Learning-based Analysis and Control of Safety-Critical Systems

Enrique Mallada

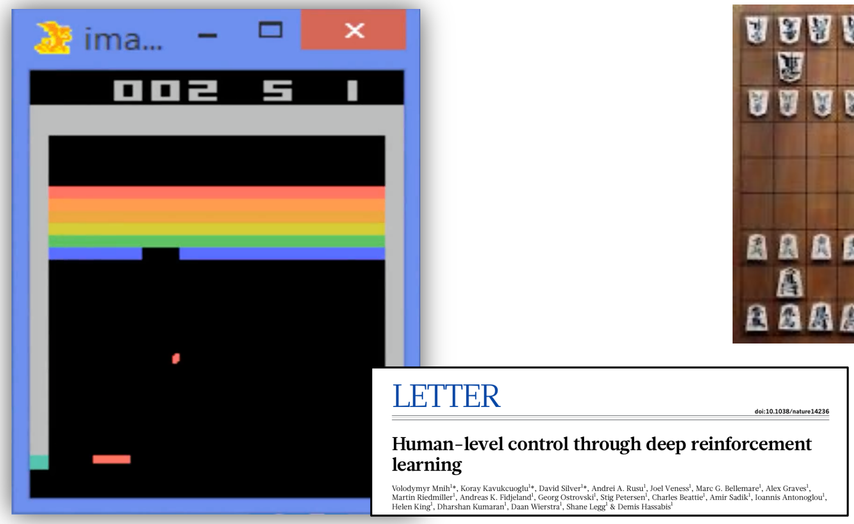
JOHNS HOPKINS UNIVERSITY

University of California San Diego

May 26, 2022

A World of Success Stories

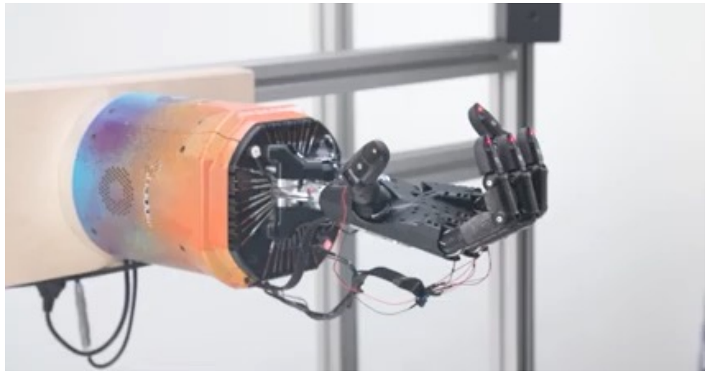
2017 Google DeepMind's DQN



2017 AlphaZero – Chess, Shogi, Go

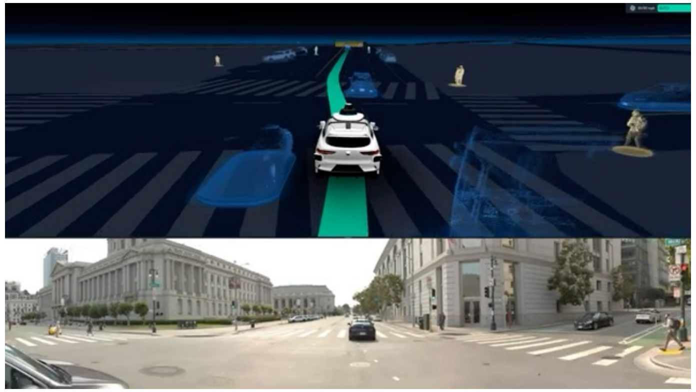
2019 AlphaStar – Starcraft II

OpenAI – Rubik's Cube



Boston Dynamics

Waymo



Reality Kicks In

Angry Residents, Abrupt Stops: Waymo Vehicles Are Still Causing Problems in Arizona

RAY STERN | MARCH 31, 2021 | 8:26AM

GARY MARCUS BUSINESS 08.14.2019 09:00 AM

DeepMind's Losses and the Future of Artificial Intelligence

Alphabet's DeepMind unit, conqueror of Go and other games, is losing lots of money. Continued deficits could imperil investments in AI.

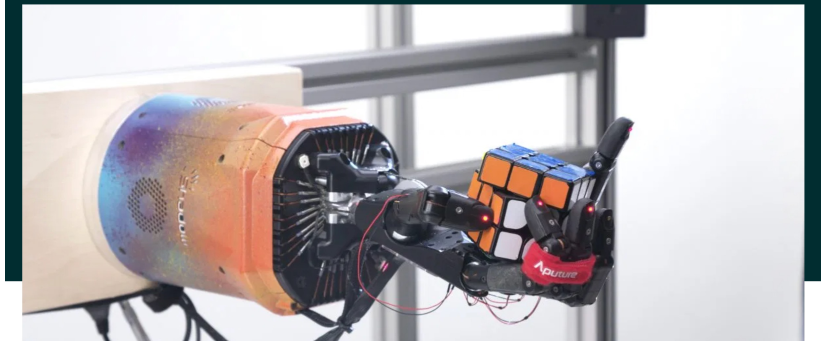
AARIAN MARSHALL BUSINESS 12.07.2020 04:06 PM

Uber Gives Up on the Self-Driving Dream

The ride-hail giant invested more than $1 billion in autonomous vehicles. Now it's selling the unit to Aurora, which makes self-driving tech.

OpenAI disbands its robotics research team

Kyle Wiggers @Kyle_L_Wiggers July 16, 2021 11:24 AM

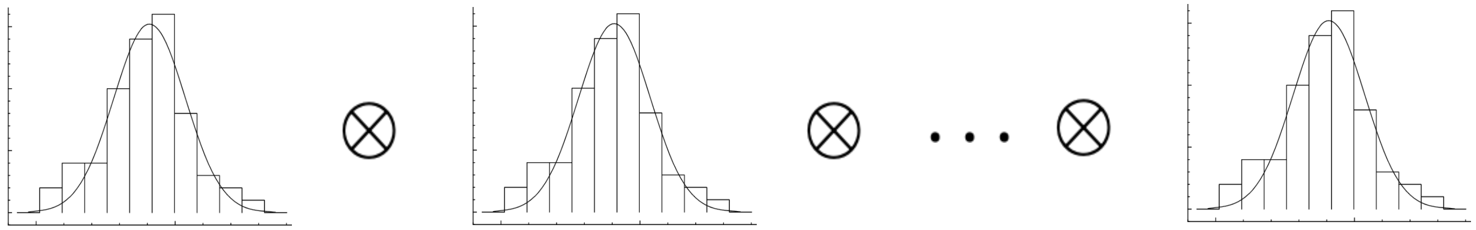


Self-driving Uber car that hit and killed woman did not recognize that pedestrians jaywalk

The automated car lacked "the capability to classify an object as a pedestrian unless that object was near a crosswalk," an NTSB report said.

Core challenge: The curse of dimensionality

- **Sampling in d dimension with resolution ϵ**



Sample complexity:

$$O(\varepsilon^{-d})$$

For $\epsilon = 0.1$ and $d = 100$, we would need 10^{100} points.

- **Verifying non-negativity of polynomials**

Copositive matrices:

$$[x_1^2 \ldots x_d^2] A [x_1^2 \ldots x_d^2]^{\mathrm{T}} \geq 0$$

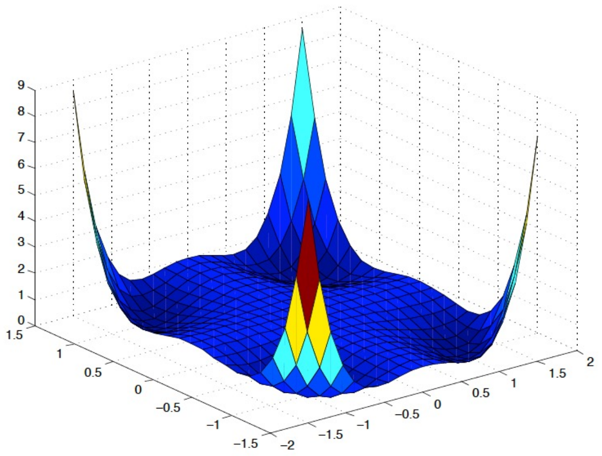
Murty&Kadabi [1987]: Testing co-positivity is NP-Hard

Sum of Squares (SoS):

$$z(x)^T Q z(x) \geq 0, \quad z_i(x) \in \mathbb{R}[x], \ x \in \mathbb{R}^d, Q \succcurlyeq 0$$

Artin [1927] (Hilbert's 17th problem):

Non-negative polynomials are sum of square of *rational* functions



Motzkin [1967]:
$p = x^4y^2 + x^2y^4 + 1 - 3x^2y^2$
is nonnegative,
not a sum of squares,
but $(x^2 + y^2)^2 p$ is SoS

Question: Are we asking too much?

- Learnability requires uniform approximation errors across the ***entire domain***

  Q: Can we provide local guarantees, and progressively expand as needed?

[arXiv ‘22] Shen, Bichuch, M

- Lyapunov functions and control barrier functions require strict and exhaustive notions of ***invariance***

  Q: Can we substitute invariance with less restrictive properties?

[arXiv ‘22] Shen, Bichuch, M

- Control synthesis usually aims for the ***best*** (optimal) controller

  Q: Can we focus on feasibility, rather than optimality?

[arXiv ’21, L4DC 22] Castellano, Min, Bazerque, M

[arXiv 22] Shen, Bichuch, M, *Model-free Learning of Regions of Attraction via Recurrent Sets,* **submitted to CDC 2022,** preprint arXiv:2204.10372.
[L4DC 22] Castellano, Min, Bazerque, M, *Reinforcement Learning with Almost Sure Constraints,* **Learning for Dynamics and Control (L4DC) Conference, 2022**
[arXiv 21] Castellano, Min, Bazerque, M, *Learning to Act Safely with Limited Exposure and Almost Sure Certainty,* **submitted to IEEE TAC, 2021, under review**, preprint arXiv:2105.08748

Question: Are we asking too much?

- Learnability requires uniform approximation errors across the ***entire domain***

  Q: Can we provide local guarantees, and progressively expand as needed?

[arXiv ‘22] Shen, Bichuch, M

- Lyapunov functions and control barrier functions require strict and exhaustive notions of ***invariance***

  Q: Can we substitute invariance with less restrictive properties?

[arXiv ‘22] Shen, Bichuch, M

- Control synthesis usually aims for the ***best*** (optimal) controller

  Q: Can we focus on feasibility, rather than optimality?

[arXiv ’21, L4DC 22] Castellano, Min, Bazerque, M

[arXiv 22] Shen, Bichuch, M, *Model-free Learning of Regions of Attraction via Recurrent Sets,* **submitted to CDC 2022,** preprint arXiv:2204.10372.
[L4DC 22] Castellano, Min, Bazerque, M, *Reinforcement Learning with Almost Sure Constraints,* **Learning for Dynamics and Control (L4DC) Conference, 2022**
[arXiv 21] Castellano, Min, Bazerque, M, *Learning to Act Safely with Limited Exposure and Almost Sure Certainty,* **submitted to IEEE TAC, 2021, under review**, preprint arXiv:2105.08748

[Submitted on 21 Apr 2022]

Model-free Learning of Regions of Attraction via Recurrent Sets

Yue Shen, Maxim Bichuch, Enrique Mallada

arXiv > cs > arXiv:2204.10372

Yue Shen

JOHNS HOPKINS UNIVERSITY

Maxim Bichuch

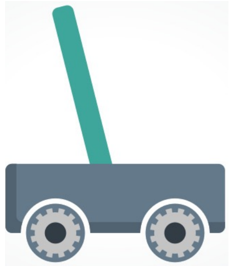
JOHNS HOPKINS UNIVERSITY

Motivation: Estimation of regions of attraction

Having an approximation of the region of attraction allows us to

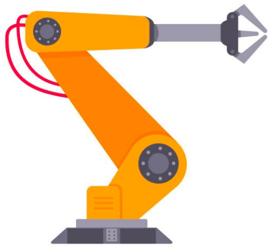
- **Test the limits of controller designs**

  especially for those based on (possibly linear) approximations of nonlinear systems



cart-pole

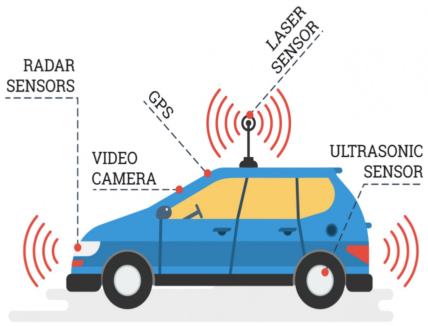
quadcopter



robot arm

. . .

- **Verify safety of certain operating condition**



self-driving

HVAC system

power grids

. . .

Problem setup

Continuous time dynamical system: $\dot{x}(t) = f(x(t))$

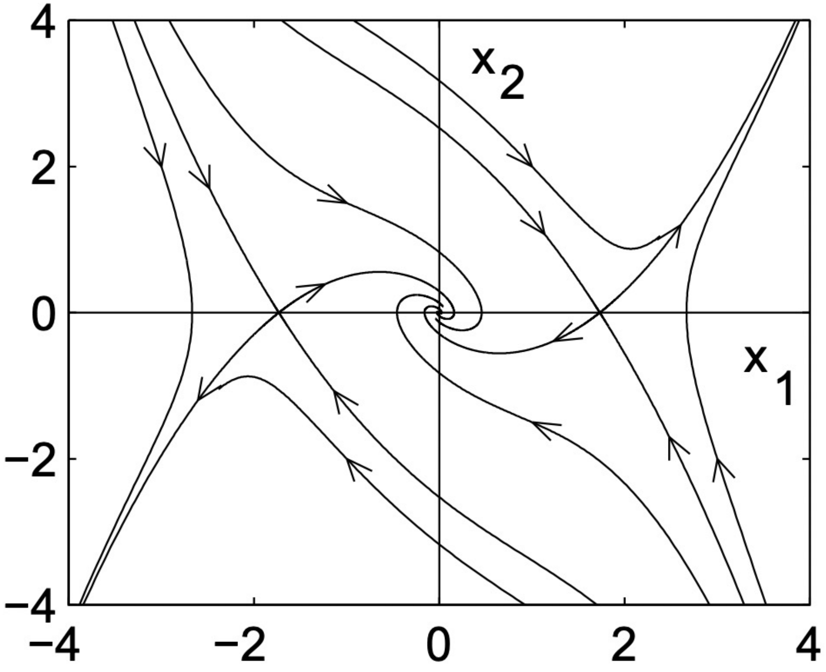
- Initial condition $x_0 = x(0)$, solution at time t: $\phi(t, x_0)$.
- The ω-limit set of the system: $\Omega(f)$

Region of attraction (ROA) of a set $S \subseteq \Omega(f)$:

$$\mathcal{A}(S) := \left\{ x_0 \in \mathbb{R}^d | \lim_{t \to \infty} \phi(t, x_0) \in S \right\}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -x_1 + \frac{1}{3}x_1^3 - x_2 \end{bmatrix}$$

$$\Omega(f) = \{(0,0), (-\sqrt{3}, 0), (\sqrt{3}, 0)\}$$



Problem setup

Continuous time dynamical system: $\dot{x}(t) = f(x(t))$

- Initial condition $x_0 = x(0)$, solution at time t: $\phi(t, x_0)$.
- The ω-limit set of the system: $\Omega(f)$

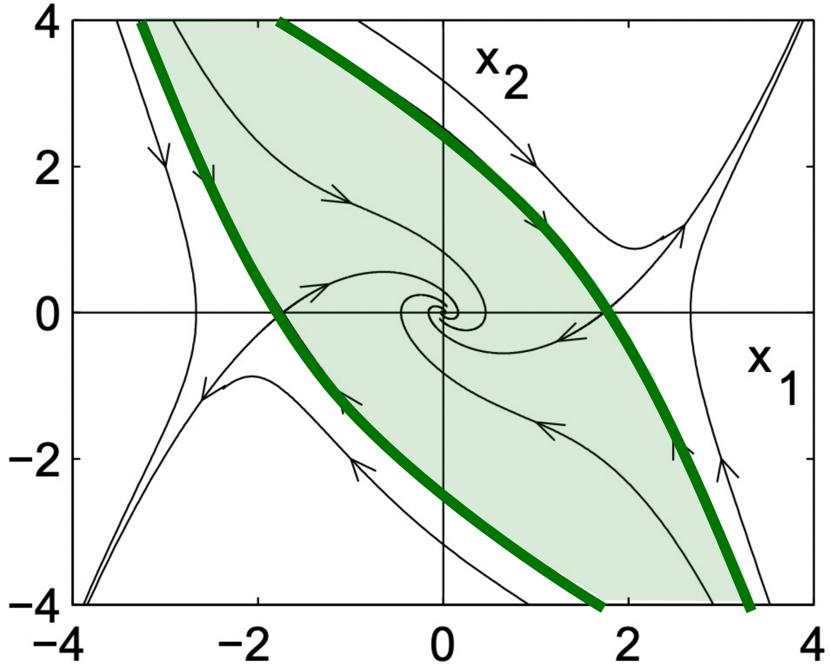
Region of attraction (ROA) of a set $S \subseteq \Omega(f)$:

$$\mathcal{A}(S) := \left\{ x_0 \in \mathbb{R}^d \middle| \lim_{t\to\infty} \phi(t, x_0) \in S \right\}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -x_1 + \frac{1}{3}x_1^3 - x_2 \end{bmatrix}$$

$$\Omega(f) = \{(0,0), (-\sqrt{3}, 0), (\sqrt{3}, 0)\}$$

$\mathcal{A}(\Omega(f))$



Problem setup

Continuous time dynamical system: $\dot{x}(t) = f(x(t))$

- Initial condition $x_0 = x(0)$, solution at time t: $\phi(t, x_0)$.
- The ω-limit set of the system: $\Omega(f)$

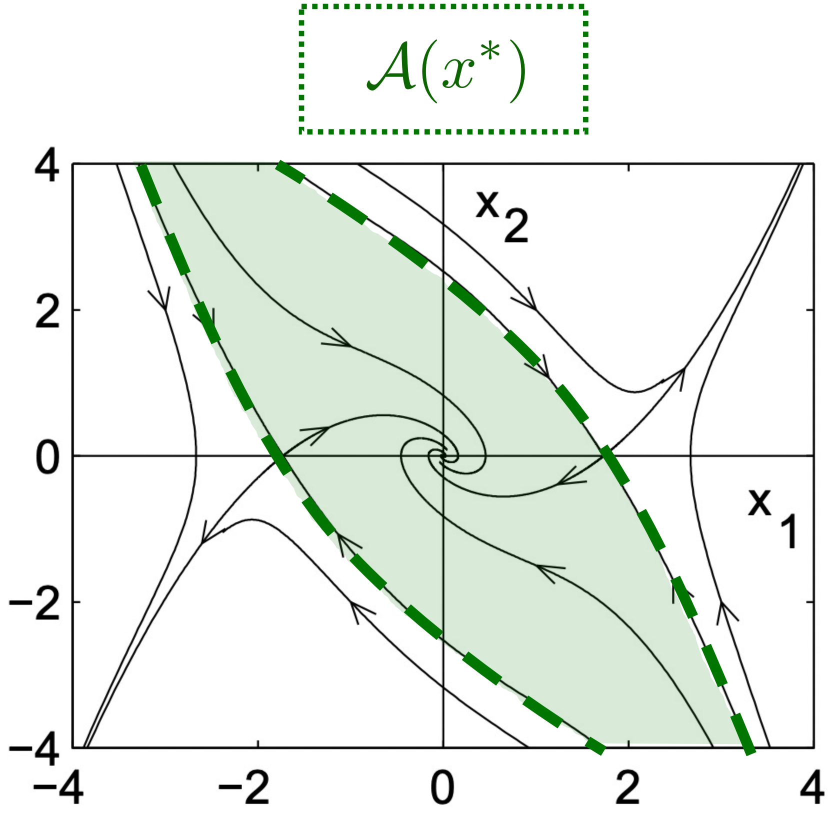
Region of attraction (ROA) of a set $S \subseteq \Omega(f)$:

$$\mathcal{A}(S) := \left\{ x_0 \in \mathbb{R}^d \middle| \lim_{t\to\infty} \phi(t, x_0) \in S \right\}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -x_1 + \frac{1}{3}x_1^3 - x_2 \end{bmatrix}$$

$$\Omega(f) = \{(0,0), (-\sqrt{3}, 0), (\sqrt{3}, 0)\}$$

Asymptotically stable equilibrium at $x^* = (0,0)$



Problem setup

Continuous time dynamical system: $\dot{x}(t) = f(x(t))$

- Initial condition $x_0 = x(0)$, solution at time t: $\phi(t, x_0)$.
- The ω-limit set of the system: $\Omega(f)$

Region of attraction (ROA) of a set $S \subseteq \Omega(f)$:

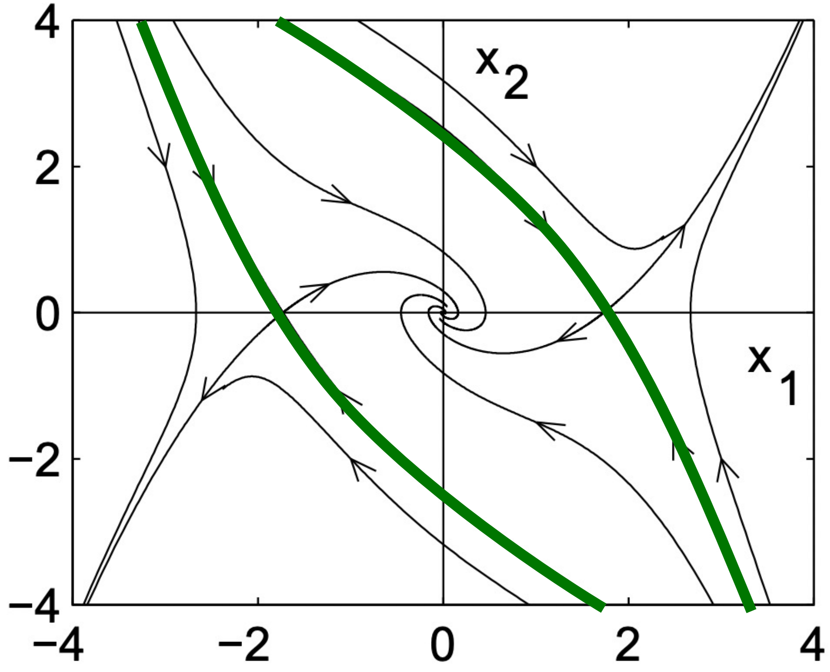
$$\mathcal{A}(S) := \left\{ x_0 \in \mathbb{R}^d \middle| \lim_{t \to \infty} \phi(t, x_0) \in S \right\}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -x_1 + \frac{1}{3} x_1^3 - x_2 \end{bmatrix}$$

$$\Omega(f) = \{(0,0), (-\sqrt{3}, 0), (\sqrt{3}, 0)\}$$

Unstable equilibria $\{(\sqrt{3}, 0), (-\sqrt{3}, 0)\}$

$\mathcal{A}(\{(\sqrt{3}, 0), (-\sqrt{3}, 0)\})$



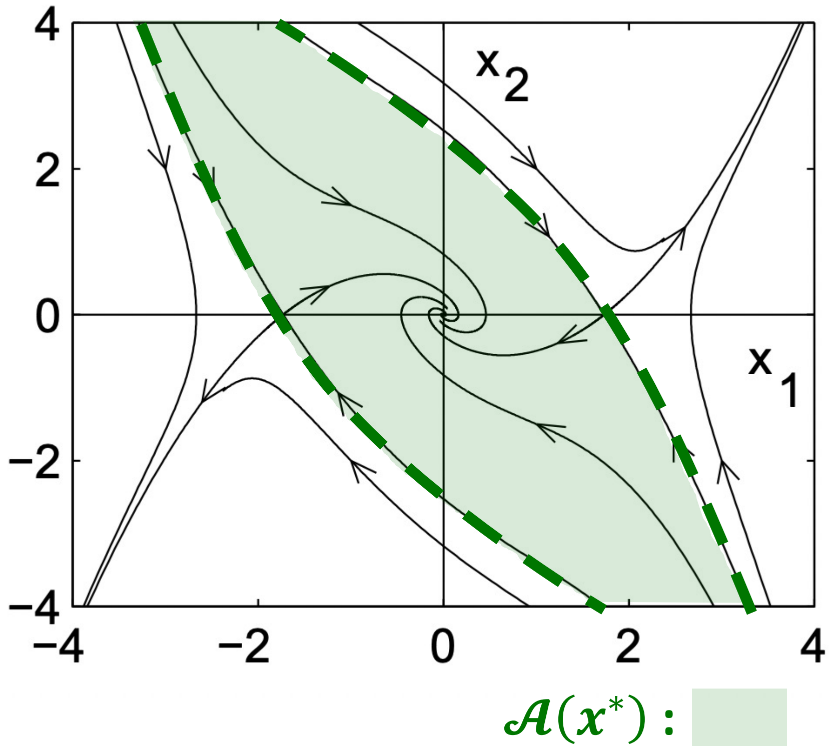
Region of attraction of stable equilibria

Region of attraction (ROA) of a set $S \subseteq \Omega(f)$:

$$\mathcal{A}(S) := \left\{ x_0 \in \mathbb{R}^d \middle| \lim_{t\to\infty} \phi(t, x_0) \in S \right\}$$

Assumption *1. The system $\dot{x}(t) = f(x(t))$ has an asymptotically stable equilibrium at x^*.*

Remark *1. It follows from Assumption 1 that the* ***positively invariant*** *ROA $\mathcal{A}(x^*)$* ***is an open contractible set*** *[Sontag, 2013], i.e., the identity map of $\mathcal{A}(x^*)$ to itself is null-homotopic [Munkres, 2000].*



E. Sontag. "Mathematical Control Theory: Deterministic Finite Dimensional Systems." Springer 2013

J. R. Munkres. "Topology." Prentice Hall 2000

Invariant sets

A set $I \subseteq \mathbb{R}^d$ is **positively invariant** if and only if: $x_0 \in \mathcal{I} \implies \phi(t, x_0) \in \mathcal{I}, \quad \forall t \in \mathbb{R}^+$

Any trajectory starting in the set remains in inside it

- **Invariant sets guarantee stability**
  Lyapunov stability: solutions starting "close enough" to the equilibrium (within a distance δ) remain "close enough" forever (within a distance ε))

- **Invariant sets further certify asymptotic stability via Lyapunov's direct method**
  Asymptotic stability: solutions that start close enough not only remain close enough but also eventually converge to the equilibrium.)

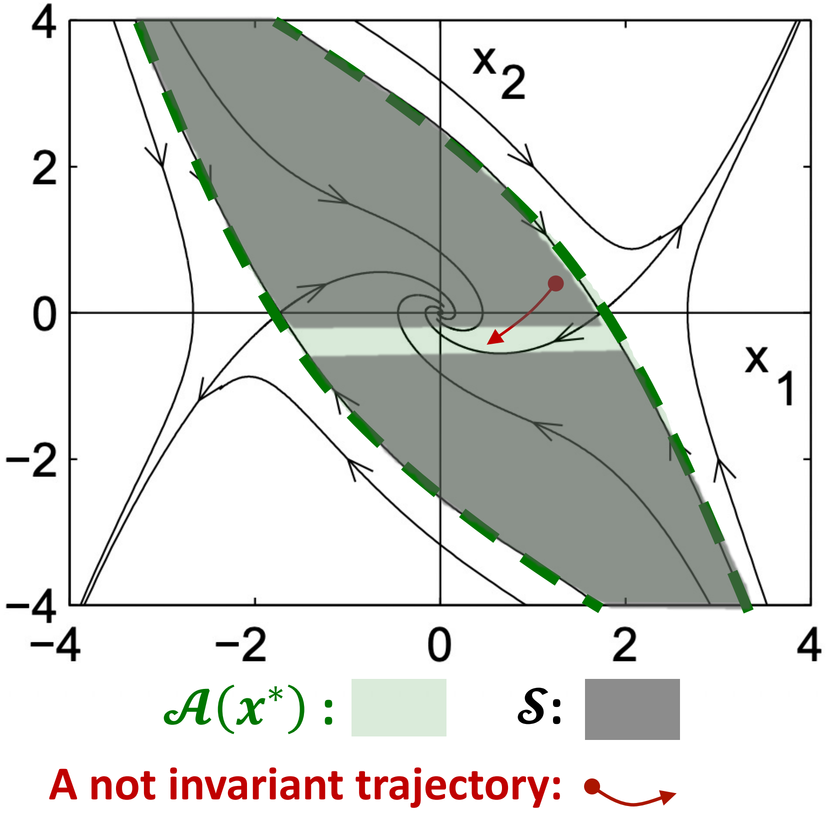
- **Regions of attraction are invariant sets, and so are the outcome of most approximation methods!**

Challenges of working with invariant set

Learning ROA $\mathcal{A}(x^*)$ by finding an invariant set $\mathcal{S} \subseteq \mathcal{A}(x^*)$

- $\mathcal{S}$ needs to be a connected set

Example 1: $\mathcal{S} \subseteq \mathcal{A}(x^*)$ is not connected, not invariant!

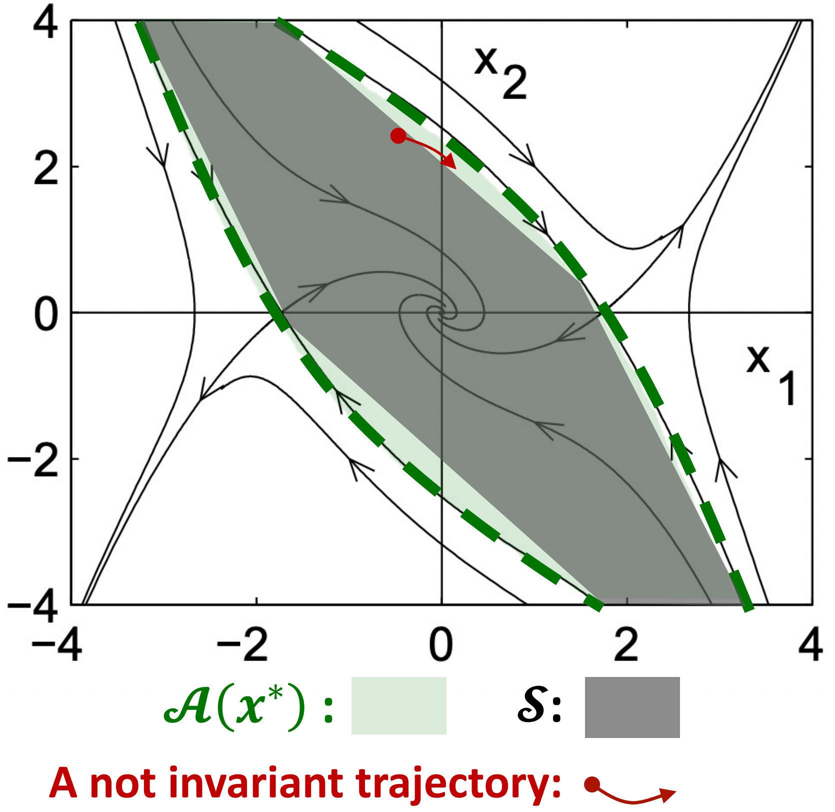


Challenges of working with invariant set

Learning ROA $\mathcal{A}(x^*)$ by finding an invariant set $\mathcal{S} \subseteq \mathcal{A}(x^*)$

- $\mathcal{S}$ needs to be a connected set
- f should point inwards for $x \in \partial\mathcal{S}$

Example 2: $\boldsymbol{\mathcal{S}} \subseteq \boldsymbol{\mathcal{A}}(\boldsymbol{x}^*)$, f points outward on $\boldsymbol{\partial\mathcal{S}}$, not invariant

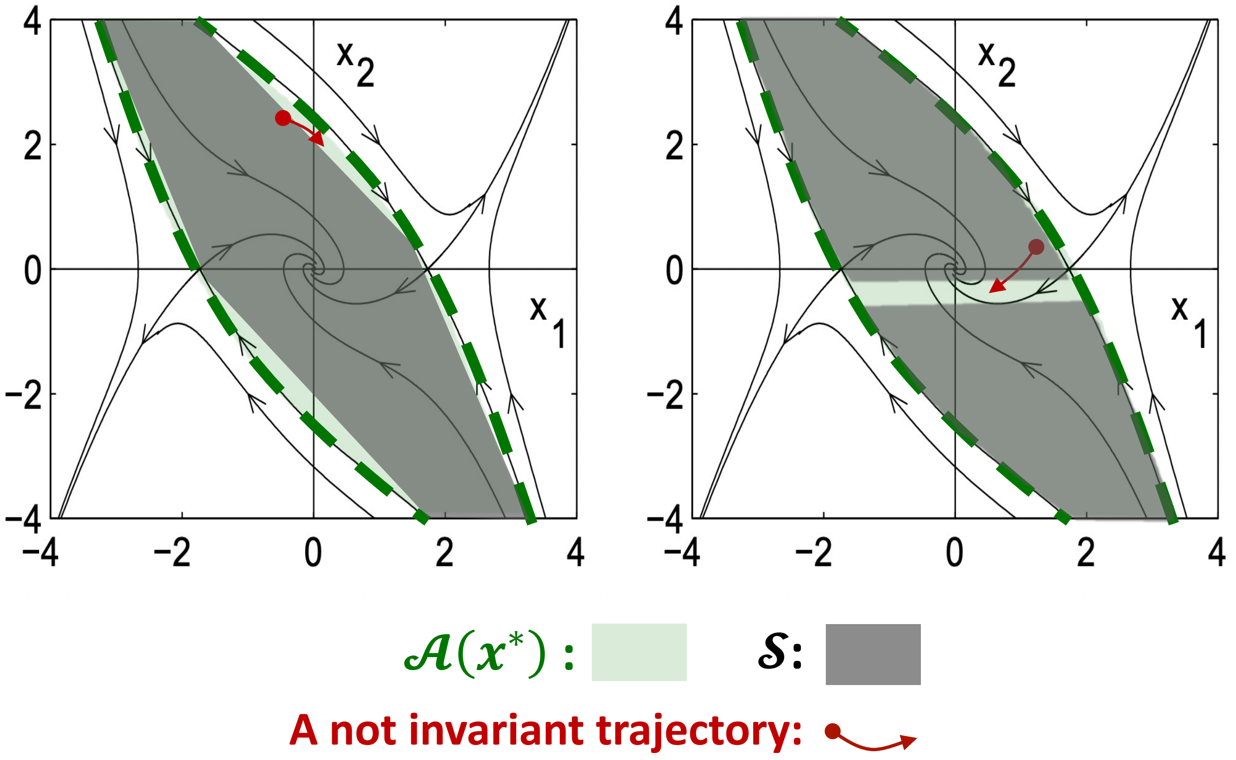


Challenges of working with invariant set

Learning ROA $\mathcal{A}(x^*)$ by finding an invariant set $\mathcal{S} \subseteq \mathcal{A}(x^*)$

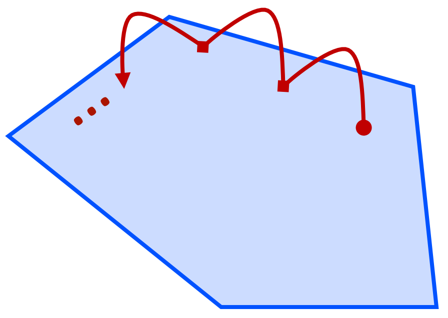
- $\mathcal{S}$ needs to be a connected set
- f should point inwards for $x \in \partial\mathcal{S}$

A subset of an invariant set is not necessary an invariant set



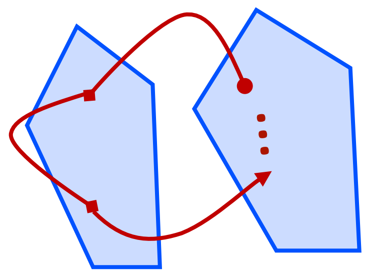
Recurrent sets: Letting things go, and come back

A set $\mathcal{R} \subseteq \mathbb{R}^d$ is **recurrent** if and only if for any $x_0 \in \mathcal{R}$, whenever $\phi(t, x_0) \notin \mathcal{R}$, $t \geq 0$, then $\exists t' > t$ such that $\phi(t', x_0) \in \mathcal{R}$.



Property of Recurrent Sets

- $\mathcal{R}$ need **not** be **connected**
- $\mathcal{R}$ does **not** require f to **point inwards** on all $\partial\mathcal{R}$



***Lemma** 1. Consider a compact recurrent set $\mathcal{R}$. Then for any point $x_0 \in \mathcal{R}$ and time $\tau > 0$, there exist a $\tau' > \tau$, such that $\phi(\tau', x_0) \in \mathcal{R}$.*

Recurrent sets, while not invariant, guarantee that solutions that start in this set, will come back **infinitely often, forever!**

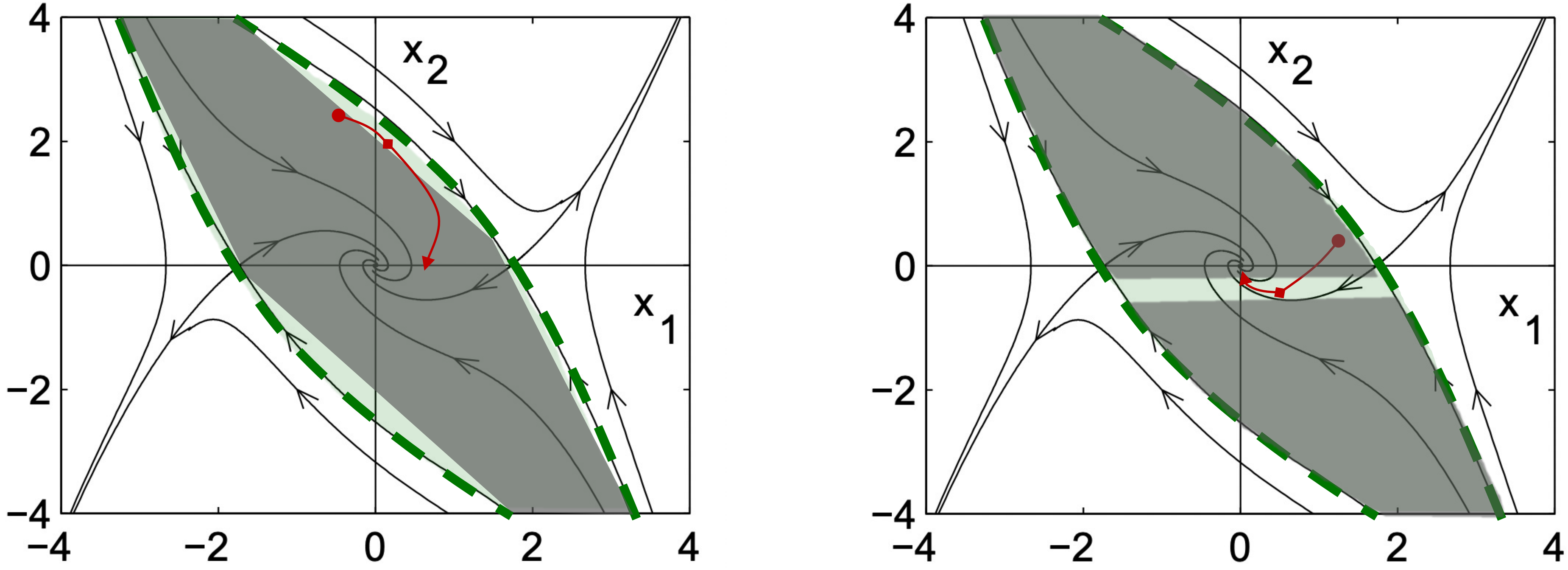
Recurrent set $\mathcal{R}$:

A recurrent trajectory:

Recurrent Sets: Letting things go, and come back

A set $\mathcal{R} \subseteq \mathbb{R}^d$ is **recurrent** if and only if whenever $x_0 \in \mathcal{R}, \exists t' > 0\ s.t.\ \phi(t', x_0) \in \mathcal{R}$

Previous two good inner approximations of $\mathcal{A}(x^*)$ are recurrent sets



Recurrent sets are subsets of the region of attraction

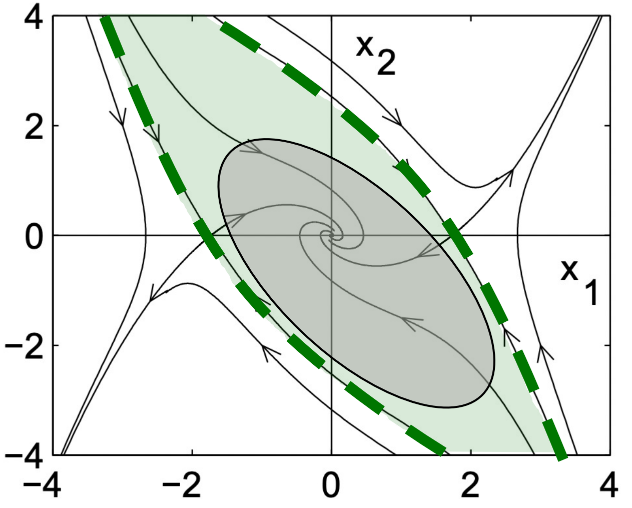
A set $\mathcal{R} \subseteq \mathbb{R}^d$ is **recurrent** if and only if whenever $x_0 \in \mathcal{R}, \exists t' > 0 \; s.t. \; \phi(t', x_0) \in \mathcal{R}$

Theorem *2. Let $\mathcal{R} \subset \mathbb{R}^d$ be a <u>compact</u> set satisfying $\partial\mathcal{R} \cap \Omega(f) = \emptyset$.*

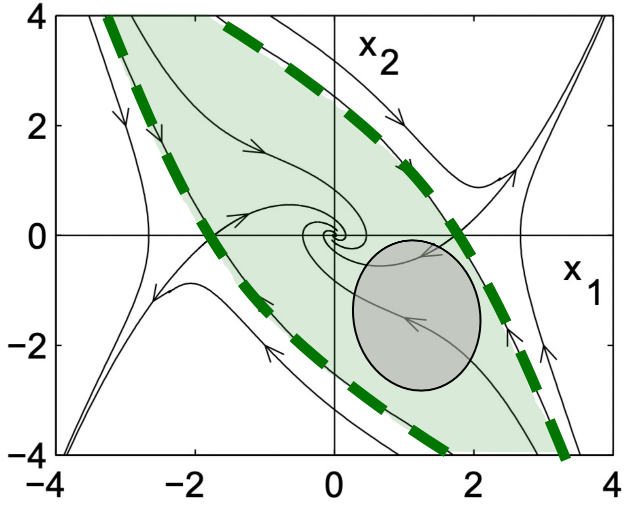
Then:

$$\boxed{\mathcal{R} \text{ is recurrent} \iff \begin{array}{c} \mathcal{R} \cap \Omega(f) \neq \emptyset \\ \mathcal{R} \subset \mathcal{A}(\mathcal{R} \cap \Omega(f)) \end{array}}$$

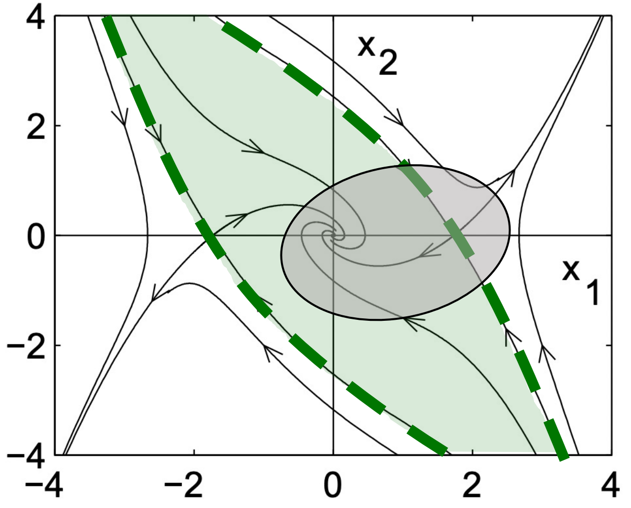
recurrent



not recurrent



not recurrent



$\mathcal{R}$:

$\mathcal{A}(x^*)$:

Recurrent sets are subsets of the region of attraction

A set $\mathcal{R} \subseteq \mathbb{R}^d$ is **recurrent** if and only if whenever $x_0 \in \mathcal{R}, \exists t' > 0\ s.t.\ \phi(t', x_0) \in \mathcal{R}$

***Theorem** 2. Let $\mathcal{R} \subset \mathbb{R}^d$ be a <u>compact</u> set satisfying $\partial\mathcal{R} \cap \Omega(f) = \emptyset$.*

Then:

$$\boxed{\mathcal{R} \text{ is recurrent} \Longleftrightarrow \begin{array}{c} \mathcal{R} \cap \Omega(f) \neq \emptyset \\ \mathcal{R} \subset \mathcal{A}(\mathcal{R} \cap \Omega(f)) \end{array}}$$

***Proof**: [Sketch]*

($\Longrightarrow$)

- $x_0 \in \mathcal{R}$*, the solution* $\phi(t, x_0)$ *visits* $\mathcal{R}$ *infinitely often, forever.*
- *Build a sequence* $\{x(t_n)\}_{n=0}^{\infty} \in \mathcal{R}$ *with* $\lim_{n \to +\infty} t_n = +\infty$
- *Bolzano-Weierstrass* $\Longrightarrow$ *convergent subsequence* $x(t_{n_i}) \to \bar{\boldsymbol{x}} \in \Omega(f) \cap \mathcal{R} \neq \emptyset$

($\Longleftarrow$) Trivial.

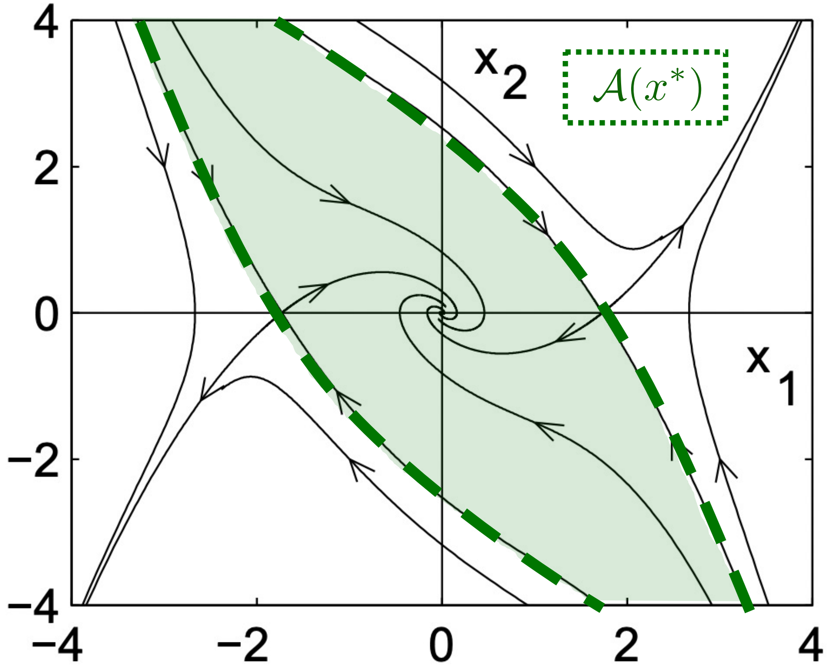
Recurrent sets are subsets of the region of attraction

A set $\mathcal{R} \subseteq \mathbb{R}^d$ is **recurrent** if and only if whenever $x_0 \in \mathcal{R}, \exists t' > 0 \; s.t. \; \phi(t', x_0) \in \mathcal{R}$

***Assumption** 2. The ω-limit set $\Omega(f)$ is composed by **hyperbolic equilibrium points**, with only one of them, say x^*, being asymptotically stable.*

***Corollary** 2. Let Assumptions 1 and 2 hold, and $\mathcal{R} \subset \mathbb{R}^d$ be a <u>compact</u> set satisfying $\partial\mathcal{R} \cap \Omega(f) = \emptyset$. Then:*

$$\mathcal{R} \text{ is recurrent} \iff \begin{array}{c} \mathcal{R} \cap \Omega(f) = \{x^*\} \\ \mathcal{R} \subset \mathcal{A}(x^*) \end{array}$$



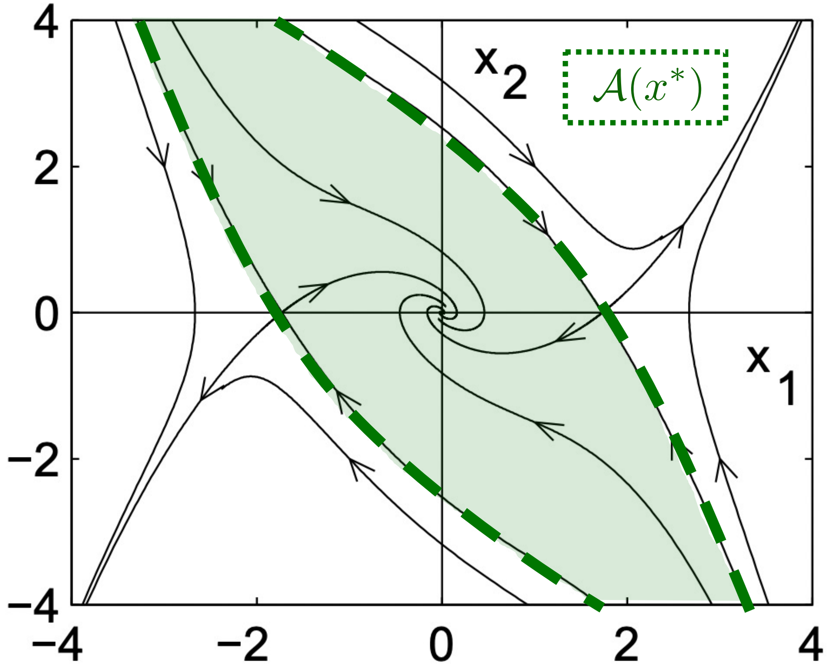
Recurrent sets are subsets of the region of attraction

A set $\mathcal{R} \subseteq \mathbb{R}^d$ is **recurrent** if and only if whenever $x_0 \in \mathcal{R}$, $\boxed{\exists t' > 0}$ $s.t.\ \phi(t', x_0) \in \mathcal{R}$

?

***Corollary** 2. Let Assumptions 1 and 2 hold, and $\mathcal{R} \subset \mathbb{R}^d$ be a compact set satisfying $\partial\mathcal{R} \cap \Omega(f) = \emptyset$. Then:*

$$\mathcal{R} \text{ is recurrent} \Longleftrightarrow \begin{array}{c} \mathcal{R} \cap \Omega(f) = \{x^*\} \\ \mathcal{R} \subset \mathcal{A}(x^*) \end{array}$$



Idea: Use recurrence as a mechanism for finding inner approximations of $\mathcal{A}(x^*)$

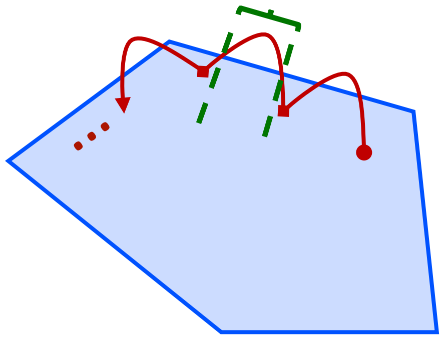
Potential Issues:

- We do not know how long it takes to come back!
- We need to adapt results to trajectory samples

τ-recurrent sets

A set $\mathcal{R}$ is **τ-recurrent** if whenever $x_0 \in \mathcal{R}, \exists\, t' \in (0, \tau]$ s.t. $\phi(t', x_0) \in \mathcal{R}$

Time elapsed $\leq \tau$



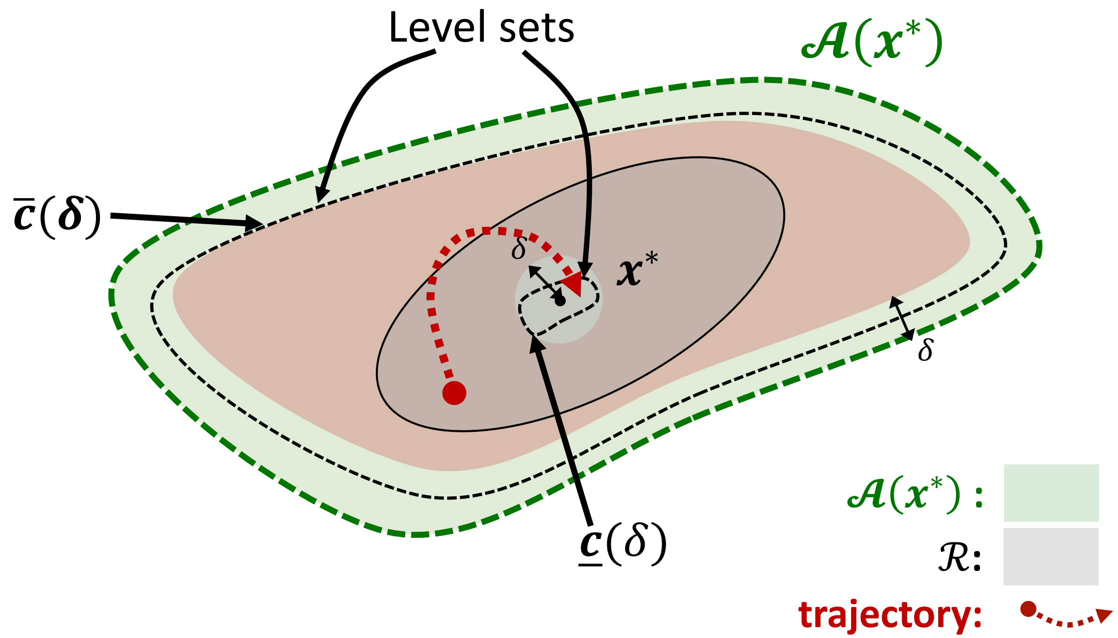
τ-recurrent set $\mathcal{R}$:

trajectory:

***Theorem** 3. Under Assumption 1, any compact set* $\mathcal{R}$ satisfying:

$$x^* + \mathcal{B}_\delta \subseteq \mathcal{R} \subseteq \mathcal{A}(x^*) \backslash \{\partial \mathcal{A}(x^*) + \text{int}\, \mathcal{B}_\delta\}$$

is τ-recurrent for $\tau \geq \bar{\tau}(\delta) := \frac{\underline{c}(\delta) - \bar{c}(\delta)}{a(\delta)}$.



***Proof**: [Sketch]*

- *Assumption 1* $\implies \exists$ *Lyapunov function (Zubov '64)*
  - $V(x^*) = 0$, $0 < V(x) < 1$ for all $x \in \mathcal{A}(x^*) \backslash x^*$
  - $\nabla V(x^*)^T f(x^*) = 0$
  - $\nabla V(x)^T f(x) < 0$ for all $x \in \mathcal{A}(x^*) \backslash x^*$
- *Define* $\bar{c}(\delta) := \max_{x \in \mathcal{A}_\delta} V(x), \quad \underline{c}(\delta) := \min_{x \in \mathcal{A}_\delta} V(x),$

  and $\quad a(\delta) := \max_{x \in C_\delta} \nabla V(x)^T f(x),$

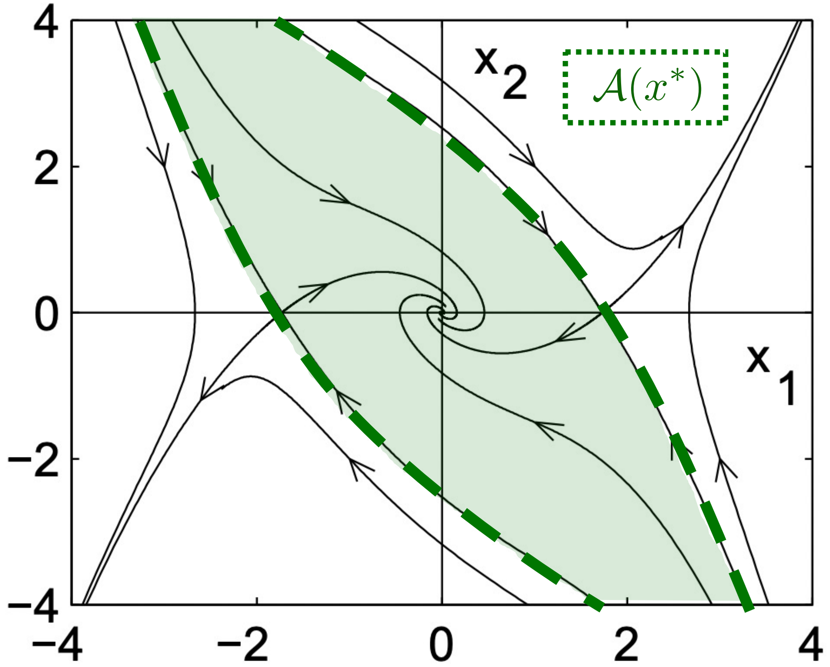
  where $C_\delta = \{x \in \mathbb{R}^d : \underline{c}(\delta) \leq V(x) \leq \bar{c}(\delta)\}$.

Recurrent sets are subsets of the region of attraction

A set $\mathcal{R} \subseteq \mathbb{R}^d$ is **recurrent** if and only if whenever $x_0 \in \mathcal{R}$, $\boxed{\exists t' > 0}$ $s.t.\ \phi(t', x_0) \in \mathcal{R}$ **?**

Corollary *2. Let Assumptions 1 and 2 hold, and $\mathcal{R} \subset \mathbb{R}^d$ be a <u>compact</u> set satisfying $\partial\mathcal{R} \cap \Omega(f) = \emptyset$. Then:*

$$\mathcal{R} \text{ is recurrent} \Longleftrightarrow \begin{array}{c} \mathcal{R} \cap \Omega(f) = \{x^*\} \\ \mathcal{R} \subset \mathcal{A}(x^*) \end{array}$$



Idea: Use recurrence as a mechanism for finding inner approximations of $\mathcal{A}(x^*)$

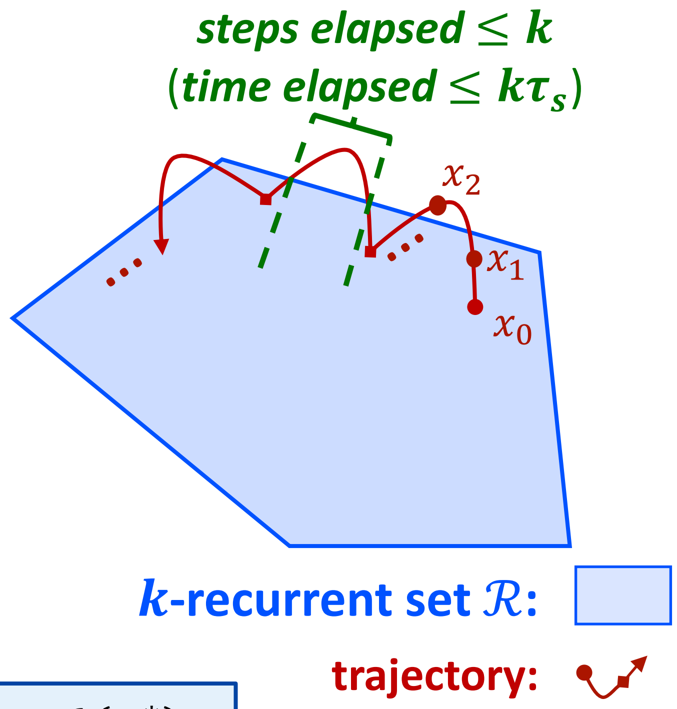
Potential Issues:

- We do not know how long it takes to come back! ✔
- We need to adapt results to trajectory samples

Learning recurrent sets from k-length trajectory samples

- Consider finite length trajectories:
$$x_n = \phi(n\tau_s, x_0), \qquad x_0 \in \mathbb{R}^d, n \in \mathbb{N},$$
where $\tau_s > 0$ is the sampling period.

- A set $\mathcal{R} \subseteq \mathbb{R}^d$ is k**-recurrent** if whenever $x_0 \in \mathcal{R}$, then $\exists\, n \in \{1, \ldots, k\}$ s.t. $x_n \in \mathcal{R}$



Sufficiency:

$\mathcal{R}$ is k-recurrent ➡ $\mathcal{R}$ is τ-recurrent with $\tau = k\tau_s$ ✚ $\mathcal{R}$ is compact $\partial\mathcal{R} \cap \Omega(f) = \emptyset$ ➡ $\mathcal{R} \subset \mathcal{A}(x^*)$

(Corollary 2, under Assumption 2)

Necessity:

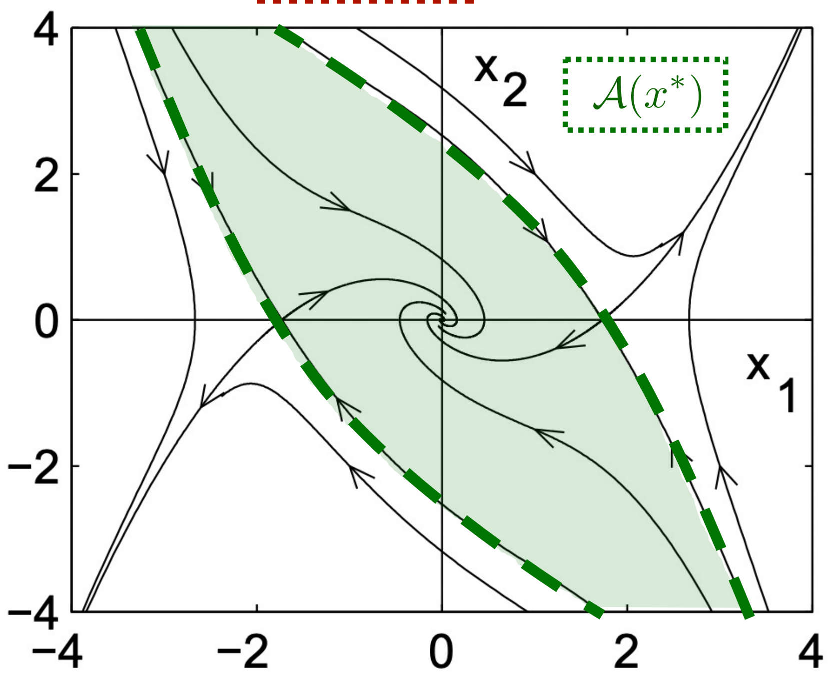
Theorem *4. Under Assumption 1, any compact set* $\mathcal{R}$ satisfying:
$$\mathcal{B}_\delta + x^* \subseteq \mathcal{R} \subseteq \mathcal{A}(x^*)\backslash\{\partial\mathcal{A}(x^*) + \text{int}\, \mathcal{B}_\delta\}$$
is k-recurrent for any $k > \bar{k} := \bar{\tau}(\delta)/\tau_s$.

Recurrent sets are subsets of the region of attraction

A set $\mathcal{R} \subseteq \mathbb{R}^d$ is **recurrent** if and only if whenever $x_0 \in \mathcal{R}$, $\boxed{\exists t' > 0}$? $s.t.\ \phi(t', x_0) \in \mathcal{R}$

***Corollary** 2. Let Assumptions 1 and 2 hold, and $\mathcal{R} \subset \mathbb{R}^d$ be a <u>compact</u> set satisfying $\partial\mathcal{R} \cap \Omega(f) = \emptyset$. Then:*

$$\mathcal{R} \text{ is recurrent} \Longleftrightarrow \begin{array}{c} \mathcal{R} \cap \Omega(f) = \{x^*\} \\ \mathcal{R} \subset \mathcal{A}(x^*) \end{array}$$



Idea: Use recurrence as a mechanism for finding inner approximations of $\mathcal{A}(x^*)$

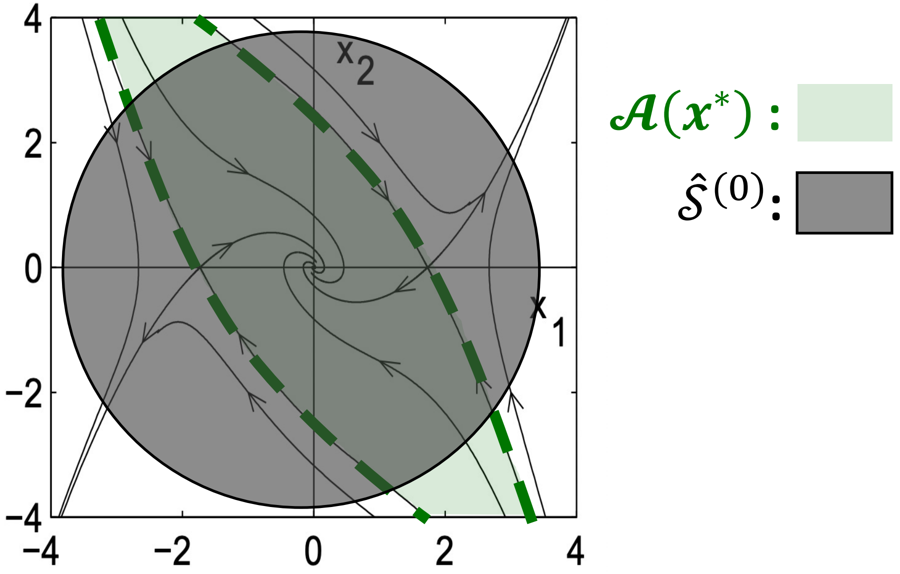
Potential Issues:

- We do not know how long it takes to come back! ✔
- We need to adapt results to trajectory samples ✔

Sphere approximations of RoA

Algorithm:

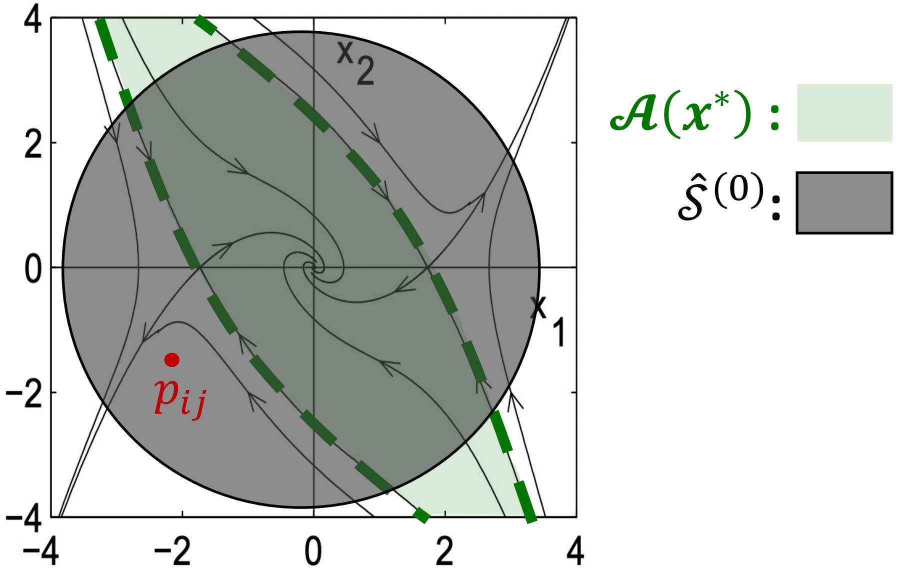
- **Initialize $\hat{\mathcal{S}}^{(0)}$ *as*** $\boxed{\hat{\mathcal{S}}^{(0)} := \{x \mid \|x\|_2 \le b^{(0)} := c\} \supseteq \mathcal{B}_\delta}$



Sphere approximations of RoA

Algorithm:

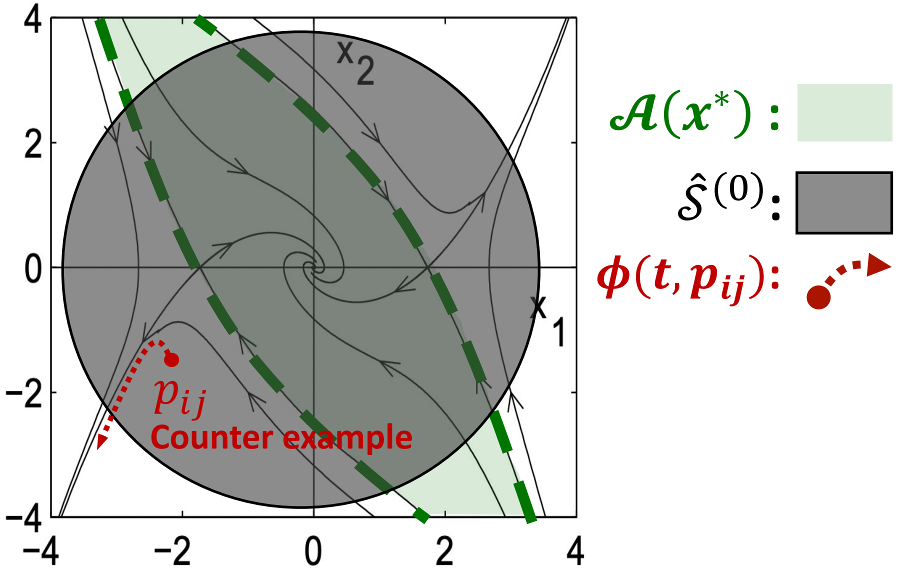
- **Initialize** $\hat{\mathcal{S}}^{(0)}$ ***as*** $\boxed{\hat{\mathcal{S}}^{(0)} := \{x \mid \|x\|_2 \le b^{(0)} := c\} \supseteq \mathcal{B}_\delta}$
- **For** *iteration* $i = 0,1,\ldots$ **do:** (set updates)
  - ***For*** *iteration* $j = 0,1,\ldots$ **do:** (samples)
    - *Generate random sample* $p_{ij} \in \hat{\mathcal{S}}^{(i)}$ *uniformly*



Sphere approximations of RoA

Algorithm:

- **Initialize** $\hat{\mathcal{S}}^{(0)}$ ***as*** $\hat{\mathcal{S}}^{(0)} := \{x \mid \|x\|_2 \leq b^{(0)} := c\} \supseteq \mathcal{B}_\delta$
- **For** *iteration* $i = 0,1,\dots$ **do:**
  - ***For*** *iteration* $j = 0,1,\dots$ **do:**
    - *Generate random sample* $p_{ij} \in \hat{\mathcal{S}}^{(i)}$ *uniformly*
    - **If** p_{ij} *is a counter-example w.r.t* $\hat{\mathcal{S}}^{(i)}$ **do:**



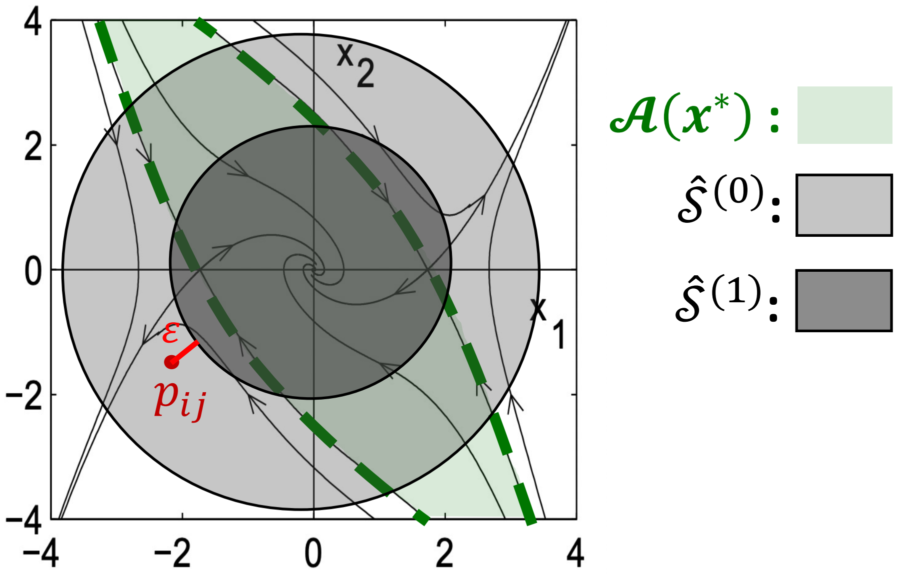
We say sample point p_{ij} is a valid k-recurrent point w.r.t current approximation $\hat{\mathcal{S}}^{(i)}$ if starting from $x_0 = p_{ij}$,

$$\exists\, n \in \{1,\dots,k\},\ \text{s.t.}\ \ x_n \in \hat{\mathcal{S}}^{(i)}.$$

Otherwise, we say p_{ij} is a counter-example.

Sphere approximations of RoA

Algorithm:



- **Initialize** $\hat{\mathcal{S}}^{(0)}$ ***as*** $\hat{\mathcal{S}}^{(0)} := \{x \mid \|x\|_2 \le b^{(0)} := c\} \supseteq \mathcal{B}_\delta$
- **For** *iteration* $i = 0,1,\dots$ **do:**
  - ***For*** *iteration* $j = 0,1,\dots$ **do:**
    - *Generate random sample* $p_{ij} \in \hat{\mathcal{S}}^{(i)}$ *uniformly*
    - **If** p_{ij} *is a counter-example w.r.t* $\hat{\mathcal{S}}^{(i)}$ **do:**
      - *Update* $b^{(i)}$ *to* $b^{(i+1)}$, $\hat{\mathcal{S}}^{(i)}$ *to* $\hat{\mathcal{S}}^{(i+1)}$

We say sample point p_{ij} *is a valid* k*-recurrent point w.r.t current approximation* $\hat{\mathcal{S}}^{(i)}$ *if starting from* $x_0 = p_{ij}$,

$$\exists\, n \in \{1,\dots,k\}, \text{ s.t. } x_n \in \hat{\mathcal{S}}^{(i)}.$$

Otherwise, we say p_{ij} *is a counter-example.*

If p_{ij} *is a counter-example*, we update:

$$b^{(i+1)} = \|p_{ij}\|_2 - \varepsilon;$$

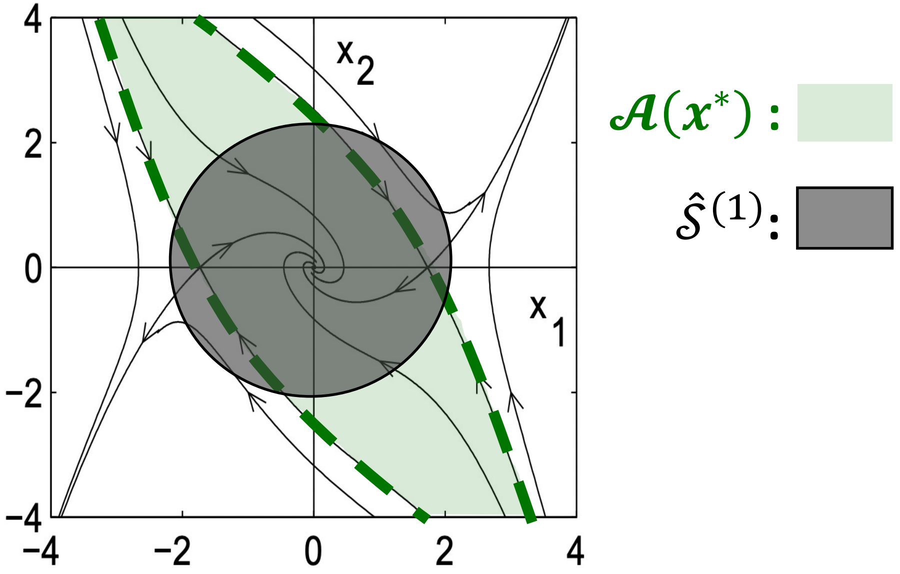
$$\hat{\mathcal{S}}^{(i+1)} = \{x \,|\, \|x\|_2 \le b^{(i+1)}\},$$

where $\varepsilon > 0$ *is an algorithm parameter expressing the level of conservativeness in our update.*

Sphere approximations of RoA

Algorithm:

- **Initialize $\hat{\mathcal{S}}^{(0)}$ *as*** $\boxed{\hat{\mathcal{S}}^{(0)} := \{x \mid \|x\|_2 \le b^{(0)} := c\} \supseteq \mathcal{B}_\delta}$
- **For *iteration* $i = 0,1,\ldots$ do:**
  - ***For iteration* $j = 0,1,\ldots$ do:**
    - *Generate random sample $p_{ij} \in \hat{\mathcal{S}}^{(i)}$ uniformly*
    - **If** *p_{ij} is a counter-example w.r.t $\hat{\mathcal{S}}^{(i)}$* **do:**
      - *Update $b^{(i)}$ to $b^{(i+1)}$, $\hat{\mathcal{S}}^{(i)}$ to $\hat{\mathcal{S}}^{(i+1)}$*
      - **Break**
    - **End if**
  - **End for**
- **End for**



We say sample point p_{ij} is a valid k-recurrent point w.r.t current approximation $\hat{\mathcal{S}}^{(i)}$ if starting from $x_0 = p_{ij}$,

$$\exists\, n \in \{1,\ldots,k\},\ \text{s.t. } x_n \in \hat{\mathcal{S}}^{(i)}.$$

Otherwise, we say p_{ij} is a counter-example.

If p_{ij} is a counter-example, we update:

$$b^{(i+1)} = \|p_{ij}\|_2 - \varepsilon;$$

$$\hat{\mathcal{S}}^{(i+1)} = \{x \mid \|x\|_2 \le b^{(i+1)}\},$$

where $\varepsilon > 0$ is an algorithm parameter expressing the level of conservativeness in our update.

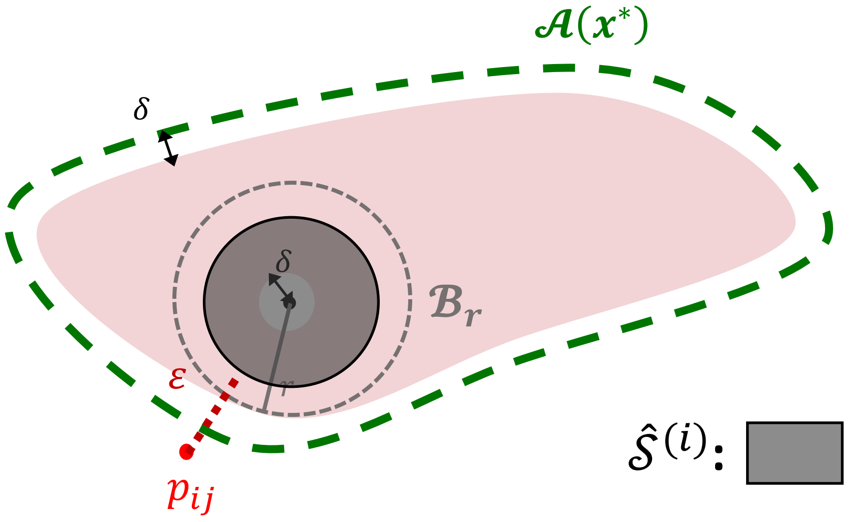
Parameter choice

Choice of ε: $b^{(i+1)} = \left\|p_{ij}\right\| - \varepsilon$

- Given $k > \bar{k}$, any set $\mathcal{S}^{(i)} = \{x: ||x|| \leq b^{(i)}\}$ satisfying:

  $$\mathcal{B}_\delta \subseteq \mathcal{S}^{(i)} \subseteq \mathcal{A}(0) \backslash \{\partial\mathcal{A}(0) + \text{int}\, \mathcal{B}_\delta\}$$

  is k-recurrent.
- Let $\mathcal{B}_r$ the largest ball inside $\mathcal{A}(0) \backslash \{\partial\mathcal{A}(0) + \text{int}\, \mathcal{B}_\delta\}$
- Then, if $\varepsilon \leq r - \delta$ we always guarantee $\mathcal{B}_\delta \subseteq \mathcal{S}^{(i)}$



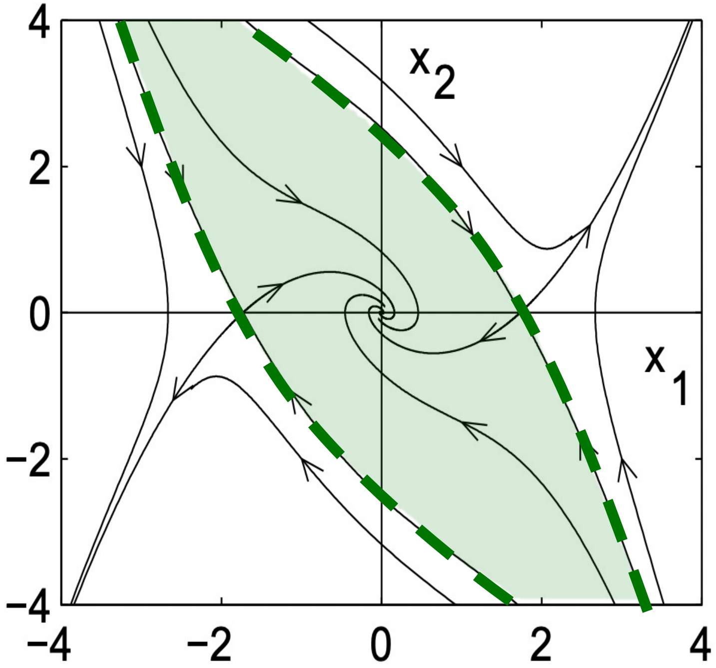
Choice of trajectory length k:

- $\bar{k}$ depends highly non-trivially on δ.
- If $k < \bar{k}$, we get $b^{(i)} < 0 \Longrightarrow$ Failure!
- **Solution:** doubling the size of k, i.e., $k^+ = 2k$, **every time we fail.**

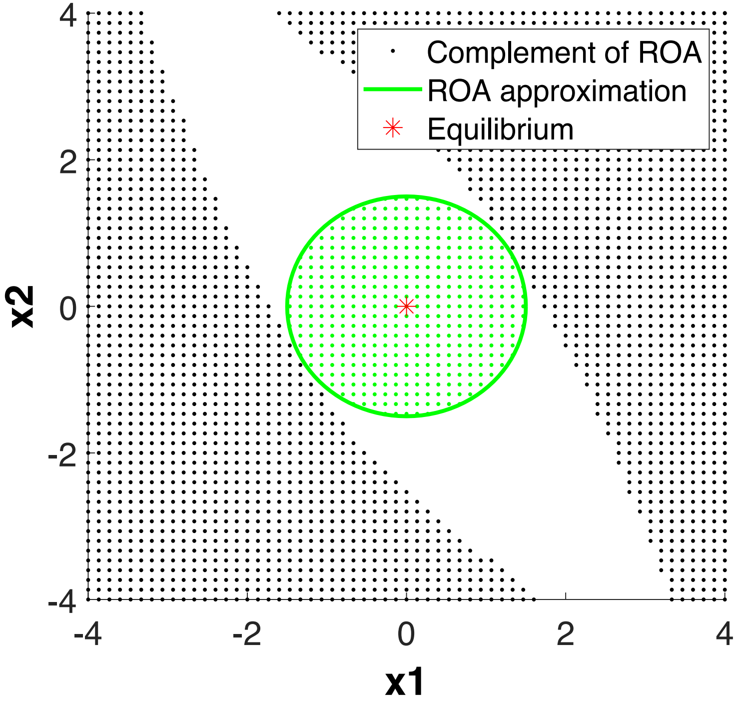
With k-doubling, the total number of counter-examples is bounded by

$$\#\text{counter-examples} \leq \frac{b^{(0)}}{\varepsilon} \log_2 \bar{k}$$

Algorithm Result - Sphere Approximations



$\mathcal{A}(0)$:

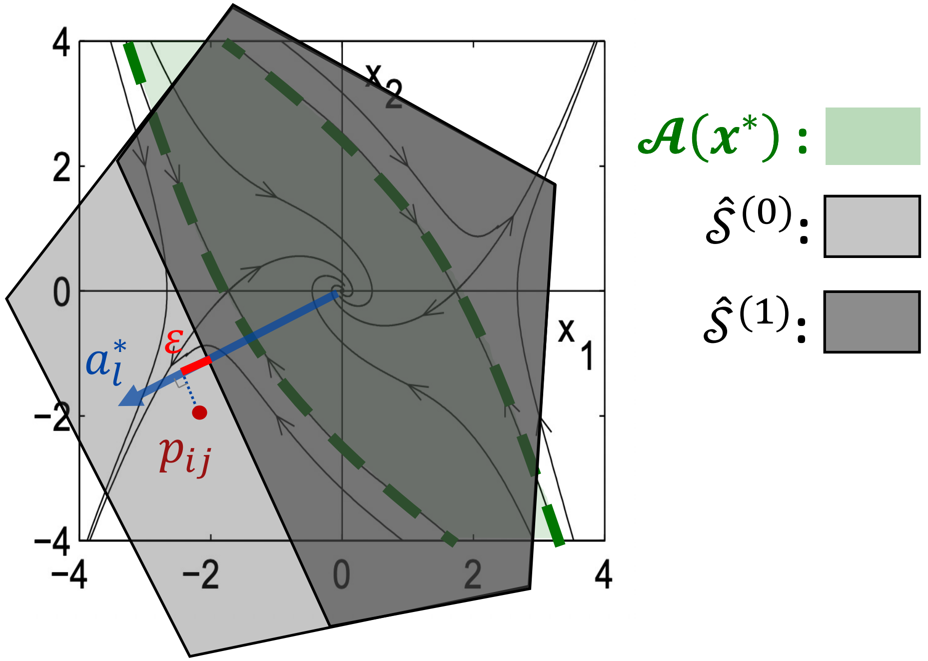


Polytope approximations of RoA

Algorithm:

- **Initialize** $\hat{\mathcal{S}}^{(0)}$ ***as*** $\hat{\mathcal{S}}^{(0)} := \{x | Ax \leq b^{(0)} := c\mathbb{1}_n\} \supseteq \mathcal{B}_\delta$

Exploration directions matrix $A := [a_1, \dots, a_n] \subseteq \mathbb{R}^{n \times d}$, where each row vector a_l is a normalized exploration direction indexed by $l \in \{1, \dots, n\}$.



- **For** *iteration* $i = 0,1, \dots$ **do:**
  - ***For*** *iteration* $j = 0,1, \dots$ **do:**
    - *Generate random sample* $p_{ij} \in \hat{\mathcal{S}}^{(i)}$ *uniformly*
    - **If** p_{ij} *is a counter-example w.r.t* $\hat{\mathcal{S}}^{(i)}$ **do:**
      - *Update* $b^{(i)}$ *to* $b^{(i+1)}$, $\hat{\mathcal{S}}^{(i)}$ *to* $\hat{\mathcal{S}}^{(i+1)}$
      - **Break**
    - **End if**
  - **End for**
- **End for**

If p_{ij} *is a counter-example*, we update:

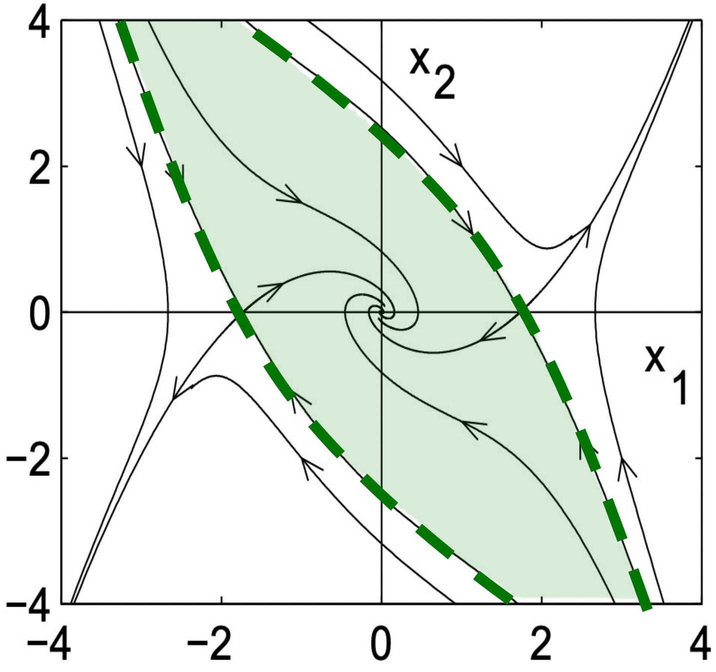
$$b^{(i+1)} = \begin{cases} b_{l^*}^{(i+1)} = a_{l^*} p_{ij} - \varepsilon \\ b_l^{(i+1)} = b_l^{(i)}; \end{cases}$$

$$\hat{\mathcal{S}}^{(i+1)} = \{x | Ax \leq b^{(i+1)}\},$$

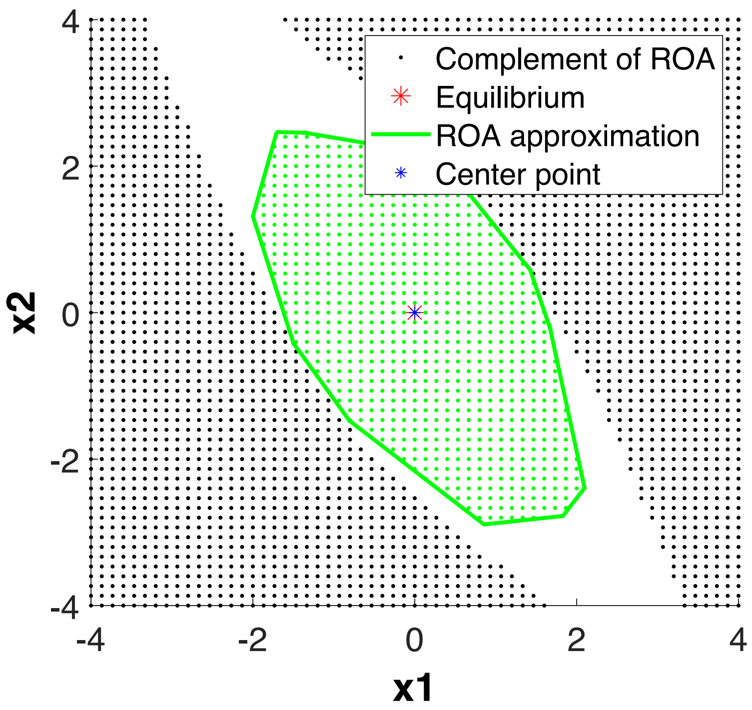
where $\varepsilon > 0$ *is fixed and* $l^* = \mathrm{argmax}_{l \in \{1,\dots,n\}} \frac{a_l^T p_{ij}}{\|a_l\| \|p_{ij}\|}$,

is the index of exploration direction that minimizes the angle between p_{ij} *and* a_l.

Algorithm Result – Polytope Approximation

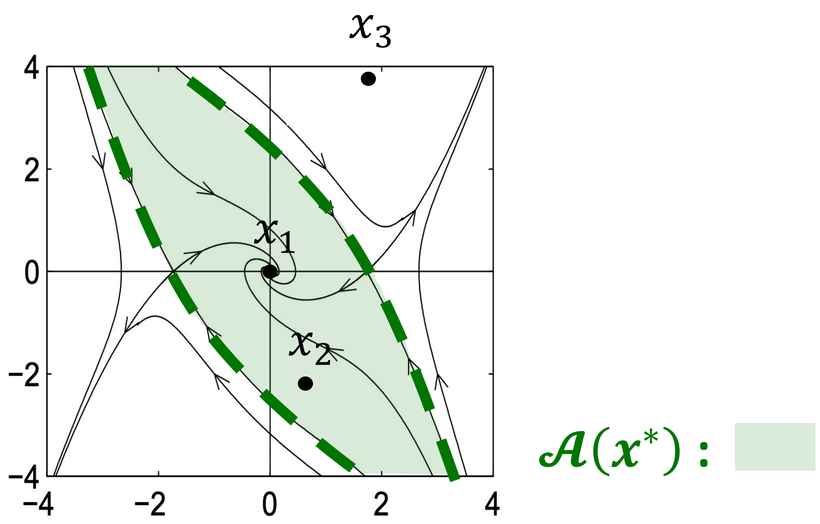


$\mathcal{A}(0)$:



Multi-center approximation

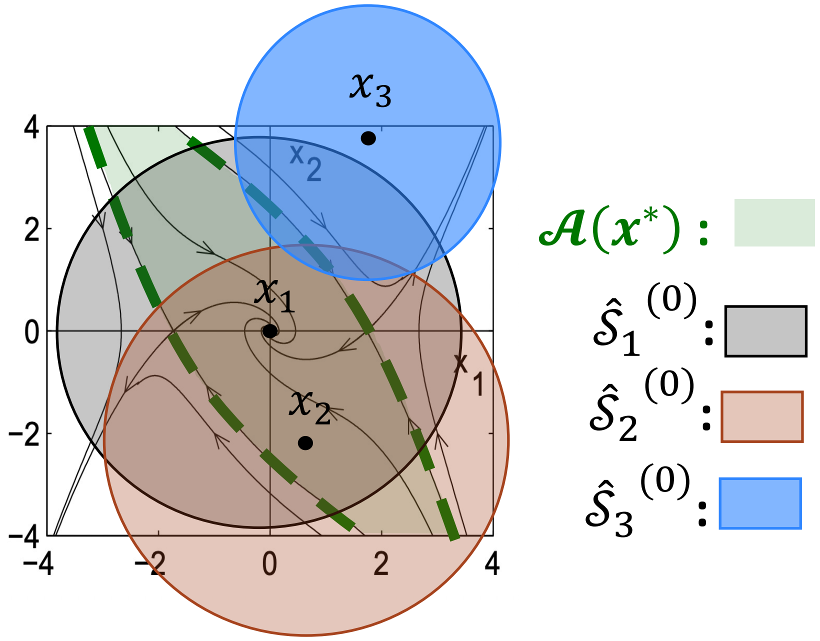
- **Consider $\boldsymbol{h} \in \mathbb{N}^+$ center points $\boldsymbol{x_q}$ indexed by $\boldsymbol{q} \in \{\mathbf{1}, \dots, \boldsymbol{h}\}$.**
  - Let the first center point $x_1 = x^* = 0$
  - Additional center point $x_2, \dots, x_h$ can be designed chosen uniformly.



Multi-center approximation

- **Consider $h \in \mathbb{N}^+$ center points x_q indexed by $q \in \{1, \dots, h\}$.**
  - Let the first center point $x_1 = x^* = 0$
  - Additional center point $x_2, \dots, x_h$ can be designed chosen uniformly.

- **Respectively defined approximations centered at each x_q**
  - (Sphere case) $\hat{\mathcal{S}}_q^{(i)} := \{x \mid \|x - x_q\|_2 \leq b_q^{(i)}\}$
  - (Polytope case) $\hat{\mathcal{S}}_q^{(i)} := \{x \mid A(x - x_q) \leq b_q^{(i)}\}$

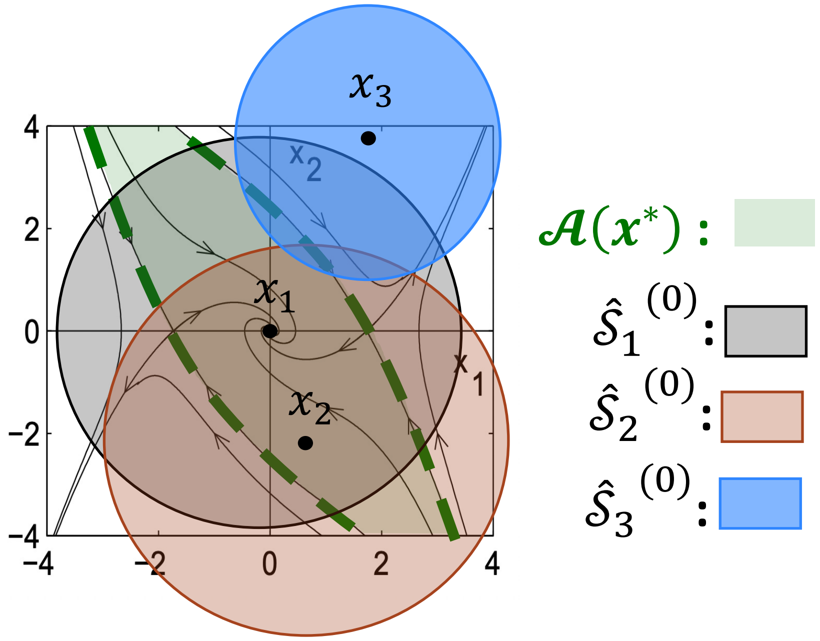


Multi-center approximation

- **Consider $\boldsymbol{h} \in \mathbb{N}^{+}$ center points $\boldsymbol{x_q}$ indexed by $\boldsymbol{q} \in \{\boldsymbol{1}, \dots, \boldsymbol{h}\}$.**
  - Let the first center point $x_1 = x^* = 0$
  - Additional center point $x_2, \dots, x_h$ can be designed chosen uniformly.

- **Respectively defined approximations centered at each $\boldsymbol{x_q}$**
  - (Sphere case) $\hat{\mathcal{S}}_q^{(i)} \coloneqq \{x \mid \|x - x_q\|_2 \le b_q^{(i)}\}$
  - (Polytope case) $\hat{\mathcal{S}}_q^{(i)} \coloneqq \{x \mid A(x - x_q) \le b_q^{(i)}\}$

- **Multiple centers approximation** $\hat{\mathcal{S}}_{\text{multi}}^{(i)} := \cup_{q=1}^{h} \hat{\mathcal{S}}_q^{(i)}$



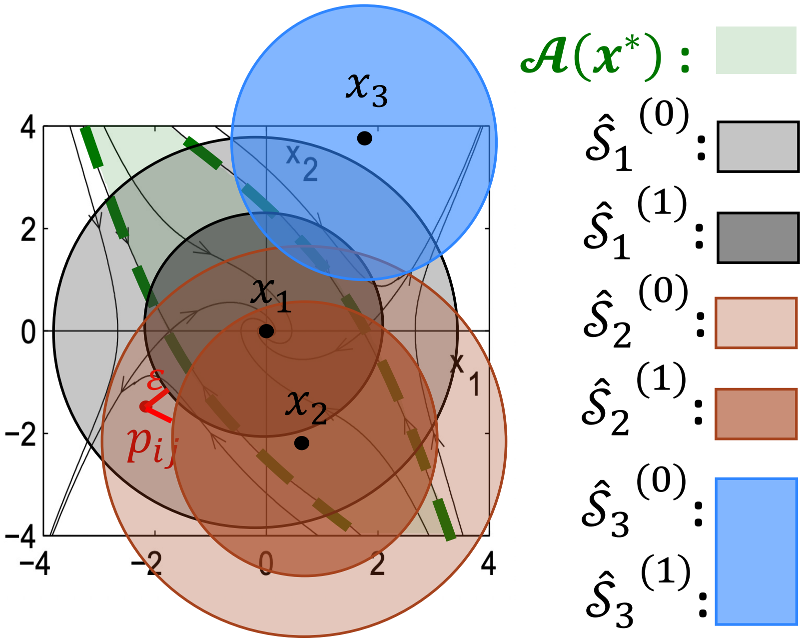
Multi-center approximation

- **Consider $\boldsymbol{h} \in \mathbb{N}^+$ center points $\boldsymbol{x_q}$ indexed by $\boldsymbol{q} \in \{\mathbf{1}, \dots, \boldsymbol{h}\}$.**
  - Let the first center point $x_1 = x^* = 0$
  - Additional center point $x_2, \dots, x_h$ can be designed chosen uniformly.

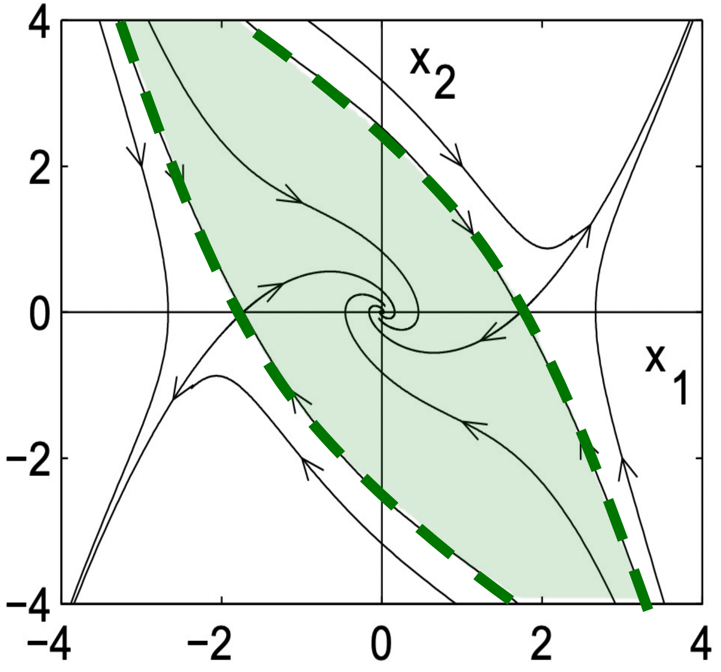
- **Respectively defined approximations centered at each $\boldsymbol{x_q}$**
  - (Sphere case) $\hat{\mathcal{S}}_q^{(i)} \coloneqq \{x \mid \|x - x_q\|_2 \le b_q^{(i)}\}$
  - (Polytope case) $\hat{\mathcal{S}}_q^{(i)} \coloneqq \{x \mid A(x - x_q) \le b_q^{(i)}\}$

- **Multiple centers approximation** $\hat{\mathcal{S}}_{\text{multi}}^{(i)} := \cup_{q=1}^{h} \hat{\mathcal{S}}_q^{(i)}$

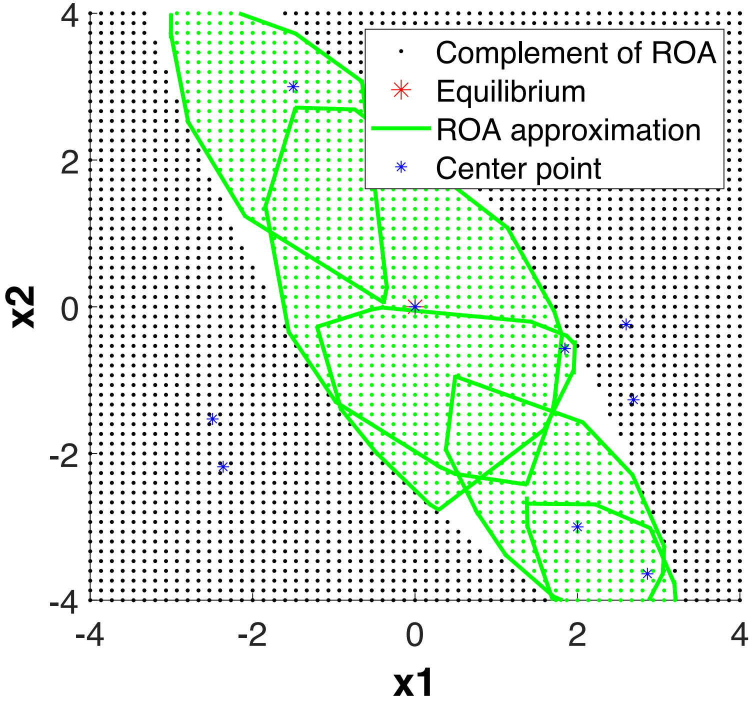
- **If $\mathbf{p_{ij}}$ is a counter-example w.r.t** $\hat{\mathcal{S}}_{\text{multi}}^{(i)}$
  - We shrink every $\hat{\mathcal{S}}_q^{(i)}$ satisfying $p_{ij} \in \hat{\mathcal{S}}_q^{(i)}$
  - For the rest approximations, we simply let $\hat{\mathcal{S}}_q^{(i+1)} = \hat{\mathcal{S}}_q^{(i)}$



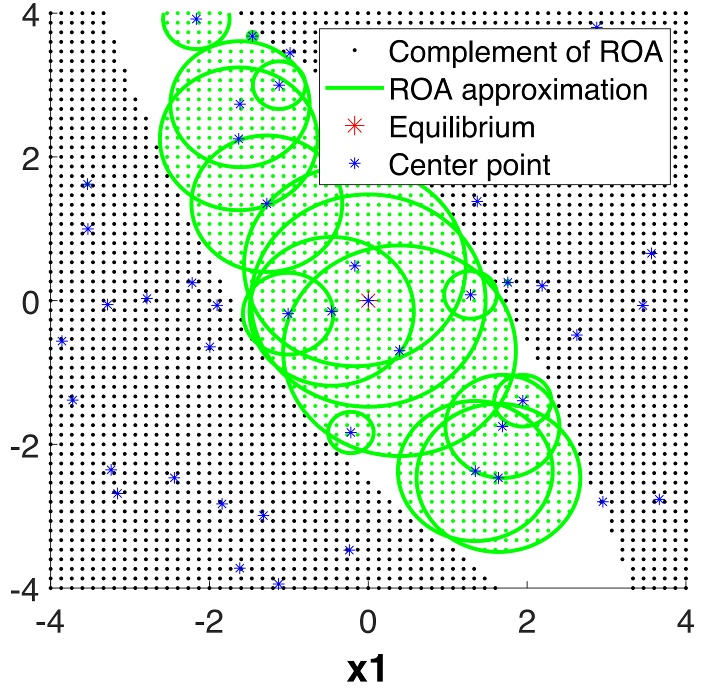
Algorithm results – Multi-center approximation



$\mathcal{A}(0)$:



(10 polytope approximations)



(50 sphere approximations)

Question: Are we asking too much?

- Learnability requires uniform approximation errors across the ***entire domain***

  Q: Can we provide local guarantees, and progressively expand as needed?

[arXiv ‘22] Shen, Bichuch, M

- Lyapunov functions and control barrier functions require strict and exhaustive notions of ***invariance***

  Q: Can we substitute invariance with less restrictive properties?

[arXiv ‘22] Shen, Bichuch, M

- Control synthesis usually aims for the ***best*** (optimal) controller

  Q: Can we focus on feasibility, rather than optimality?

[arXiv ’21, L4DC 22] Castellano, Min, Bazerque, M

[arXiv 22] Shen, Bichuch, M, *Model-free Learning of Regions of Attraction via Recurrent Sets,* **submitted to CDC 2022,** preprint arXiv:2204.10372.
[L4DC 22] Castellano, Min, Bazerque, M, *Reinforcement Learning with Almost Sure Constraints,* **Learning for Dynamics and Control (L4DC) Conference, 2022**
[arXiv 21] Castellano, Min, Bazerque, M, *Learning to Act Safely with Limited Exposure and Almost Sure Certainty,* **submitted to IEEE TAC, 2021, under review**, preprint arXiv:2105.08748

Question: Are we asking too much?

- Learnability requires uniform approximation errors across the ***entire domain***

  Q: Can we provide local guarantees, and progressively expand as needed?

[arXiv ‘22] Shen, Bichuch, M

- Lyapunov functions and control barrier functions require strict and exhaustive notions of ***invariance***

  Q: Can we substitute invariance with less restrictive properties?

[arXiv ‘22] Shen, Bichuch, M

- Control synthesis usually aims for the ***best*** (optimal) controller

  Q: Can we focus on feasibility, rather than optimality?

[arXiv ’21, L4DC 22] Castellano, Min, Bazerque, M

[arXiv 22] Shen, Bichuch, M, *Model-free Learning of Regions of Attraction via Recurrent Sets,* **submitted to CDC 2022,** preprint arXiv:2204.10372.
[L4DC 22] Castellano, Min, Bazerque, M, *Reinforcement Learning with Almost Sure Constraints,* **Learning for Dynamics and Control (L4DC) Conference, 2022**
[arXiv 21] Castellano, Min, Bazerque, M, *Learning to Act Safely with Limited Exposure and Almost Sure Certainty,* **submitted to IEEE TAC, 2021, under review**, preprint arXiv:2105.08748

[Submitted on 9 Dec 2021 (v1), last revised 7 Apr 2022 (this version, v2)]

Reinforcement Learning with Almost Sure Constraints

Agustin Castellano, Hancheng Min, Juan Bazerque, Enrique Mallada

arXiv > cs > arXiv:2112.05198

[Submitted on 18 May 2021 (v1), last revised 25 May 2021 (this version, v2)]

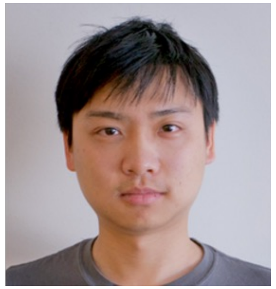
Learning to Act Safely with Limited Exposure and Almost Sure Certainty

Agustin Castellano, Hancheng Min, Juan Bazerque, Enrique Mallada

arXiv > eess > arXiv:2105.08748

Agustin Castellano

JOHNS HOPKINS UNIVERSITY



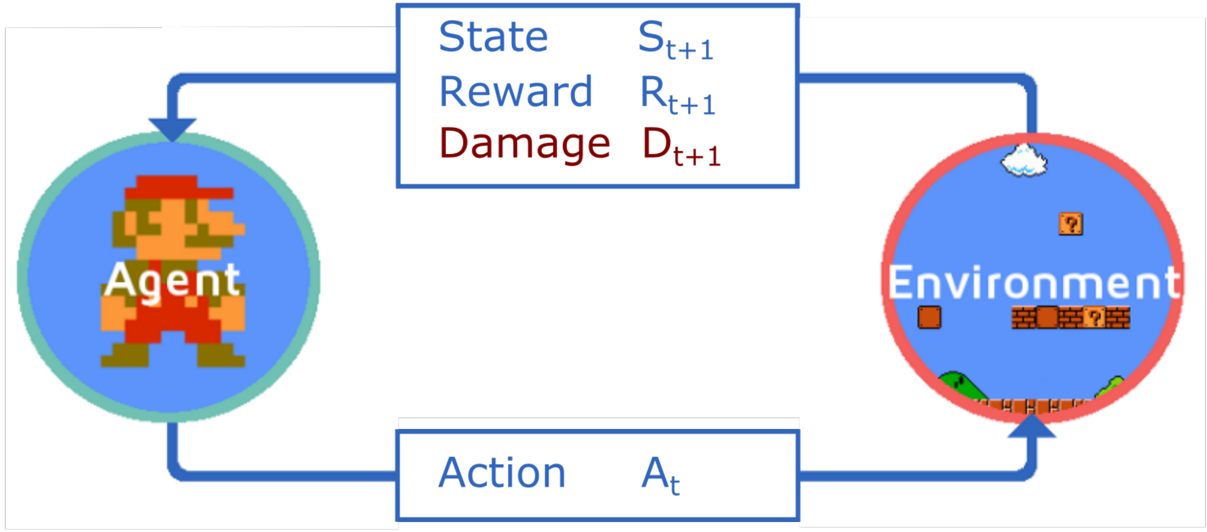
Hancheng Min

JOHNS HOPKINS UNIVERSITY

Juan Bazerque

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Safety-critical Sequential Decision Making



Requirements:

High Priority -> Safety

- o Sequential / Online / Real-time
- o Limited Failures/Mistakes
- o High-probability (or A.S.) Guarantees

Lower Priority -> Accuracy

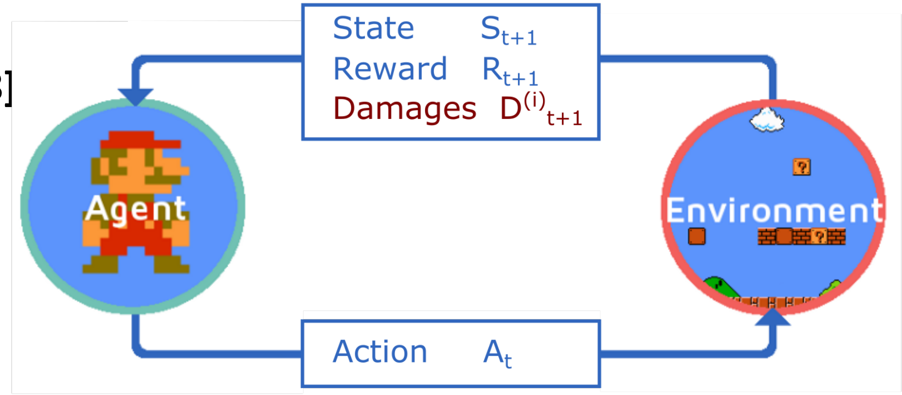
- o Optimality of the policy
- o Full characterization of the safety set?

Key ideas:

- Focus on almost sure **feasibility**, not optimality (Egerstedt et al.,2018)
- Enhanced with **logical** feedback, naturally arising from constraint violations
  - Damage may depend on R_t, or not. May not be directly accessible

Background

- **Constrained Markov Decision Processes (CMDPs) [Altman'98]**



$$\max_{\pi\in\Pi} \quad V^\pi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t R_{t+1}|S_0=s\right]$$
$$\text{s.t.:} \quad C_i^\pi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t D^{(i)}_{t+1}|S_0=s\right] \leq c_i \quad i=1,\ldots,m$$

- Solvable if MDP is "known" (Linear Program).
- $\exists$ stationary optimal solution $\pi^*(a|s)$

- **What to do if MDP is "unknown"? Examples of Offline (OFF) and Online (ON) methods**

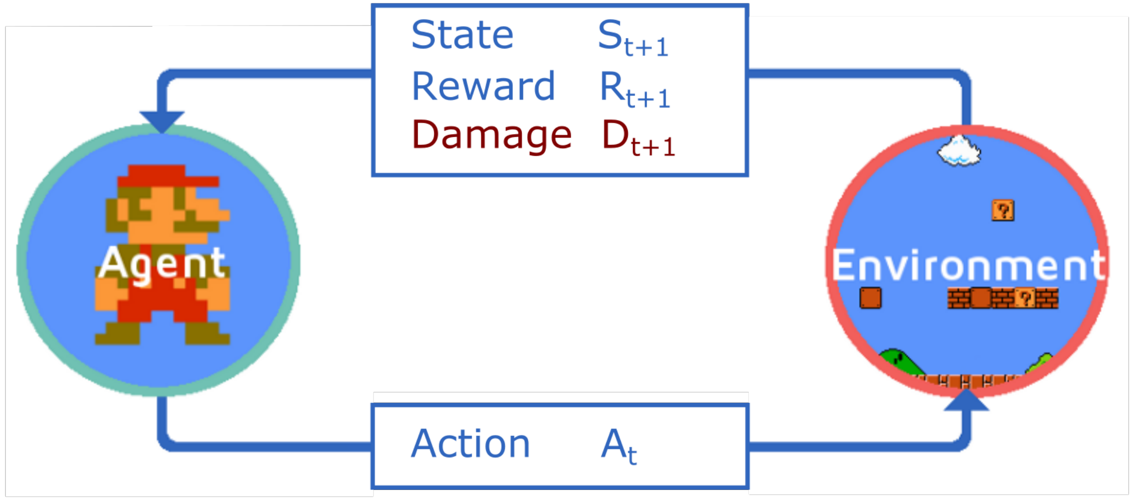
- (OFF) Learn transitions and reward/constraint signals, solve for a (near) optimal policy.

- (ON) Primal-dual methods.

Reinforcement Learning with Almost Sure Constraints

$$V^*(s) := \max_{\pi} \ \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \mid S_0 = s\right]$$

$$\text{s.t.: } \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \cancel{\gamma^t} D_{t+1} \mid S_0 = s\right] \leq 0 \iff D_{t+1} = 0 \text{ almost surely } \forall t$$



- Constraints not given a priori: Need to learn from experience!
- **Notice:** Model free ➔ Constraint violations are inevitable
- Damage indicator $D_t \in \{0,1\}$ turns on ($D_t = 1$) when constraints are violated

Formulation via hard barrier indicator

Safe RL problem:

$$V^*(s) := \max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \mid S_0 = s\right]$$

$$\text{s.t.: } D_{t+1} = 0 \text{ almost surely } \forall t$$

$\sim$

Equivalent unconstrained formulation:

$$\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} + \underbrace{\log[1 - D_{t+1}]}_{\begin{cases} 0 & if\ D_{t+1} = 0 \\ -\infty & if\ D_{t+1} = 1 \end{cases}} \mid S_0 = s\right]$$

Questions/Comments:

- Is this just a standard RL problem with $\tilde{R}_{t+1} = R_{t+1} + \log(1 - D_{t+1})$?
- Standard MDP assumptions for Value Iteration, Bellman's Eq., Optimality Principle, etc., do not hold!
- Not to mention convergence of stochastic approximations.

Key idea: Separate the problem of safety from optimality

Hard Barrier Action-Value Functions

Consider the Q-function for a given policy π,

$$Q^\pi(s,a) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\left(\gamma^t R_{t+1} - \log(1-D_{t+1})\right) \mid S_0 = s, A_0 = a\right]$$

and define the hard-barrier function

$$B^\pi(s,a) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} - \log(1-D_{t+1}) \mid S_0 = s, A_0 = a\right]$$

Notes on $\boldsymbol{B^\pi(s,a)}$:

- $\boldsymbol{B^\pi(s,a) \in \{0, -\infty\}}$
- Summarizes safety information
  - $\boldsymbol{B^\pi(s,a) = 0}$ iff π is safe after choosing $A_t = a$ when $S_t = s$
- It is independent of the reward process

Separation Principle

Theorem (Separation principle)

Assume rewards R_{t+1} are bounded almost surely for all t. Then for every policy π:

$$Q^{\pi}(s,a) = Q^{\pi}(s,a) + B^{\pi}(s,a)$$

In particular, for optimal π_*

$$Q^{*}(s,a) = Q^{*}(s,a) + B^{*}(s,a)$$

Idea: Learn feasibility (encoded in B^*) independently from optimality.

Optimal Hard Barrier Action-Value Function

Theorem (Separation principle)

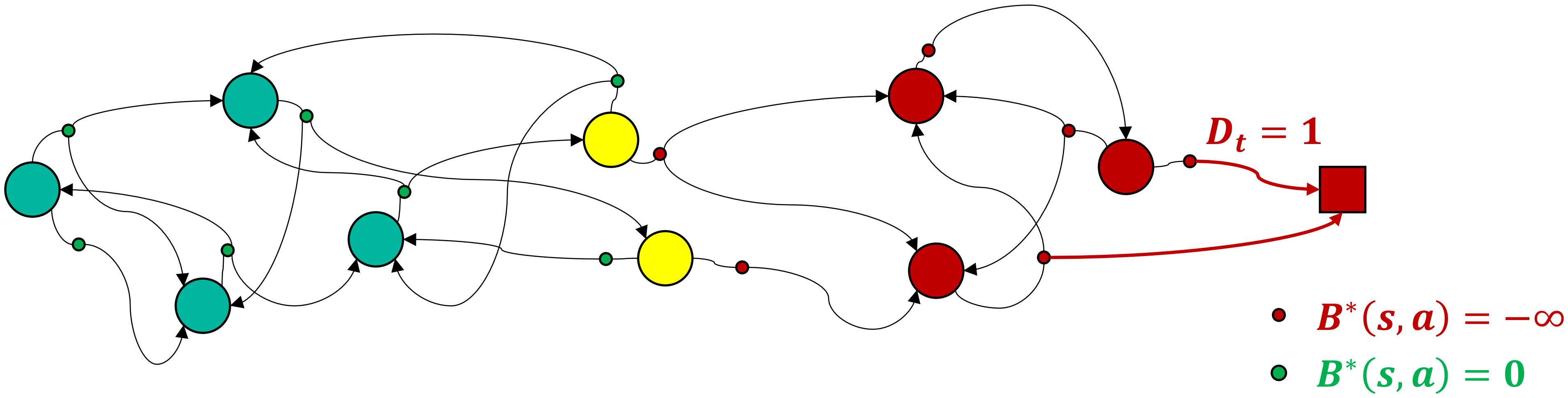
Assume rewards R_{t+1} are bounded almost surely for all t. Then for optimal π_* we have

$$Q^*(s,a) = Q^*(s,a) + B^*(s,a)$$

Understanding $\boldsymbol{B^*(s,a)}$:

$\boldsymbol{B^*(s,a)} \in \{\mathbf{0}, -\infty\}$ summarizes safety information of the entire MDP

- $\boldsymbol{B^*(s,a)} = \mathbf{0}$ if $\exists$ safe π after choosing $A_t = a$ when $S_t = s$
- $\boldsymbol{B^*(s,a)} = -\infty$ if no safe policy exists after choosing $A_t = a$ when $S_t = s$



Optimal Hard Barrier Action-Value Function

Theorem (Separation principle)

Assume rewards R_{t+1} are bounded almost surely for all t. Then for optimal π_* we have

$$Q^*(s,a) = Q^*(s,a) + B^*(s,a)$$

Understanding $\boldsymbol{B^*(s,a)}$:

$\boldsymbol{B^*(s,a)} \in \{\mathbf{0}, -\infty\}$ summarizes safety information of the entire MDP

- $\boldsymbol{B^*(s,a)} = \mathbf{0}$ if $\exists$ safe π after choosing $A_t = a$ when $S_t = s$
- $\boldsymbol{B^*(s,a)} = -\infty$ if no safe policy exists after choosing $A_t = a$ when $S_t = s$

Theorem (Bellman Equation for B^*)

Let $B^*(s,a) := \max_{\pi} B^{\pi}(s,a)$, then the following holds:

$$B^*(s,a) = \mathbb{E}\left[-\log(1 - D_{t+1}) + \max_{a'} B^*(S_{t+1}, a') \;\middle|\; S_0 = s, A_0 = a \right]$$

Idea: Use this Bellman Equation to learn B^* (coming up next)

Learning the barrier...

Algorithm 3: `barrier_update`

B-function (initialized as all-zeroes);
Input: (s, a, s', d)
Output: Barrier-function $B(s, a)$
$B(s,a) \leftarrow B(s,a) + \log(1-d) + \max_{a'} B(s', a')$

Pros:

- Wraps around learning algorithms (Q-learning, SARSA)
- Use the HBF to trim exploration set and avoid repeating unsafe actions

...with a generative model:

- Sample a transition (s, a, s', d) according to the MDP. Update barrier function.

Algorithm 5: Barrier Learner Algorithm

Data: Constrained Markov Decision Process $\mathcal{M}$
Result: Optimal action-value function B^*
$\boxed{\text{Initialize } B^{(0)}(s,a) = 0, \forall (s,a) \in \mathcal{S} \times \mathcal{A}}$
for $t = 0, 1, \cdots$ **do**
  $\boxed{\text{Draw } (s_t, a_t) \sim \text{Unif}(\{(s,a) : B^{(t)}(s,a) \neq -\infty\})}$
  Sample transition (s_t, a_t, s'_t, d_t) according to
  $P(S_1 = s'_t, D_1 = d_t | S_0 = s_t, A_0 = a_t)$
  $\boxed{B^{(t+1)} \leftarrow \texttt{barrier\_update}(B^{(t)}, s_t, a_t, s'_t, d_t)}$
end

Initially, all (s, a)-pairs are “safe”

Draw (s, a)-pair uniformly among those considered to be “safe” at time t

Update barrier function

Assured Q-Learning with Generative Model

Theorem (Safety Guarantee): Let $T = \min_t \{B^{(t)} = B^*\}$, then

$$\mathbb{E}T \leq (L+1)\frac{|S||A|}{\mu}\left(\sum_{k=1}^{|S||A|}\frac{1}{k}\right)$$

- After $T = \min_t \{B^{(t)} = B^*\}$, all "unsafe" (s, a)-pairs are detected
- μ: Lower bound on the non-zero transition probability

$$\mu = min\{p(s', d|s, a) : p(s', d|s, a) \neq 0\}$$

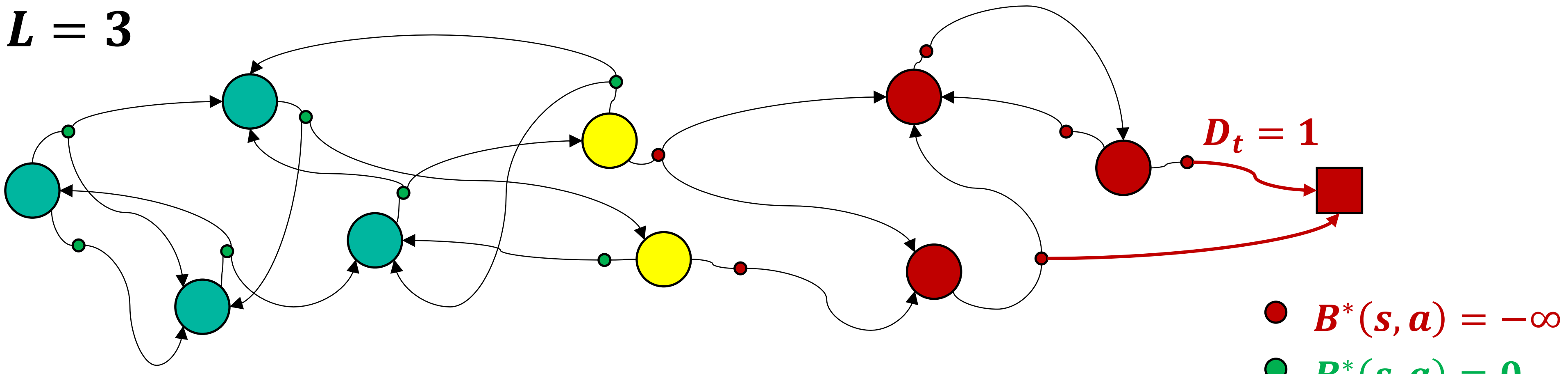
- L: **Lag of the MDP**

$$L = \max_{\substack{(s,a) \\ B^*(s,a)=-\infty}} \left\{ \begin{array}{l} \underline{\text{Minimum}} \text{ number of transitions} \\ \text{needed to observe damage,} \\ \text{starting from unsafe } (s,a) \end{array} \right\}$$

Lag of the MDP: L

$$\mathrm{L} = \max_{\substack{(s,a) \\ B^*(s,a) = -\infty}} \left\{ \begin{array}{l} \underline{\text{Minimum}} \text{ number of transitions needed to} \\ \text{observe damage, starting from unsafe } (s,a) \end{array} \right\}$$

$\boldsymbol{L = 3}$



Assured Q-Learning with Generative Model

Theorem (Sample Complexity): With at least $1-\delta$ probability, the algorithm learns optimal barrier function B^* after

$$(L+1)\frac{|S||A|}{\mu}\left(\sum_{k=1}^{|S||A|}\frac{1}{k}\right)\log\frac{1}{\delta}$$

iterations

- Concentration of sum of exponential random variables

- **Much more sample-efficient** than "learning an ϵ-optimal policy with $1-\delta$ probability" (Li et al. 2020)

$$N=\frac{|S||A|}{(1-\gamma)^4\varepsilon^2}\log^2\left(\frac{|S||A|}{(1-\gamma)\varepsilon\delta}\right)$$

Assured Q-Learning with Generative Model

Theorem (Sample Complexity): With at least $1-\delta$ probability, the algorithm learns optimal barrier function B^* after

$$(L+1)\frac{|S||A|}{\mu}\left(\sum_{k=1}^{|S||A|}\frac{1}{k}\right)\log\frac{1}{\delta}$$

iterations

- Concentration of sum of exponential random variables

- If the Barrier Function is learnt first, then learning an ϵ-optimal policy takes

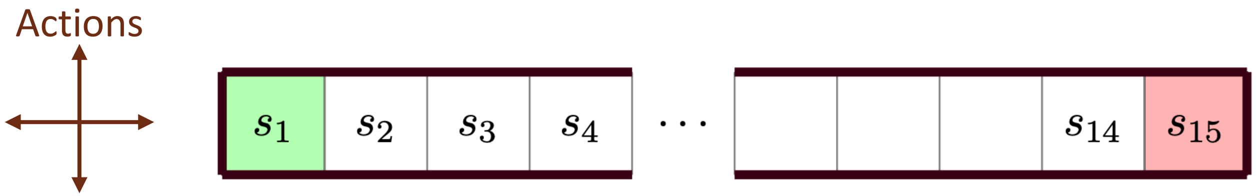
$$N' = \frac{|S_{safe}||A_{safe}|}{(1-\gamma)^4\varepsilon^2}\log^2\left(\frac{|S_{safe}||A_{safe}|}{(1-\gamma)\varepsilon\delta}\right)$$

samples (**Trimming the MDP by learning the barrier**)

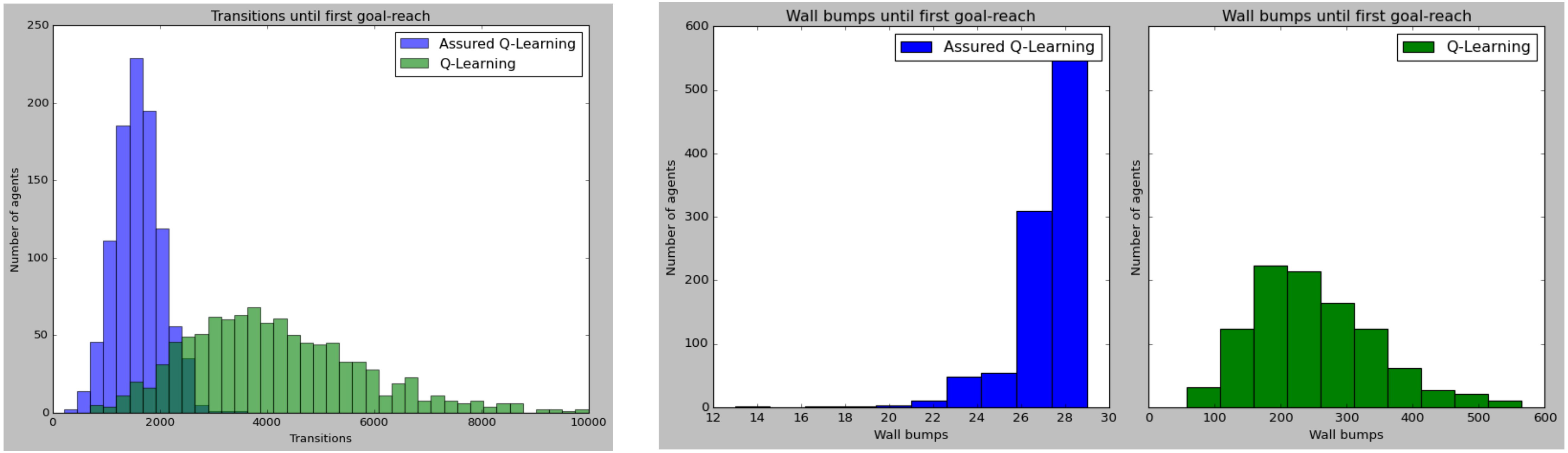
Numerical Experiments

Goal: Reach the end of the aisle $(R_{t+1} = 10)$

Touching the wall gives $D_{t+1} = 1$, resets the episode.



Results



Why does Assured Q-learning perform much better?

If $D_{t+1} = 1 \Longrightarrow B_\pi(s, a) = -\infty \Longrightarrow$ Never take action a at s again!

Takeaways:

- Adding constraints to the problem can accelerate learning
- Barrier function avoids actions that lead to further wall bumps

Numerical Experiments II

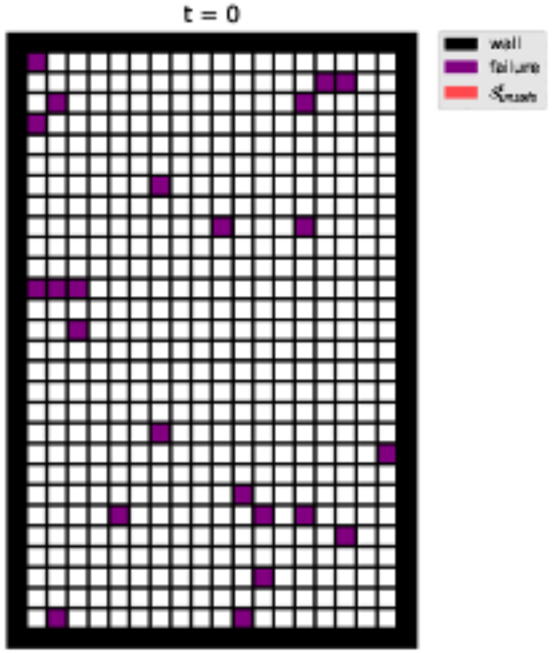
Setup: Rectangular grid, stepping into **holes** gives damage $D_t = 1$.

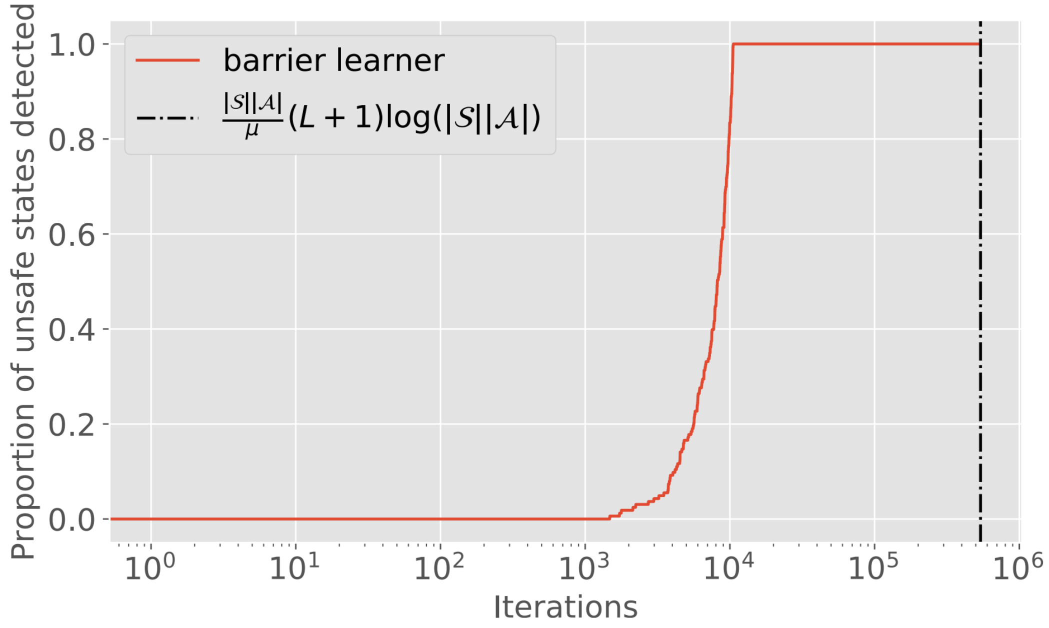
Actions $A = \{up, down, left, right\}$.

With every action, small probability to move to a random adjacent state.

Result: Barrier-learner identifies **all** the state space as unsafe.

Immediately unsafe states (near **damage**) are identified first.





Numerical Experiments II

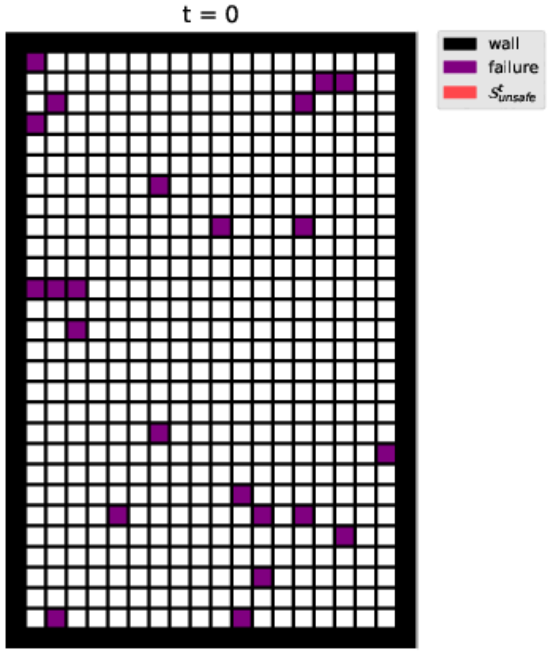
Setup: Rectangular grid, stepping into **holes** gives damage $D_t = 1$.

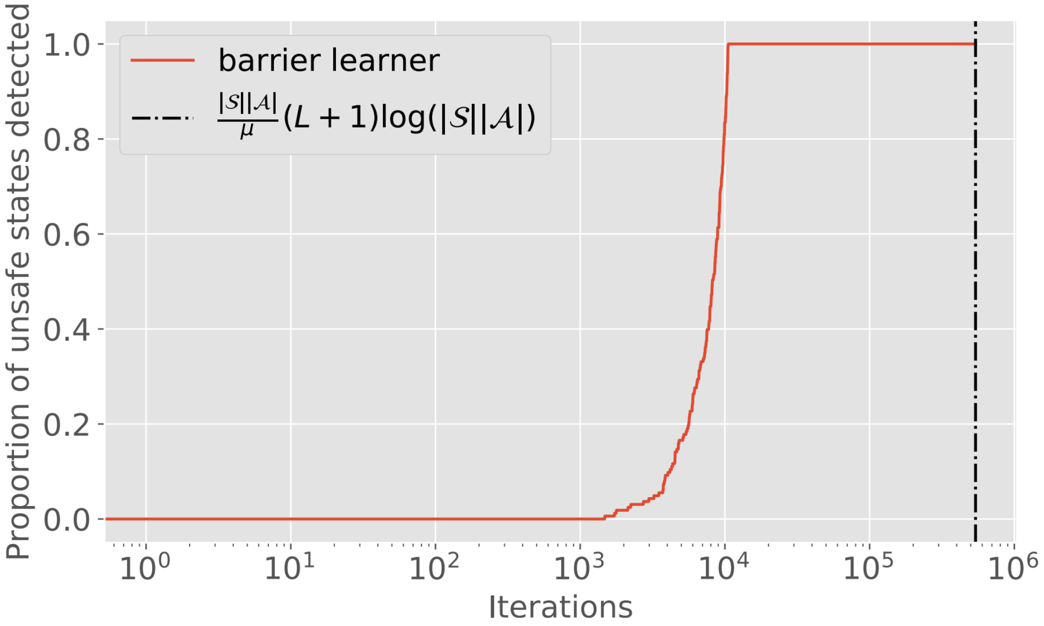
Actions $A = \{up, down, left, right\}$.

With every action, small probability to move to a random adjacent state.

Result: Barrier-learner identifies **all** the state space as unsafe.

Immediately unsafe states (near **damage**) are identified first.





Generalization

So far:

- Studied "assured" RL under a very particular type of constraint

$$V^*(s) := \max_{\pi} \ \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \mid S_0 = s \right]$$

$$\text{s.t.: } \boxed{D_{t+1} = 0 \text{ almost surely } \forall t}$$

Upcoming:

Can we generalize this? E.g.:

$$\boxed{\left(\sum_{t=0}^{\infty} D_{t+1} \right) \leq \Delta \Big| S_0 = s \quad \text{almost surely}}$$

"Allow no more than Δ units of damage along a trajectory"

RL with almost sure constraints and positive budget (Δ)

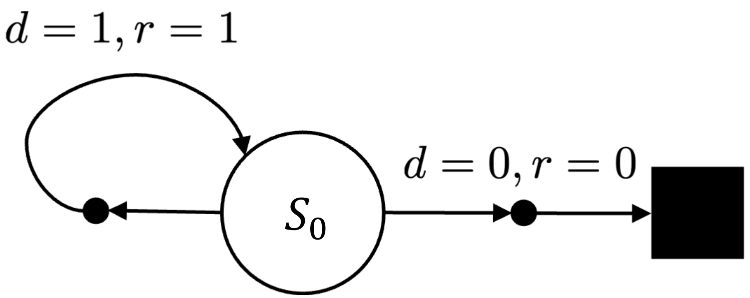
$$\max_{\pi \in \Pi_H} \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} R_{t+1} \mid S_0 = s \right]$$

$$\text{s.t: } P_\pi \left(\sum_{t=0}^{\infty} D_{t+1} \leq \Delta \mid S_0 = s \right) = 1$$

Outside the usual realm of CMDPs

Π_H : history-dependent policies $\quad h_t = (S_0, A_0, R_1, D_1, \dots, S_t); \quad \pi(a|h_t)$

- Can we find (as in Part I) an optimal **stationary** policy?
- In general, **NO!**



Optimal policy: $V^{\pi_H^*} = \Delta$

The only feasible stationary policy has $V^{\pi_S} = 0$

What if we track the total damage encountered so far?

Current budget & the augmented MDP

- Current budget at time t:

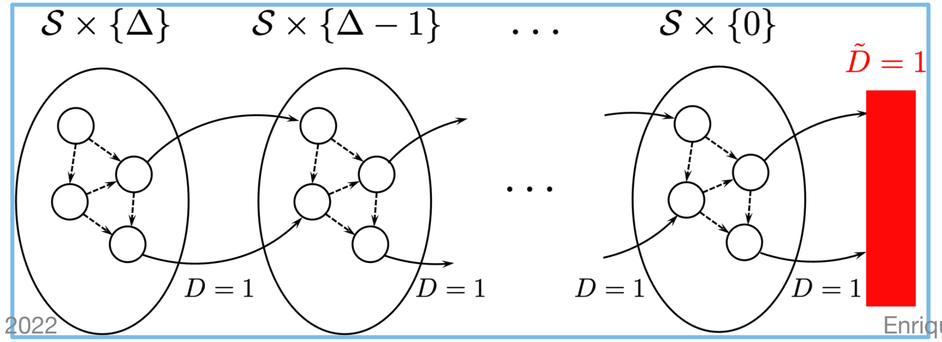
$$K_t = \Delta - \sum_{\ell=0}^{t-1} D_{\ell+1} \quad \forall t \geq 1$$

"How much more damage I can sustain and still be feasible"

Claim: $\exists$ optimal policy $\pi^*(a \mid (s,k))$

- Augmented MDP $\tilde{\mathcal{M}}$

$$\tilde{S}_t = (S_t, K_t)\,, \qquad \tilde{D}_{t+1} = \mathbf{1}\{K_t - D_{t+1} < 0\}\,.$$



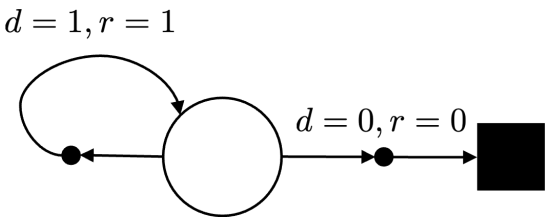
- Equivalent problem:

$$\underbrace{\begin{aligned} &\max_{\tilde{\pi} \in \tilde{\Pi}_H} \mathbb{E}_{\tilde{\pi}, \tilde{\mathcal{M}}} \left[\sum_{t=0}^{\infty} R_{t+1} \,\middle|\, (S_0, K_0) = (s, \Delta) \right] \\ &\text{s.t: } P_{\tilde{\pi}}\left(\tilde{D}_{t+1} = 0\right) = 1 \qquad \forall t \geq 0 \end{aligned}}$$

Fits previous formulation! →

- Could learn $B^*(s,k,a)$
- Separation & Feasibility Principles
- Drawback: working in higher dimensions

Experiment: comparing constraints



Goal

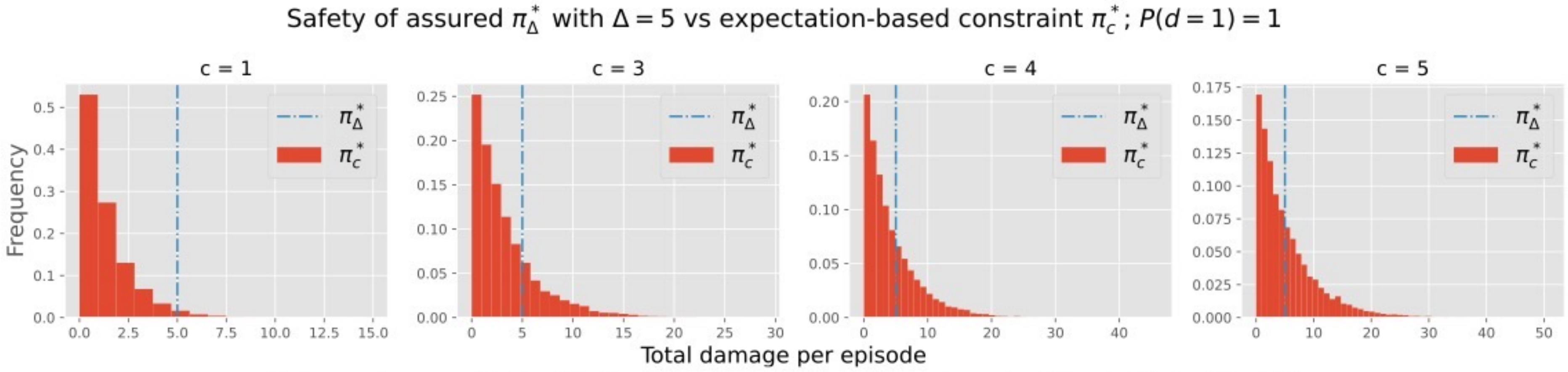
$$\max_{\pi} \quad \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} R_{t+1}\right]$$

1) Proposed constraint

$$\mathbb{P}_{\pi_\Delta}\left(\sum_{t=0}^{\infty} D_{t+1} \leq \Delta \,\middle|\, S_0 = s\right) = 1$$

2) Classic CMDP constraint

$$\mathbb{E}_{\pi_c}\left[\sum_{t=0}^{\infty} D_{t+1}\right] \leq c$$



Summary and future work

Approximations of ROA

- Propose a flexible notion of invariance known as **recurrence.**
- Provide necessary and sufficient conditions for recurrent set to be inner-approximations of ROAs
- Algorithms: sequential, and incur limited number of counter-examples.
- **Future work:** sample complexity, smart choice of multi-points, control recurrent sets

RL with Almost Sure Constraints

- Studied safe/constrained sequential learning:
  - Focus on safety first, show it can be achieved quickly, and with strong guarantees
  - Motivate the need of additional information, *damage*
- Treat constraints separately, or in parallel
- Safety can be learnt more efficiently! and helps learning optimal policies.
- **Future work:** extensions to continue state and action spaces.

Thanks!

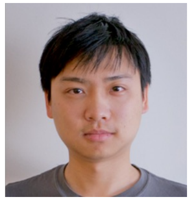
Related Publications:

[arXiv 22] Shen, Bichuch, M, *Model-free Learning of Regions of Attraction via Recurrent Sets,* **submitted to CDC 2022,** preprint arXiv:2204.10372.

[L4DC 22] Castellano, Min, Bazerque, M, *Reinforcement Learning with Almost Sure Constraints,* **Learning for Dynamics and Control (L4DC) Conference, 2022**

[arXiv 21] Castellano, Min, Bazerque, M, *Learning to Act Safely with Limited Exposure and Almost Sure Certainty,* **submitted to IEEE TAC, 2021, under review**, preprint arXiv:2105.08748

Agustin Castellano
JOHNS HOPKINS UNIVERSITY



Hancheng Min
JOHNS HOPKINS UNIVERSITY

Juan Bazerque
UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Enrique Mallada
mallada@jhu.edu
http://mallada.ece.jhu.edu

Yue Shen
JOHNS HOPKINS UNIVERSITY

Maxim Bichuch
JOHNS HOPKINS UNIVERSITY