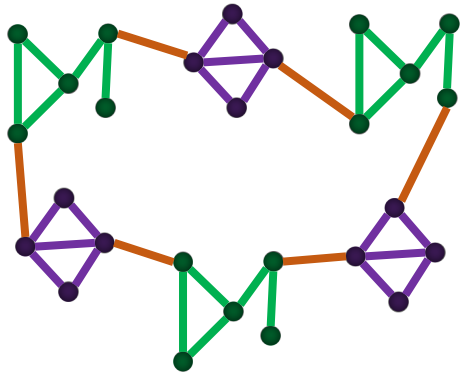
Ours

Robust MPC

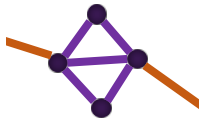
Control a jumping robot through a passageway

Other applications of neural certificates

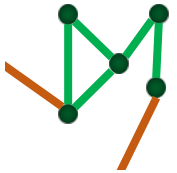
To stabilize



Only need to learn neural ISS-Lyapunov-based control for some nodes in



and



Compositional Neural Certificates for Networked Dynamical Systems [Zhang 23]

L4DC Oral, June 2023

References

Z. Qin, K. Zhang, Y. Chen, J. Chen, C. Fan, "Learning Safe Multi-agent Control with Decentralized Neural Barrier Certificates", *International Conference on Learning Representations*, 2021.

C. Dawson, B. Lowenkamp, D. Goff, C. Fan. "Learning Safe, Generalizable Perception-based Hybrid Control with Certificates", *IEEE Robotics and Automation Letters*, 2022.

O. So, C. Fan. "Solving Stabilize-Avoid Optimal Control via Epigraph Form and Deep Reinforcement Learning", *Robotics: Science and Systems*, 2023

C. Dawson, S. Gao, C. Fan. "Safe Control with Learned Certificates: A Survey of Neural Lyapunov, Barrier, and Contraction Methods for Robotics and Control", *IEEE Transactions on Robotics*, 2023

Realm Group

Realm GitHub

Summary

Single-agent certified learning-based control

Generalization in a large fleet of agents

Handle high-dimensional state and input spaces

Combined with RL for unknown or imprecise models

Provide insights on other uncontrolled agents react to autonomous agents