Performance of DeepReach on High-Dimensional Systems

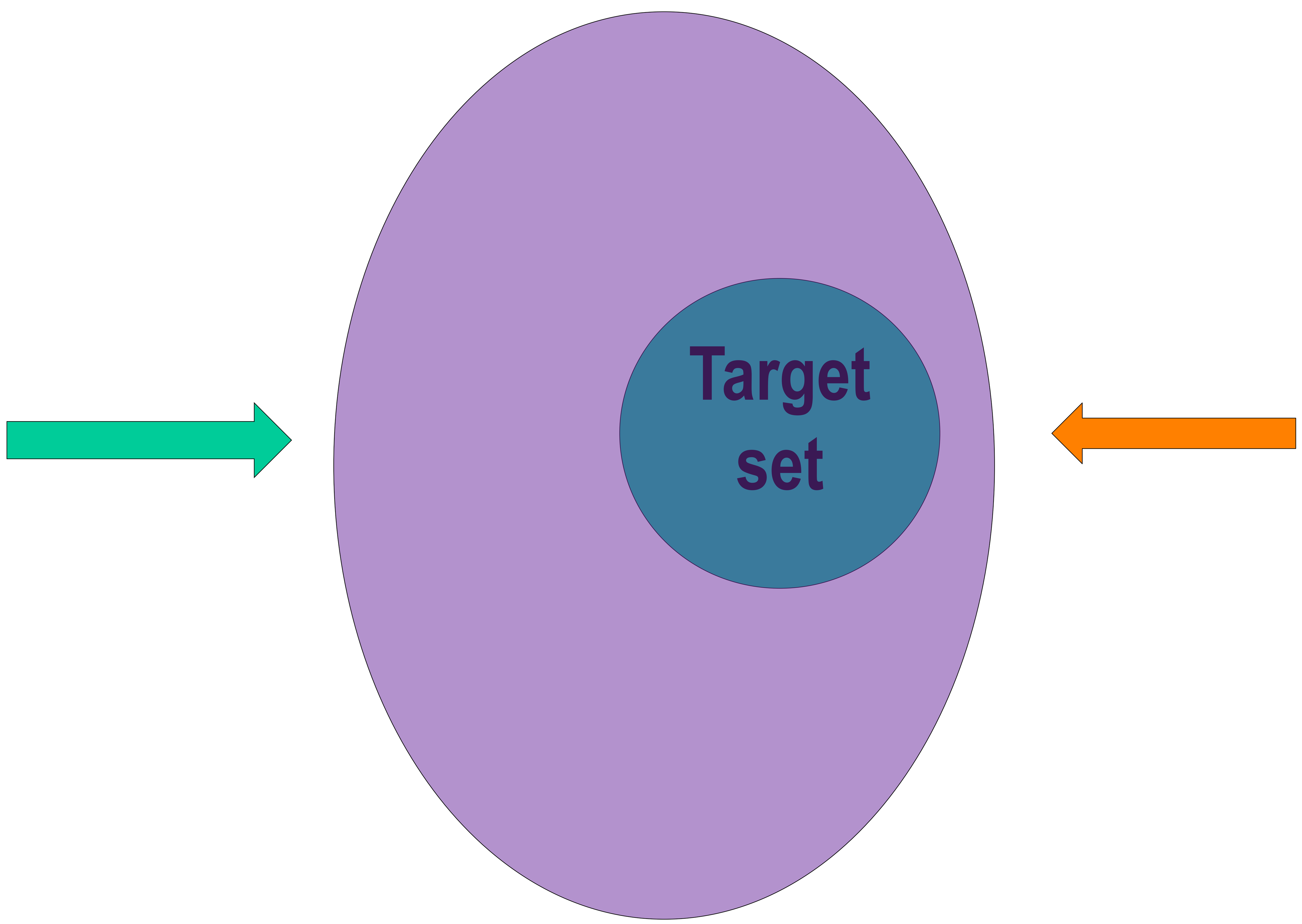
Introduction & Motivation:
- Provide safety guarantees for safety-critical autonomous/robotic systems

e.g. autonomous driving,
drone delivery,
legged robots for space exploration

Problem Formulation: Hamilton-Jacobi Reachability analysis

Problem Formulation: Hamilton-Jacobi Reachability analysis

BRT



Here target set is goal set (or failure set)

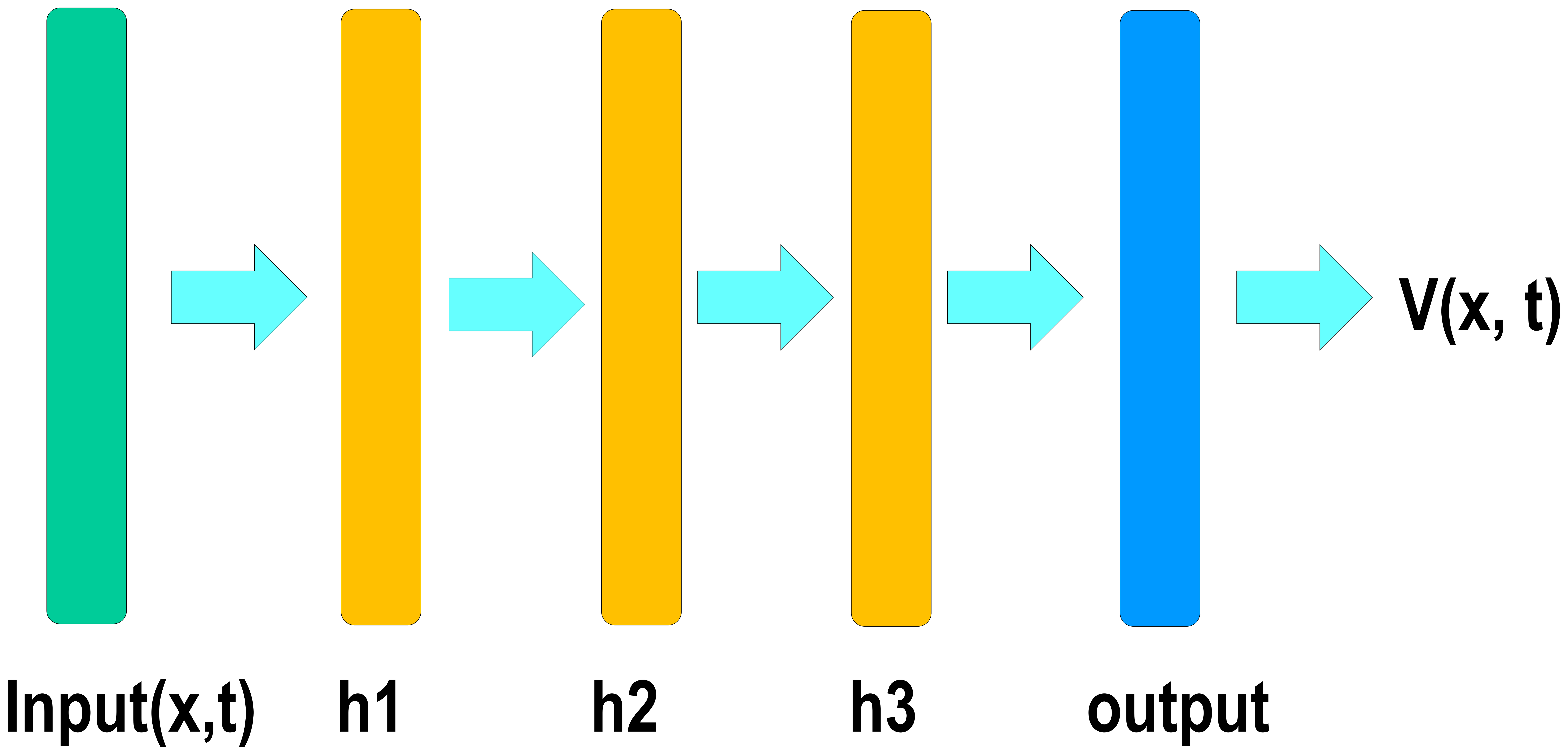
$$\mathcal{R}(T) = \{\mathrm{x}_0 : \exists u, s.t.\ \forall d, x(\cdot)\ \textit{satisfies}\ \dot{x} = f(x,u,d), x(0) = x_0; \exists t \in [0,T], s.t.\ x(t) \in \mathcal{L}\}$$

Project: improve the performance of DeepReach on high-dimensional systems by using different activation functions

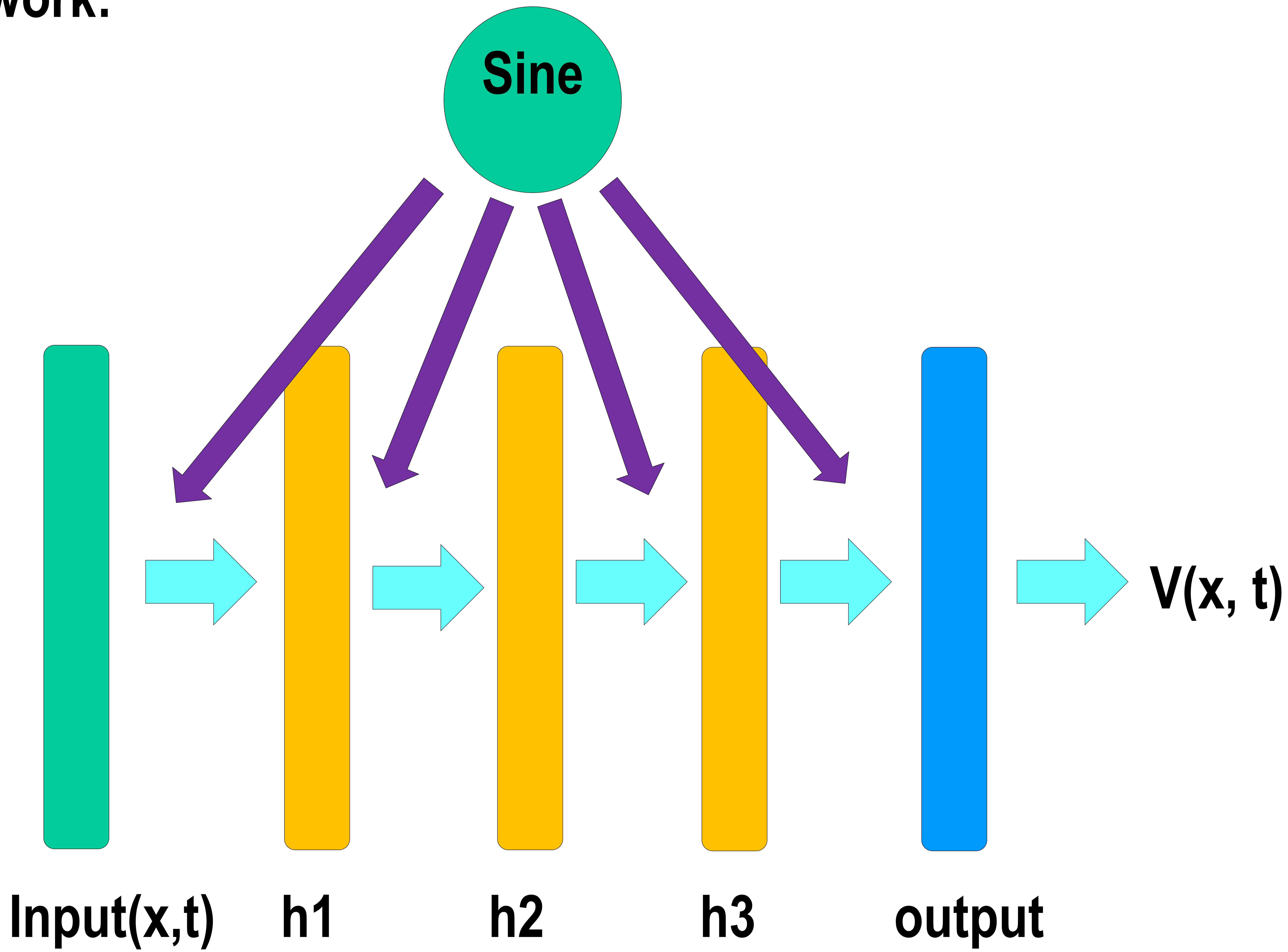
Related work:

DeepReach

- Multi-layer Perceptron with 5 layers
- Function approximation



Related work:



Related work:

Toolboxes to solve HJ Reachability:

- **HelperOC (Matlab)**
- **DeepReach (Python)**
- **BEACLS (C++)**
- **Etc.**

Why DeepReach? 🤔

Performance of DeepReach on High-Dimensional Systems

Related work:

Toolboxes to solve HJ Reachability:

- **HelperOC (Matlab)**
- **DeepReach (Python)**
- **BEACLS (C++)**
- **Etc.**

Why DeepReach? 🤔

Its computation and memory cost don't scale exponentially with the dimensionality of the system, but with the complexity of the value function.

Related work:

Running example: air3D

Framed by the relative dynamics between the two identical vehicles: one pursuer, one evader:

$$\dot{x}_1 = -v_e + v_p cos x_3 + \omega_e x_2$$
$$\dot{x}_2 = v_p sin x_3 - \omega_e x_1$$
$$\dot{x}_3 = \omega_p - \omega_e,$$

Performance of DeepReach on High-Dimensional Systems

Related work:

Running example: air3D

Framed by the relative dynamics between the two identical vehicles: one pursuer, one evader:

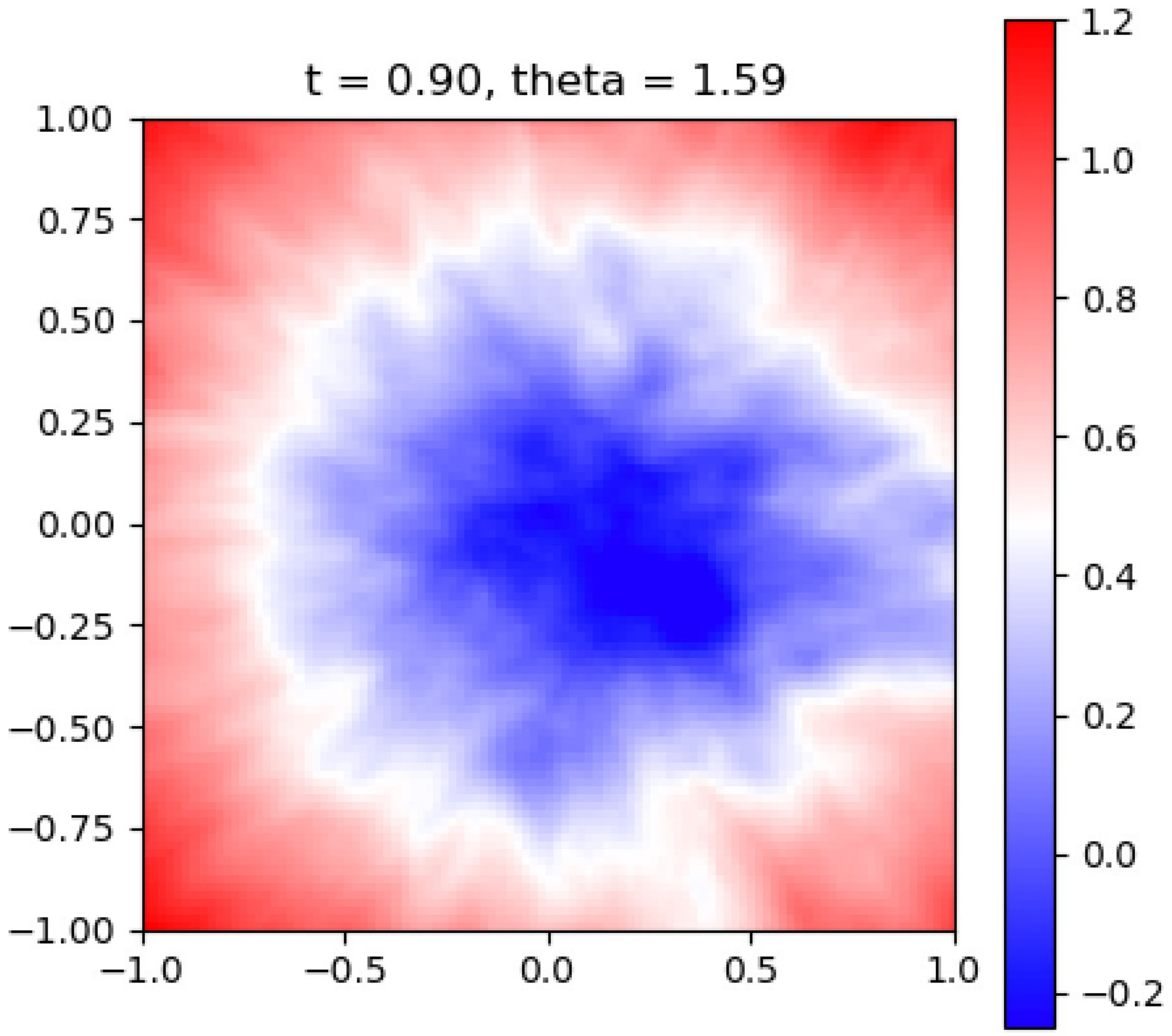
$$\dot{x}_1 = -v_e + v_p cos x_3 + \omega_e x_2$$
$$\dot{x}_2 = v_p sin x_3 - \omega_e x_1$$
$$\dot{x}_3 = \omega_p - \omega_e,$$

Failure set: distance between the two vehicles <= some constant

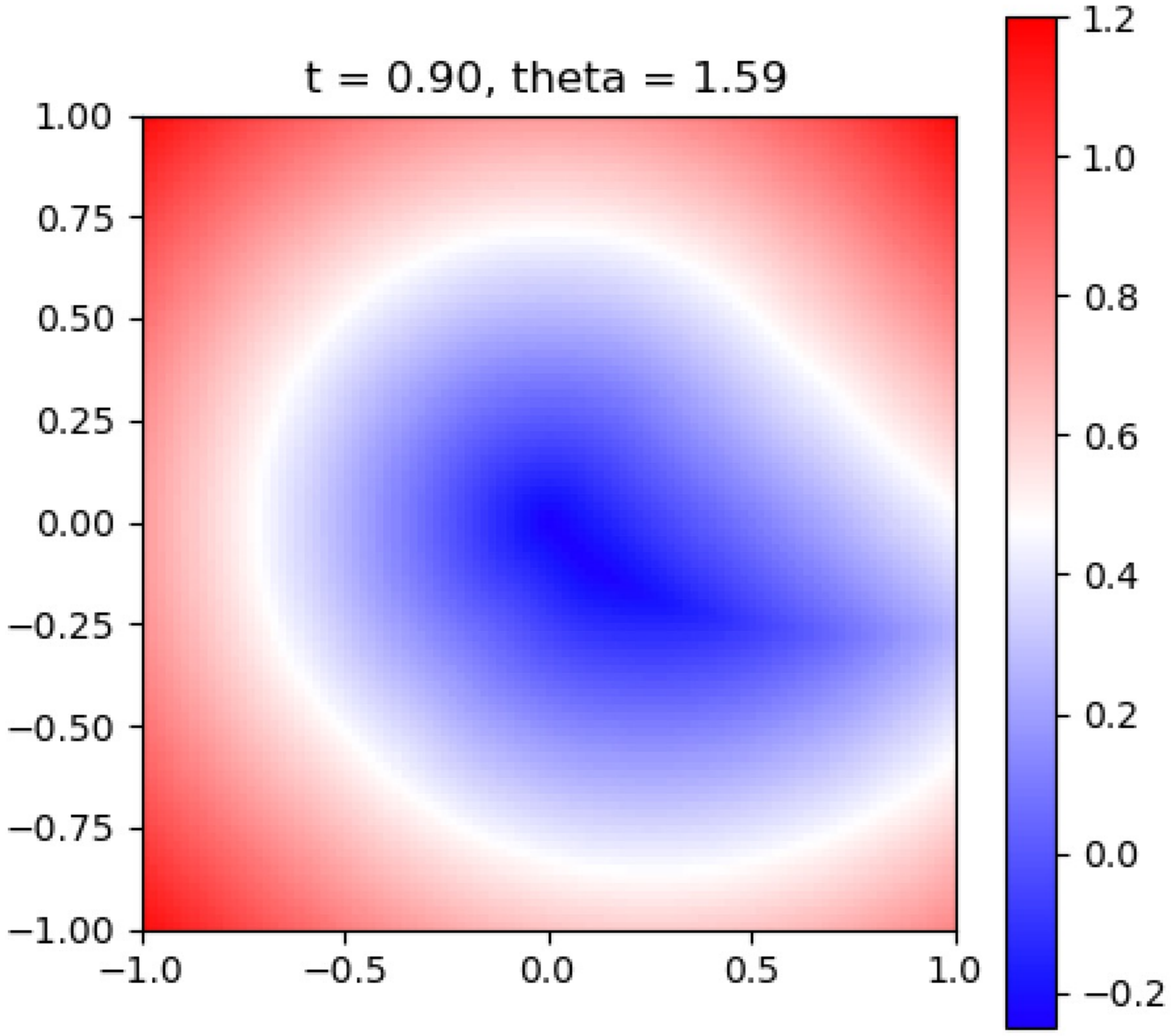
$$\mathcal{L} = \{x : \|(x_1, x_2)\| \leq \beta\}.$$

Related work:

Running example: air3D



ReLU

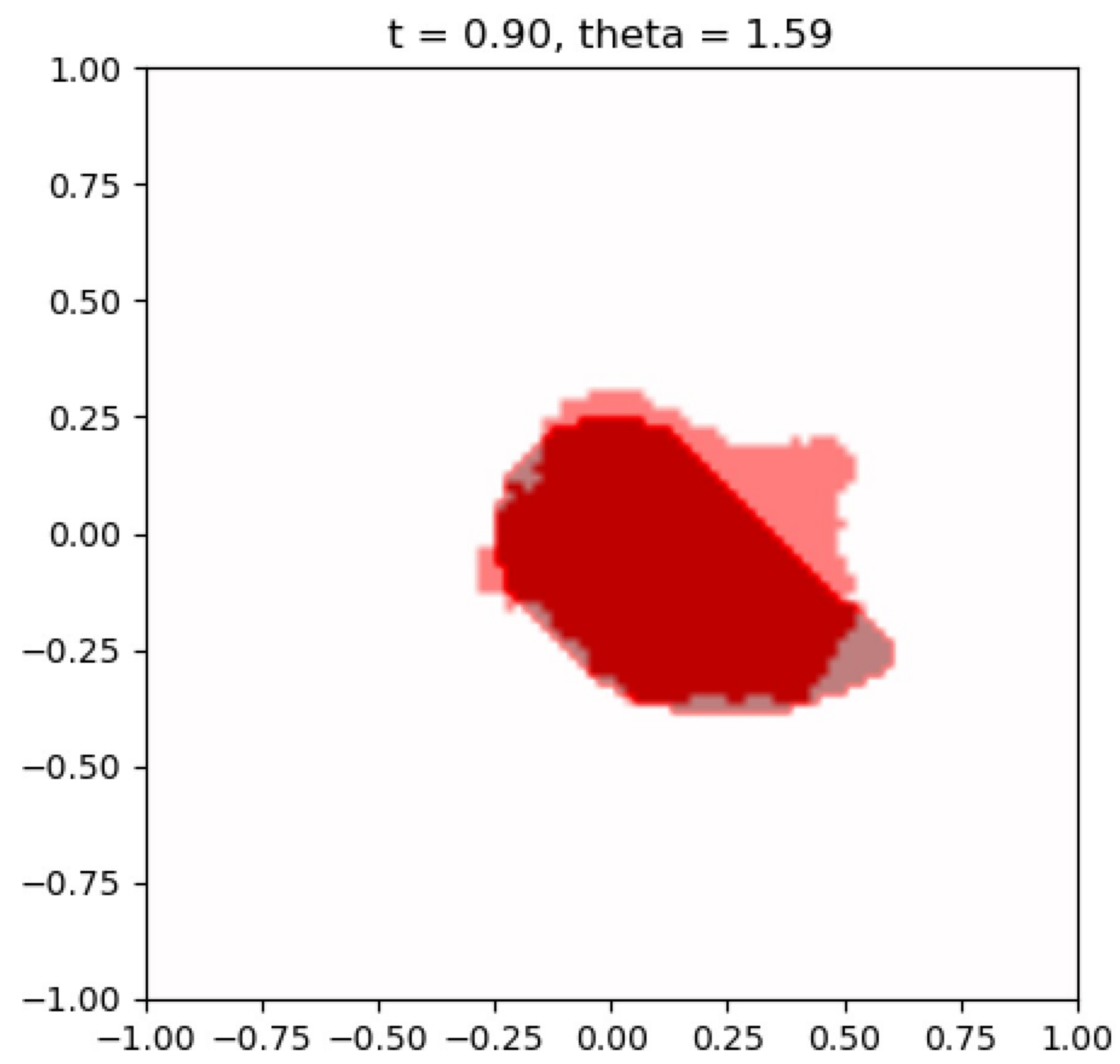


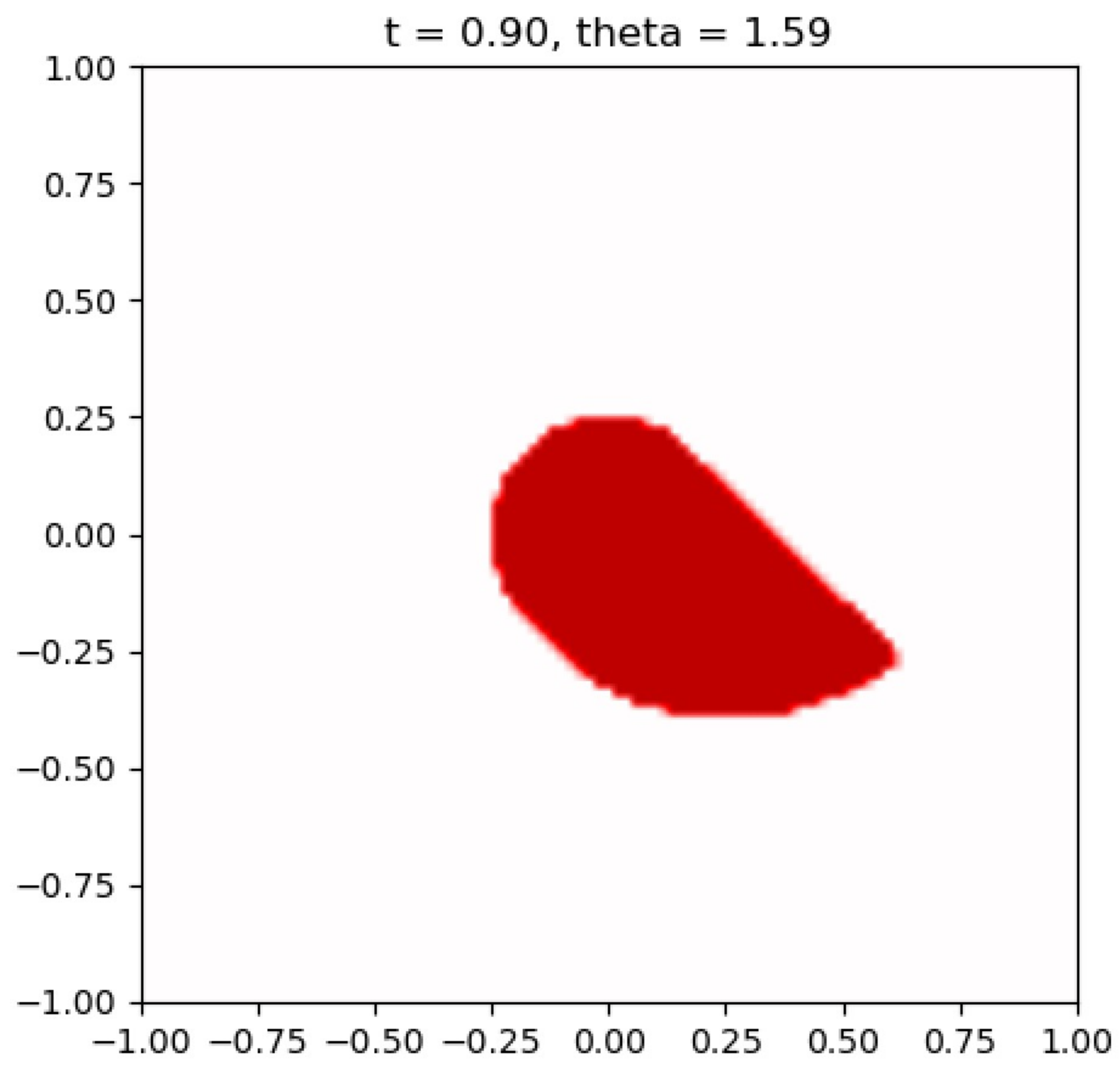
Sine

Performance of DeepReach on High-Dimensional Systems

Related work:

Running example: air3D





ReLU **Sine**

Proposed approach:

- How about using combination of Sine and ReLU?

Intuition:

Sine: better models the gradient, but harder to compute

Performance of DeepReach on High-Dimensional Systems

Proposed approach:

- How about using combination of Sine and ReLU?

Intuition:

Sine: better models the gradient, but harder to compute

ReLU: less accurate gradient, first derivative is piecewise constant, but less computational cost

Proposed approach:

- How about using combination of Sine and ReLU?

Intuition:

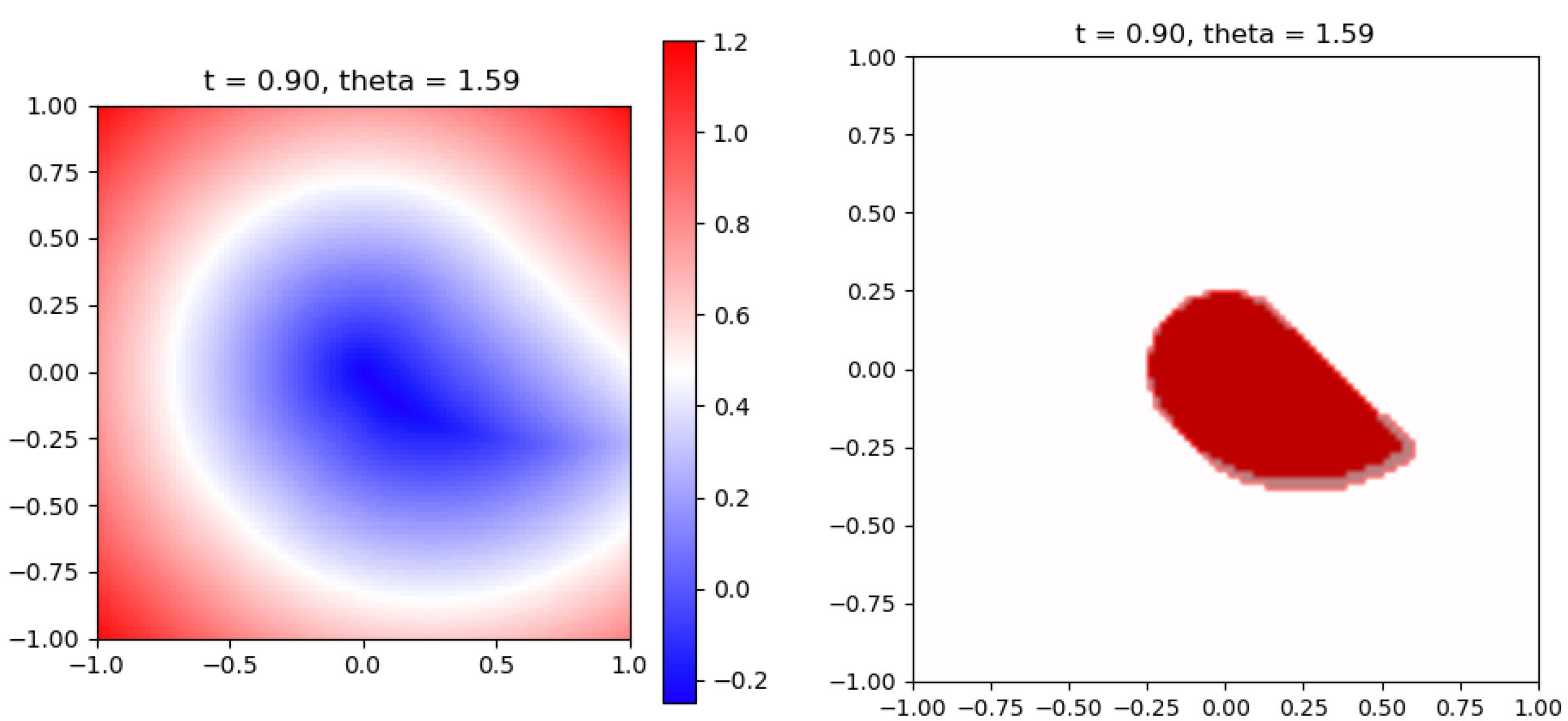
Sine: better models the gradient, but harder to compute

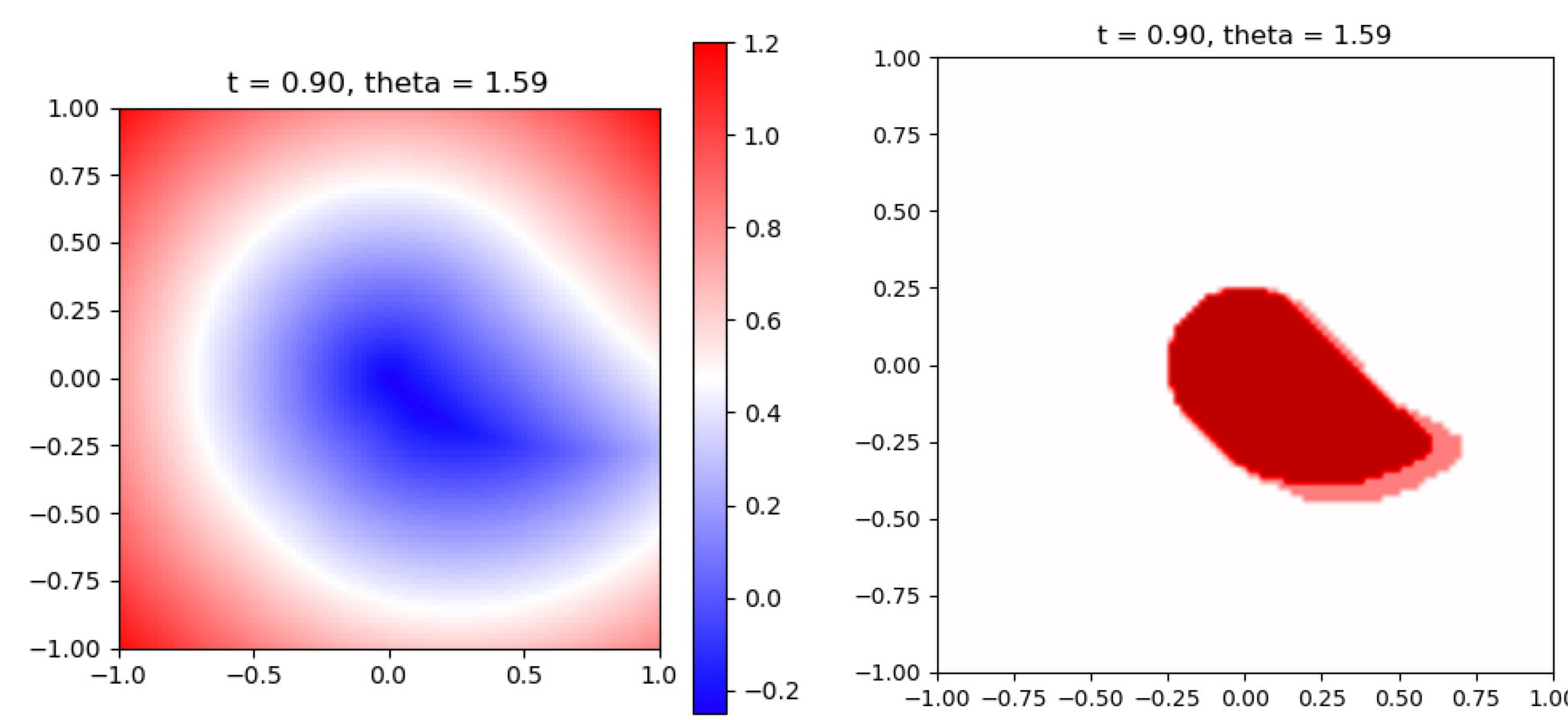
ReLU: less accurate gradient, first derivative is piecewise constant, but less computational cost

max(0, max(0, max(0, x))) vs sin(sin(sin(x)))

Results:

Running example: air3D

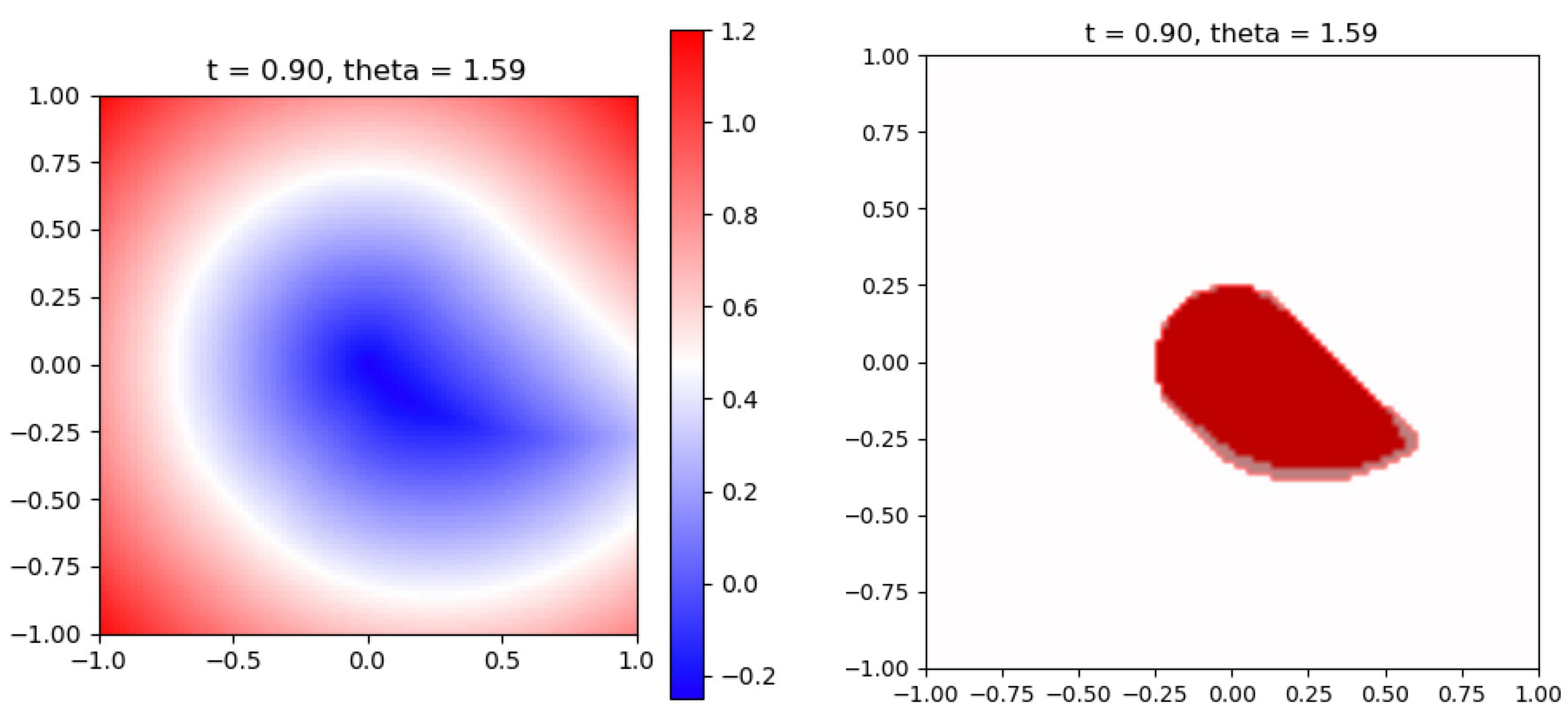




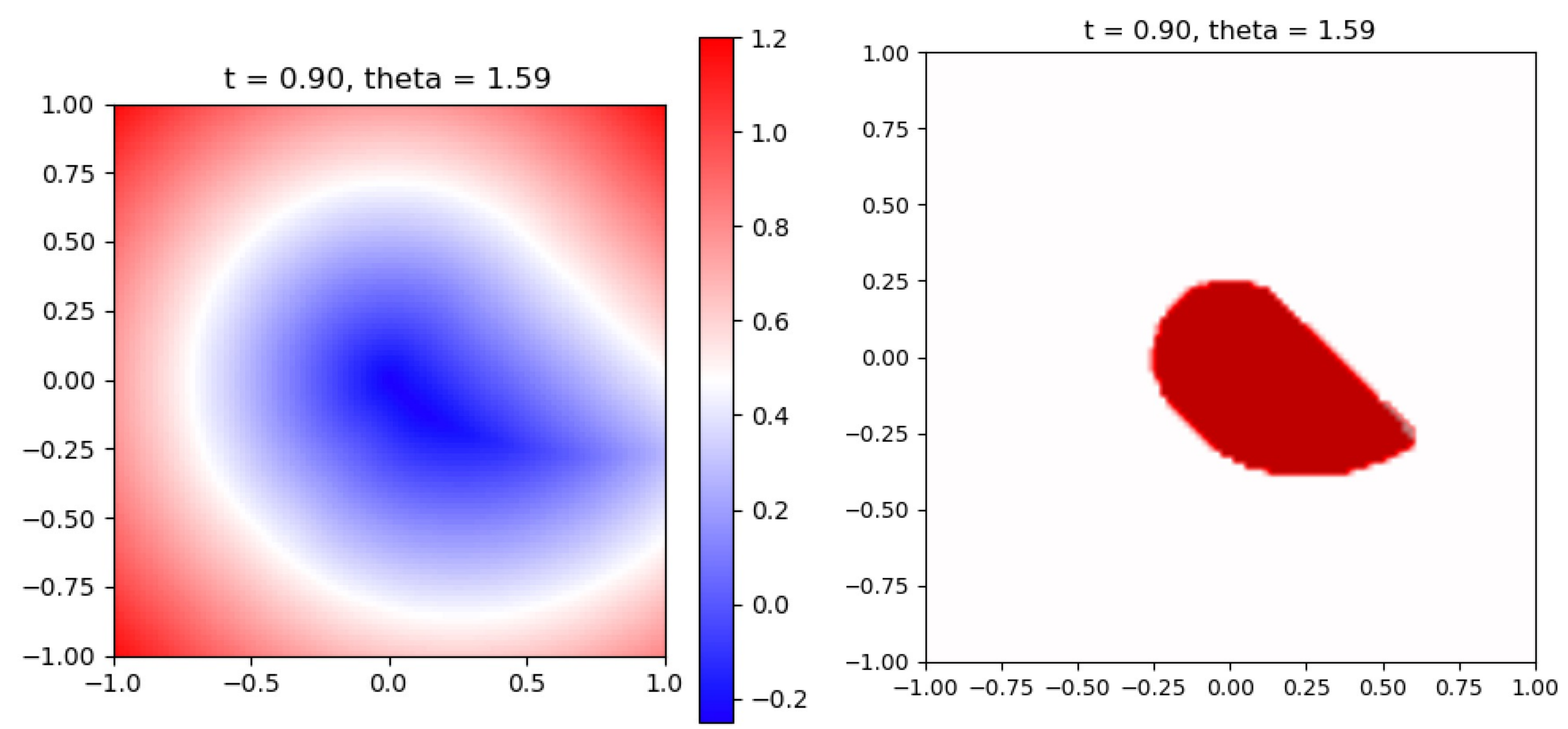
rrrsl **srrrl**

Results:

Running example: air3D



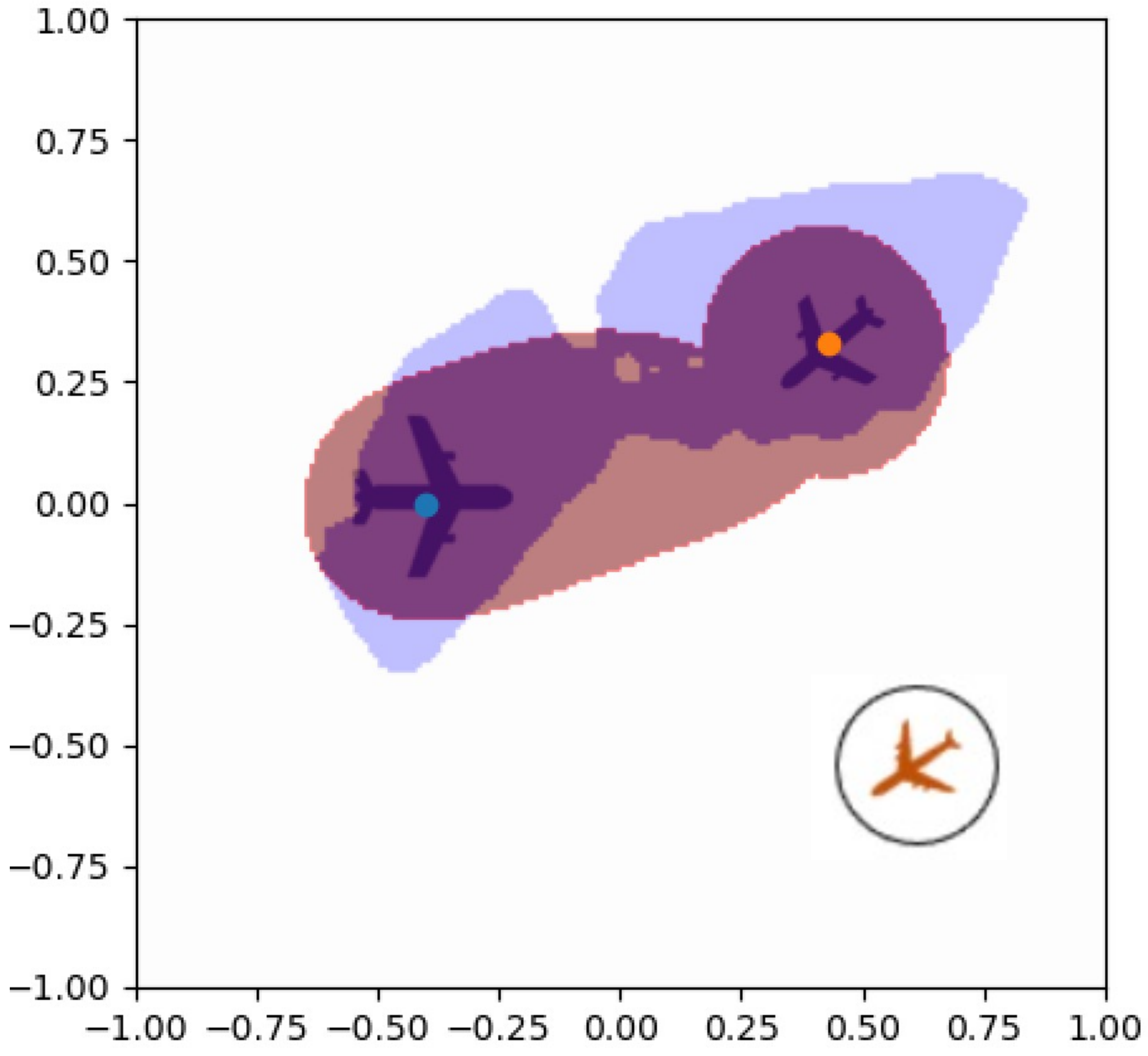
srsrl



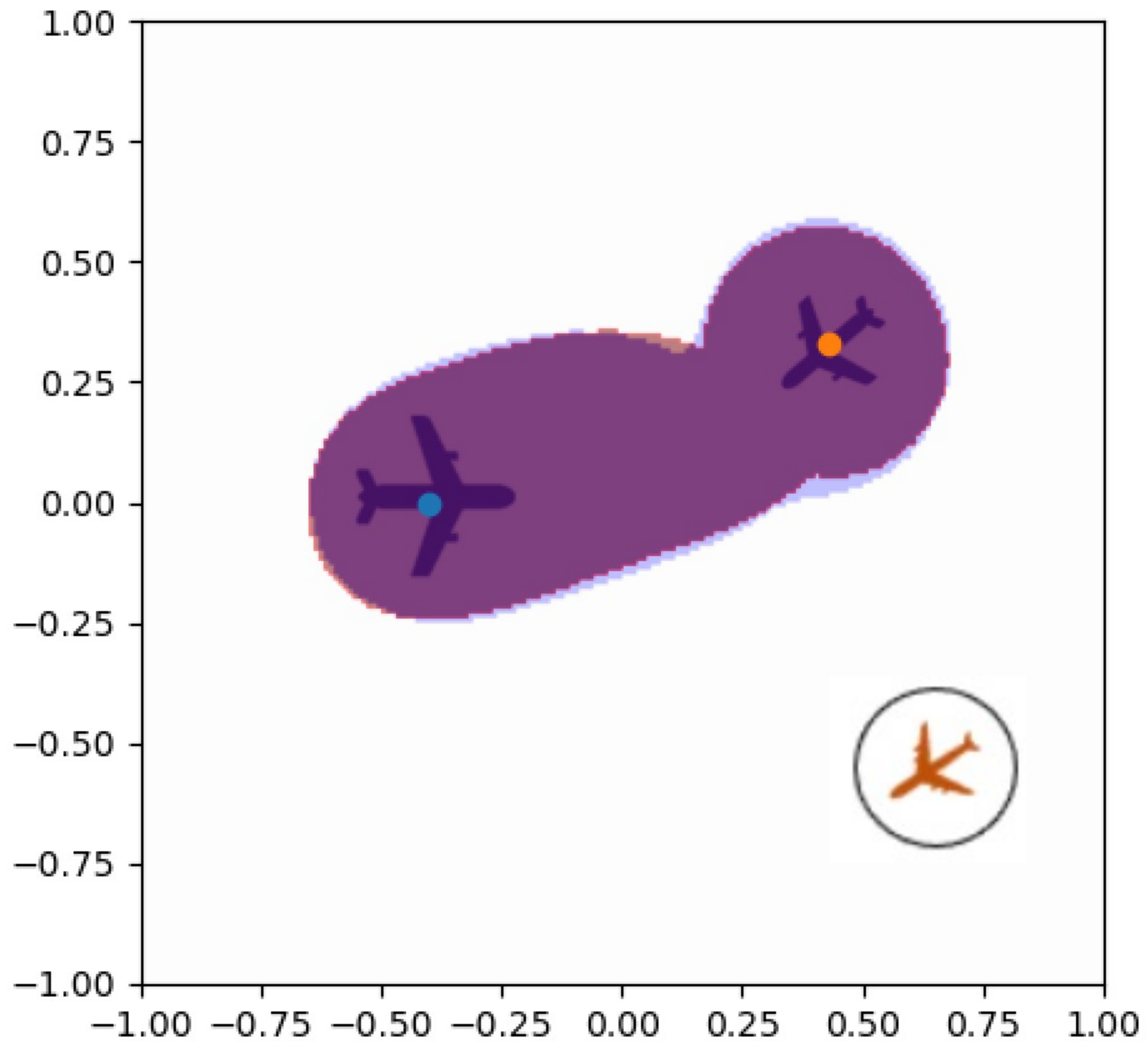
ssrsl

Results:

Running example: 9D three-vehicle collision (2Evaders, 1Pursuer)



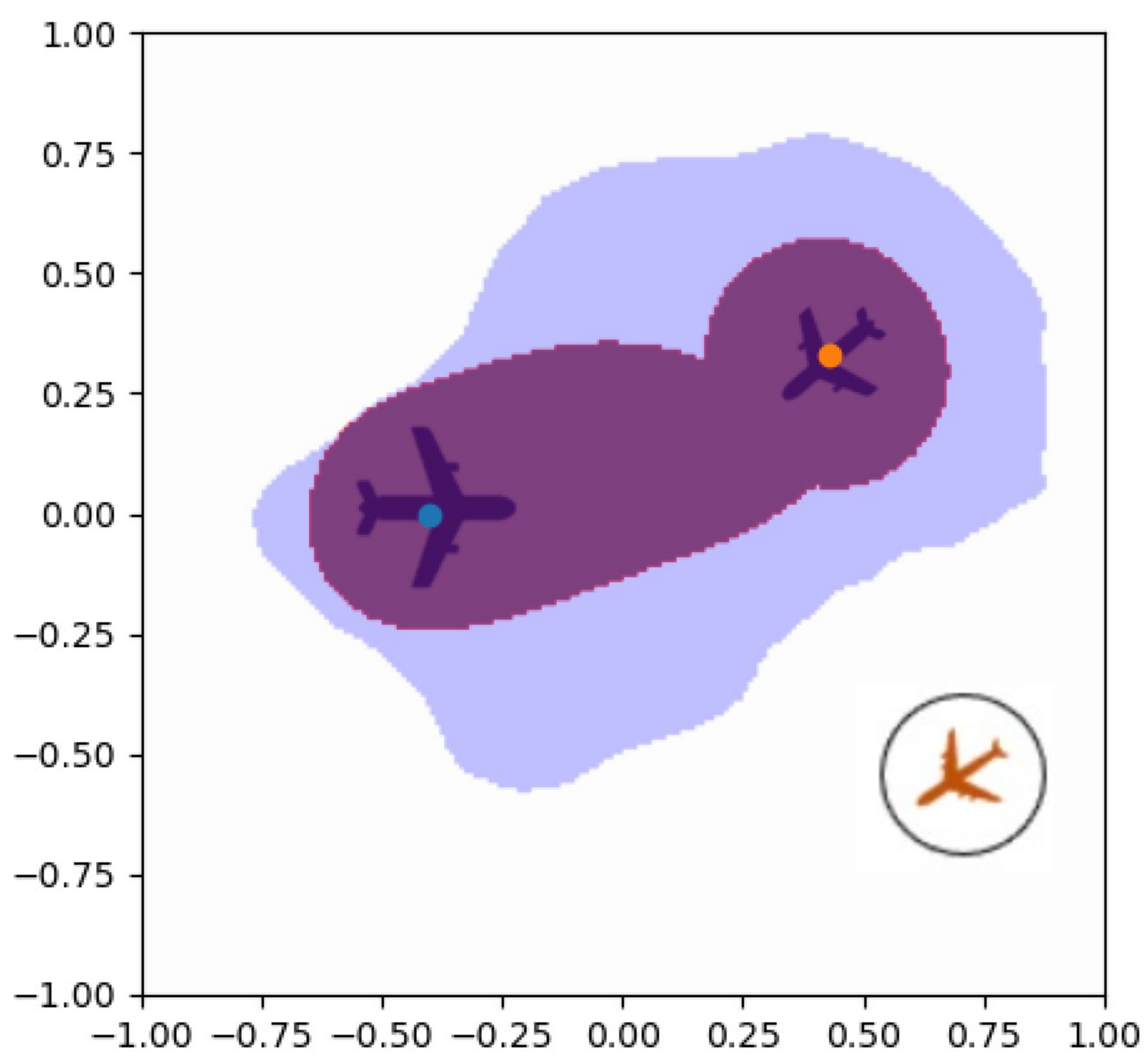
rrrrl



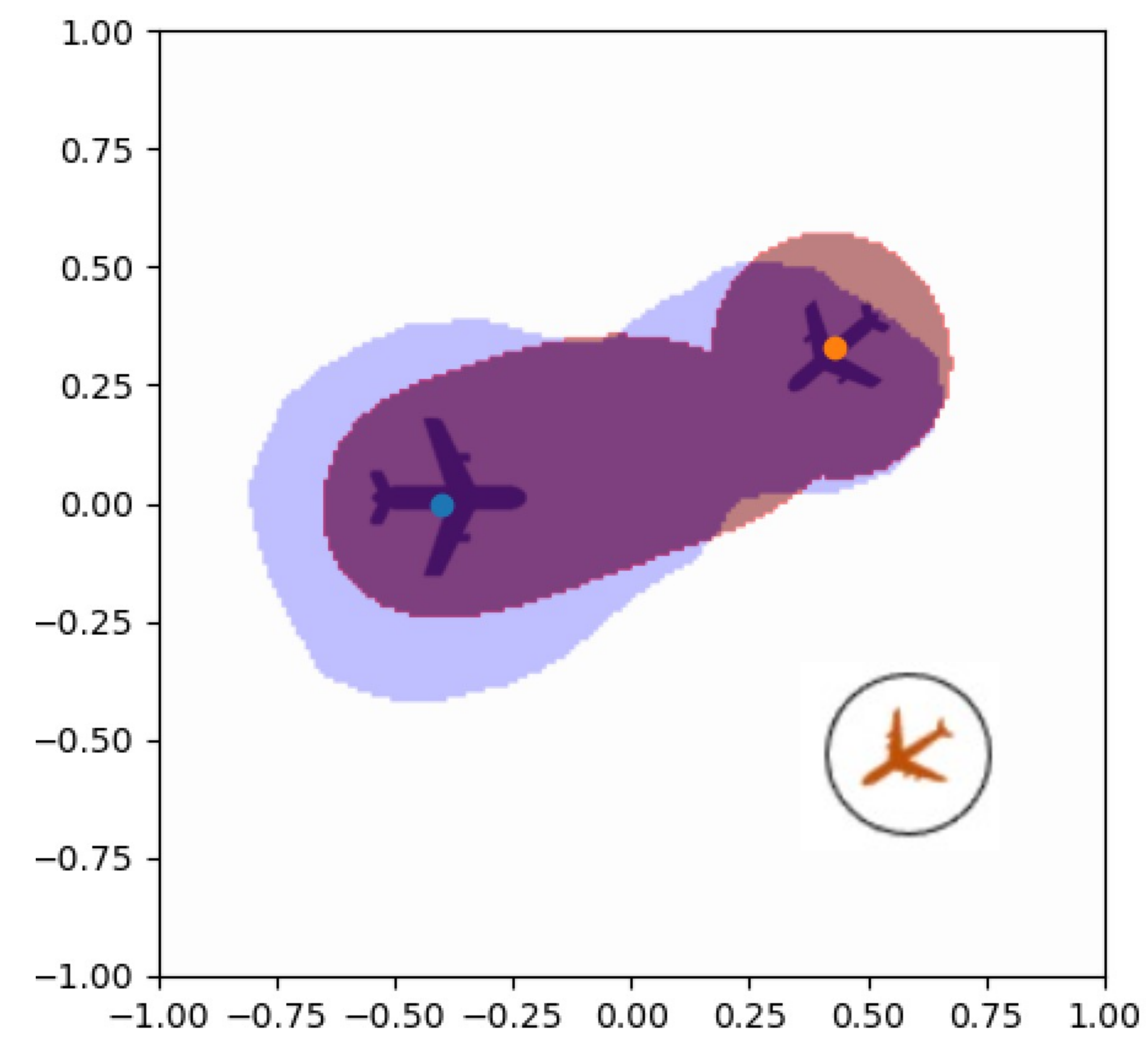
ssssl

Results:

Running example: 9D three-vehicle collision (2Evaders, 1Pursuer)



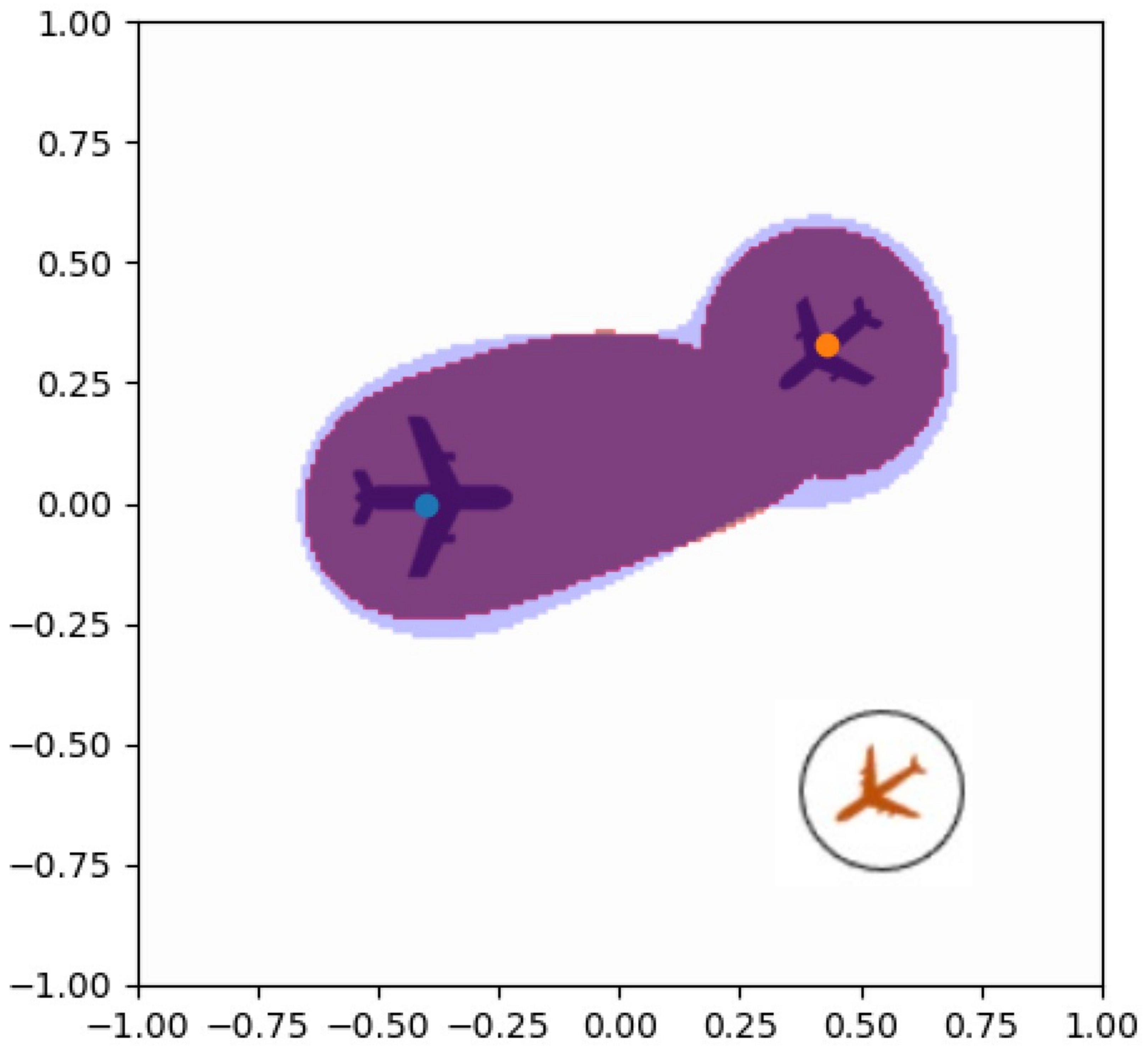
rrrsl



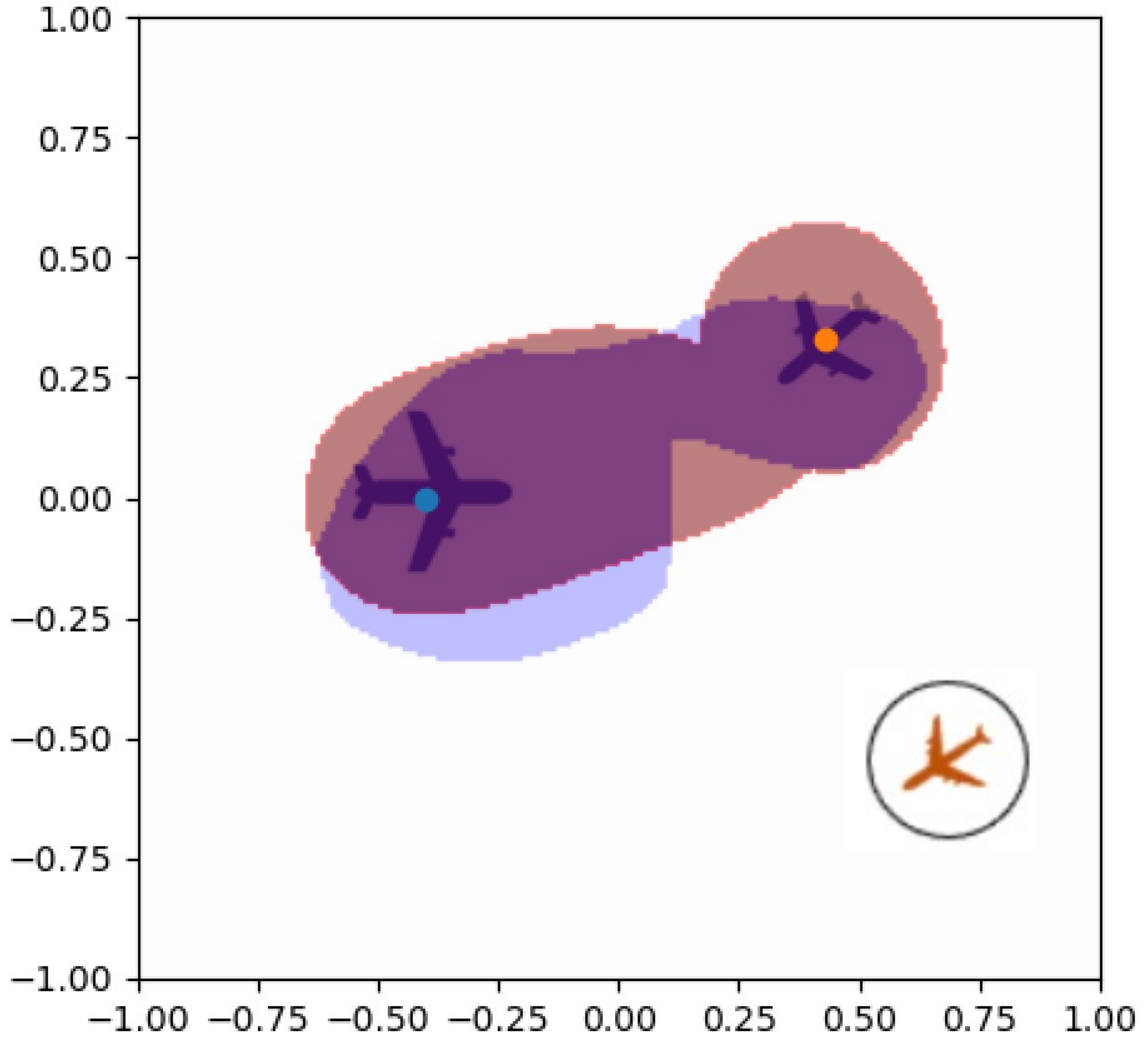
srrrl

Results:

Running example: 9D three-vehicle collision (2Evaders, 1Pursuer)



ssrsl



rsrsl

Performance of DeepReach on High-Dimensional Systems

Analysis:

- The graphs provide some visual information about the BRTs, but it relies on our intuition and is not constructive.

Performance of DeepReach on High-Dimensional Systems

Analysis:

- The graphs provide some visual information about the BRTs, but it relies on our intuition and is not constructive.

- Question: how to quantify the accuracy of BRTs?

Useful tools from the paper below:

A. Lin and S. Bansal, "Generating Formal Safety Assurances for High-Dimensional Reachability,"

Scenario Optimization – violation rate:

1. Sample a state $x \in X$ uniformly randomly and apply $u^*$and $\boldsymbol{d}^*$ to generate trajectory $\xi_{\boldsymbol{x},\boldsymbol{t}}^{\boldsymbol{u},\boldsymbol{d}}(\boldsymbol{s})$ for s $\in [0, T_f]$
2. If it meets any of the following two conditions, we mark it as a violation

1)

$$\exists x \in BRT^C \wedge s \in [0, T_f] : \xi_{x,t}^{u,d}(s) \in T$$

2)

$$\exists x \in BRT : \forall s \in [0, T_f], \xi_{x,t}^{u,d}(s) \notin T$$

Scenario Optimization – δ-level:

- the maximum learned value of an empirically unsafe initial state under the induced policy $\tilde{\pi}$

$$\delta_{\tilde{V},\tilde{\pi}} := \max_{x \in X} \left\{ \tilde{V}(x,0) : J_{\tilde{\pi}}(x,0) \leq 0 \right\}$$

- Intuitively, the violation rate measures how often the model deviates from the true BRT, and δ-level measures how far away the worst violating state is from the true BRT.

Results:

- We want to compute the violation rate the BRTs we obtained from the experiments

Activation	Violation Rate	δ-level
ssrsl	0.189	0.441
ssssl	0.190	0.550
rsrsl	0.211	0.891
rrrsl	0.213	0.803
rrrrl	0.235	0.999
srrrl	0.271	0.916

Performance of DeepReach on High-Dimensional Systems

Limitations:

- For various systems of different dimensions, we might get different empirical results. Therefore, no naive conclusions can be drawn.

Limitations:

- For various systems of different dimensions, we might get different empirical results. Therefore, no naive conclusions can be drawn.

- Only some combinations of Sine and ReLU are experimented with, while others remain unknown. Need a more systematic way of examining the influence of activation on the accuracy of BRTs.

Summary:

1. We explored several combinations of Sine and ReLU in DeepReach on 3D and 9D systems.

Performance of DeepReach on High-Dimensional Systems

Summary:

1. We explored several combinations of Sine and ReLU in DeepReach on 3D and 9D systems.

2. Based on the plots and violation rates, adding layers of sine activation helps learn more accurate value functions, thus better BRTs. However, the results cannot be generalized to other systems.

Future work/directions:

1. Explore other hyperparameters and architecture of DeepReach

Future work/directions:

1. Explore other hyperparameters and architecture of DeepReach

2. Implement error correction to reduce the error rates of BRTs obtained from DeepReach