Nonparametric Analysis and Control of Dynamical Systems

Stability, Safety, and Policy Improvement

Enrique Mallada

JOHNS HOPKINS UNIVERSITY

Shanghai Jiao Tong University

July 4th, 2025

Acknowledgements

Yue Shen
Johns Hopkins University

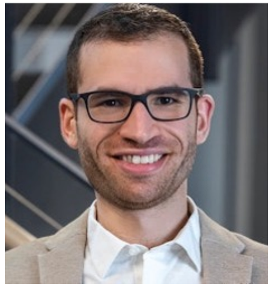
Roy Siegelmann
Johns Hopkins University

Agustin Castellano
Johns Hopkins University

Sohrab Rezaei
Johns Hopkins University

Fernando Paganini
Universidad ORT Uruguay

Maxim Bichuch
University at Buffalo

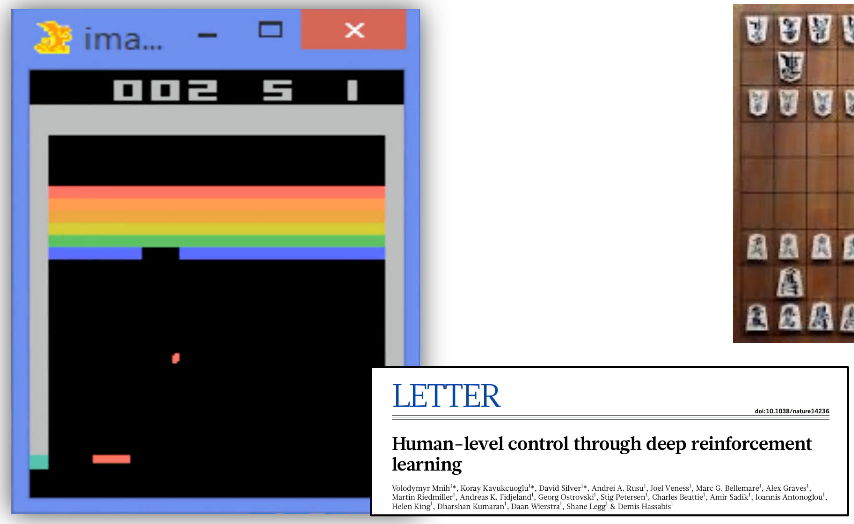


Hussein Sibai
WashU

Jared Markowitz
Johns Hopkins Applied Physics Laboratory

A Dream World of Success Stories

2017 Google DeepMind's DQN



LETTER

doi:10.1038/nature14236

Human-level control through deep reinforcement learning

Volodymyr Mnih[1]*, Koray Kavukcuoglu[1]*, David Silver[1]*, Andrei A. Rusu[1], Joel Veness[1], Marc G. Bellemare[1], Alex Graves[1], Martin Riedmiller[1], Andreas K. Fidjeland[1], Georg Ostrovski[1], Stig Petersen[1], Charles Beattie[1], Amir Sadik[1], Ioannis Antonoglou[1], Helen King[1], Dharshan Kumaran[1], Daan Wierstra[1], Shane Legg[1] & Demis Hassabis[1]

2017 AlphaZero – Chess, Shogi, Go

2019 AlphaStar – Starcraft II

Article

Grandmaster level in StarCraft II using multi-agent reinforcement learning

https://doi.org/10.1038/s41586-019-1724-z

Received: 30 August 2019

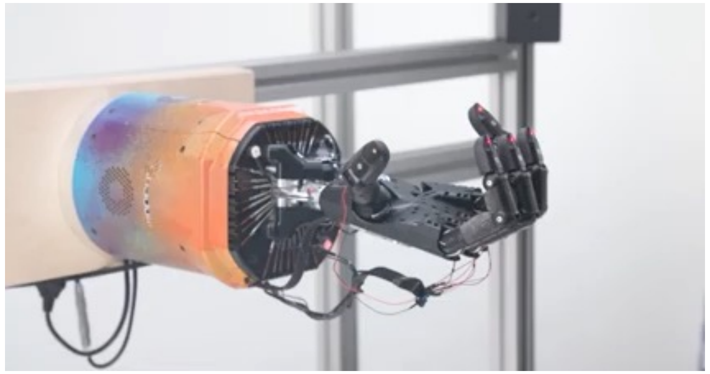
Accepted: 10 October 2019

Published online: 30 October 2019

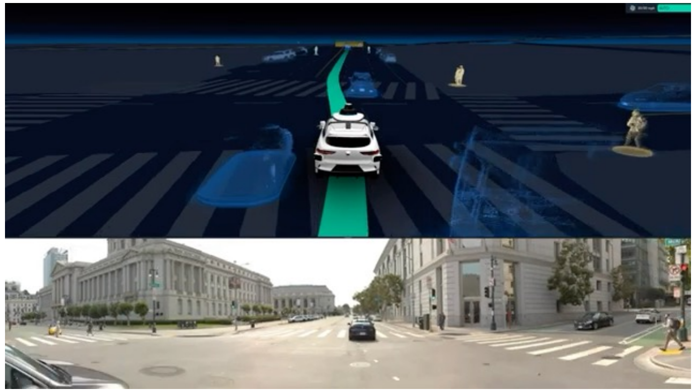
Oriol Vinyals[1,3]*, Igor Babuschkin[1,3], Wojciech M. Czarnecki[1,3], Michaël Mathieu[1,3], Andrew Dudzik[1,3], Junyoung Chung[1,3], David H. Choi[1,3], Richard Powell[1,3], Timo Ewalds[1,3], Petko Georgiev[1,3], Junhyuk Oh[1,3], Dan Horgan[1,3], Manuel Kroiss[1,3], Ivo Danihelka[1,3], Aja Huang[1,3], Laurent Sifre[1,3], Trevor Cai[1,3], John P. Agapiou[1,3], Max Jaderberg[1], Alexander S. Vezhnevets[1], Rémi Leblond[1], Tobias Pohlen[1], Valentin Dalibard[1], David Budden[1], Yury Sulsky[1], James Molloy[1], Tom L. Paine[1], Caglar Gulcehre[1], Ziyu Wang[1], Tobias Pfaff[1], Yuhuai Wu[1], Roman Ring[1], Dani Yogatama[1], Dario Wünsch[2], Katrina McKinney[1], Oliver Smith[1], Tom Schaul[1], Timothy Lillicrap[1], Koray Kavukcuoglu[1], Demis Hassabis[1], Chris Apps[1,3] & David Silver[1,3]*

Boston Dynamics

OpenAI – Rubik's Cube



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.

Tesla Recalls Nearly All Vehicles Due to Autopilot Failures

Tesla disagrees with feds' analysis of glitches

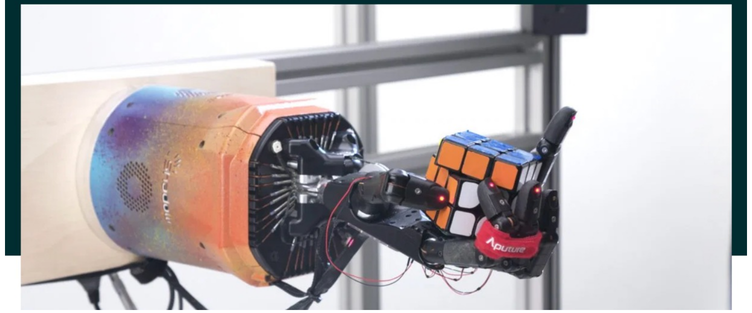
BY LINA FISHER, 2:54PM, WED. DEC. 13, 2023

CRUISE KNEW ITS SELF-DRIVING CARS HAD PROBLEMS RECOGNIZING CHILDREN — AND KEPT THEM ON THE STREETS

According to internal materials reviewed by The Intercept, Cruise cars were also in danger of driving into holes in the road.

OpenAI disbands its robotics research team

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



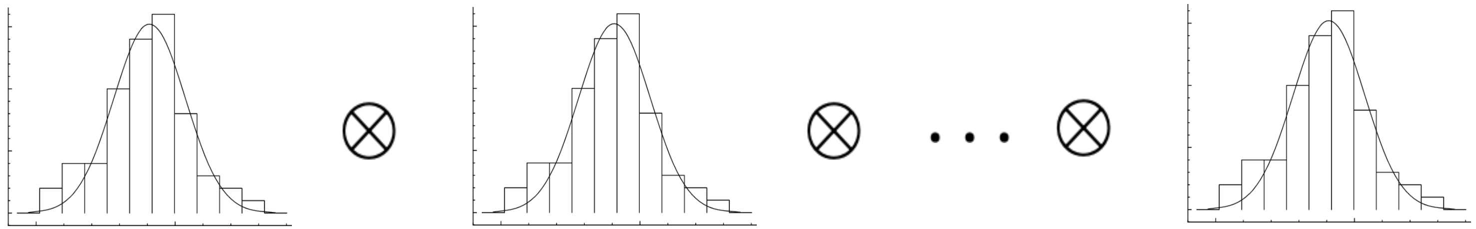
Self-driving Uber car that hit and killed woman did not recognize that pedestrians jaywalk

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

Fundamental challenge: The curse of dimensionality

- **Statistical: No clear inductive bias**

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.

Atoms in the universe: 10^{78}

- **Computational: Verifying non-negativity of polynomials**

Copositive matrices:

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

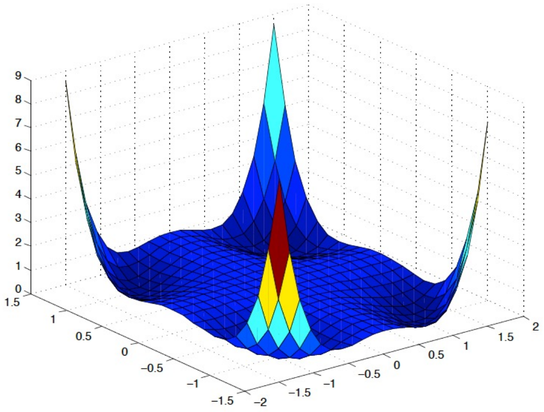
Murty&Kadabi [1987]: Testing co-positivity is NP-Hard

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]:

$p = x^4y^2 + x^2y^4 + 1 - 3x^2y^2$

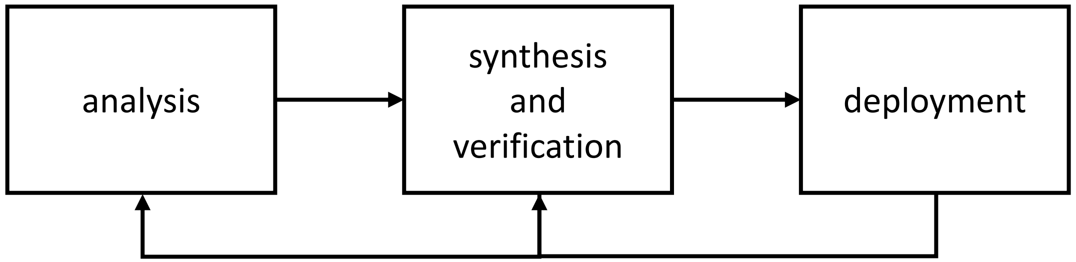
is nonnegative,

not a sum of squares,

but $(x^2 + y^2)^2 p$ is SoS

Methodological challenges

- Focused on a ***design-then-deploy*** philosophy
  - Most methods have a strict separation between control synthesis and deployment



- Synthesis usually aims for the ***best*** (optimal) controller
  - Lack of exploration of the benefits of designing sub-optimal controllers

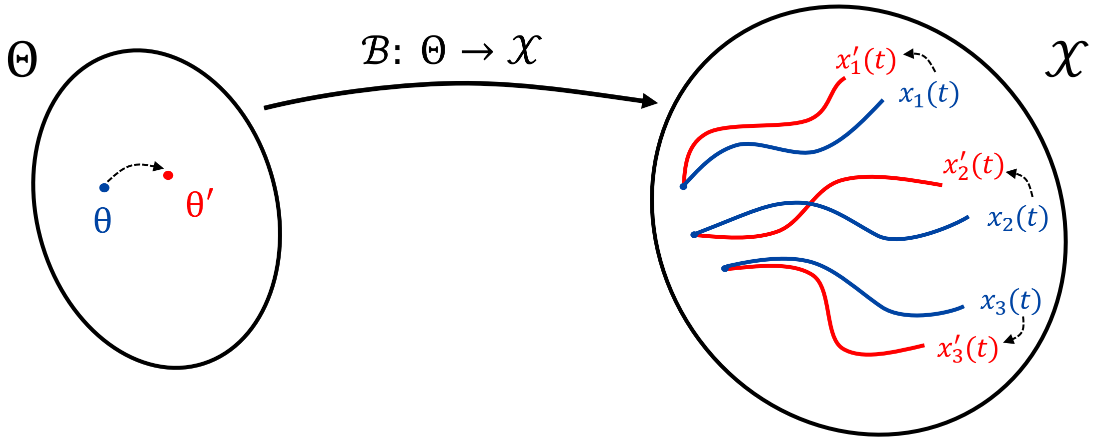
RL:

$$\max_{\pi} \quad J(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$
$$\text{s.t.} \quad s_{t+1} \sim P(\cdot \mid s_t, a_t), \quad a_t \sim \pi(\cdot \mid s_t)$$

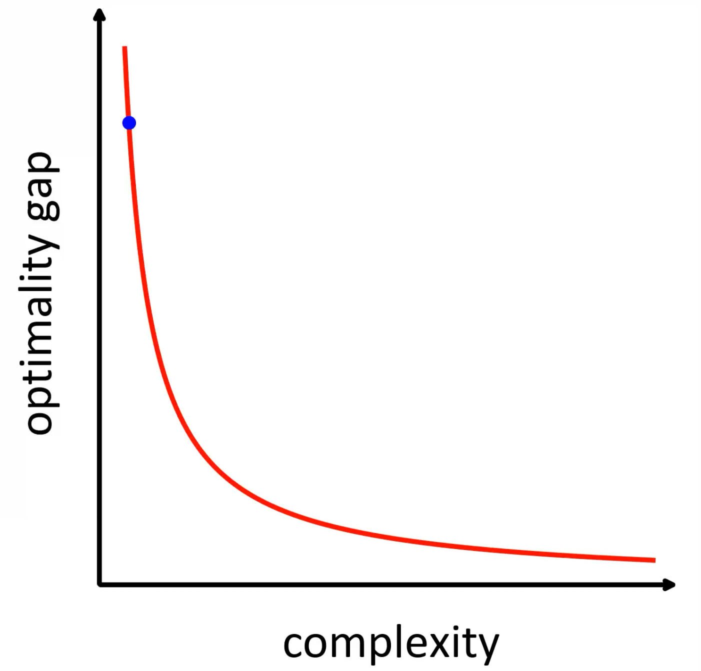
Optimal Control:

$$\min_{u(\cdot)} \quad J = \int_0^T L(x(t), u(t), t)\, dt + \Phi(x(T))$$
$$\text{s.t.} \quad \dot{x}(t) = f(x(t), u(t), t), \quad x(0) = x_0$$

- Policy ***parameters*** can ***drastically affect*** the system's ***behavior***
  - The params to behavior maps are highly sensitive to perturbations



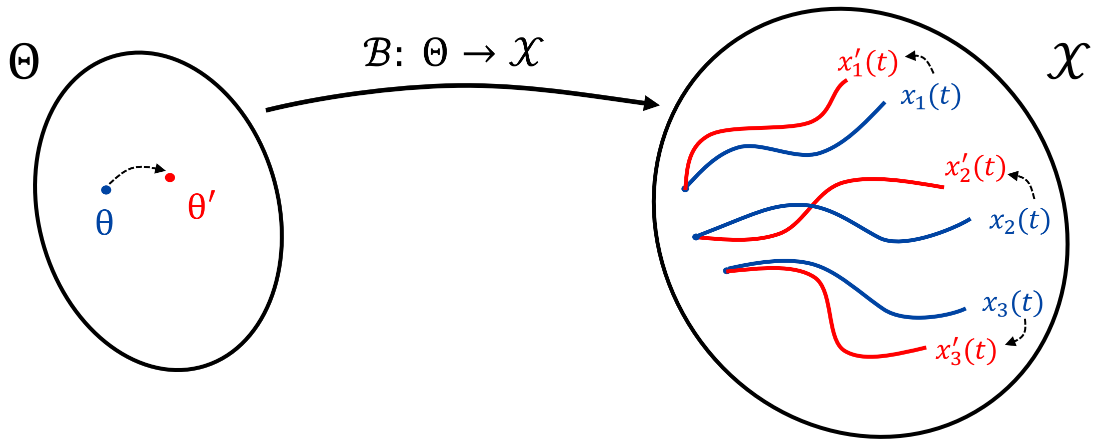
Research Goals



- To develop analysis and design methods that *trade off complexity and performance*.

- To allow for *continual improvement*, without the need for redesign, retune, or retrain

- To design control policies with controlled sensitivity to parameter changes



Outline

- **Relaxing Invariance: Merits and trade offs**
  - *Recurrent Sets:* Letting thing go and come back
- **Nonparametric Analysis via Recurrent Sets**
  - *Stability analysis:* Recurrent Lyapunov Functions (RLFs)
  - *Safety verification:* Recurrent Barrier Functions (RBFs)
- **Self-Improving via Nonparametric Control Policies**
  - Policy Improvement using Expert Demonstrations

Outline

- **Relaxing Invariance: Merits and trade offs**
  - *Recurrent Sets:* Letting thing go and come back
- **Nonparametric Analysis via Recurrent Sets**
  - *Stability analysis:* Recurrent Lyapunov Functions (RLFs)
  - *Safety verification:* Recurrent Barrier Functions (RBFs)
- **Self-Improving via Nonparametric Control Policies**
  - Policy Improvement using Expert Demonstrations

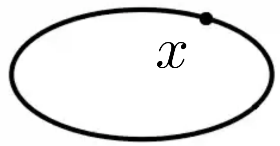
Problem setup

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

- Initial condition $x_0 = x(0)$, solution at time t: $\phi(t, x_0)$.

Asymptotic behavior: Ω-Limit Set $\Omega(f)$

$$x \in \Omega(f) \iff \exists\, x_0, \{t_n\}_{n\geq 0}, \text{ s.t. } \lim_{n\to\infty} t_n = \infty \text{ and } \lim_{n\to\infty} \phi(t_n, x_0) = x$$



Problem setup

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

- Initial condition $x_0 = x(0)$, solution at time t: $\phi(t, x_0)$.

Ω-Limit Set $\Omega(f)$:

$$x \in \Omega(f) \iff \exists\, x_0, \{t_n\}_{n\geq 0}, \text{ s.t. } \lim_{n\to\infty} t_n = \infty \text{ and } \lim_{n\to\infty} \phi(t_n, x_0) = x$$

Types of Ω-limit set

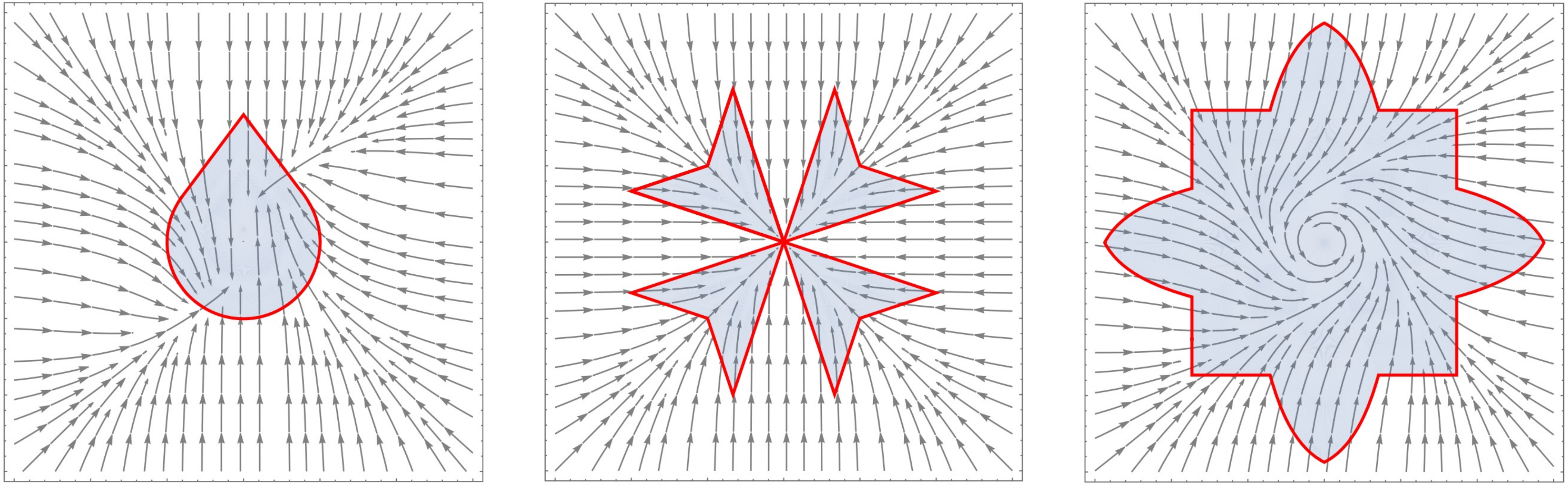
equilibrium | limit cycle | limit torus | chaotic attractor

Remark: invariance is a shared property, thus a natural tool for analysis

Invariant sets

A set $\mathcal{S} \subseteq \mathbb{R}^d$ is **positively invariant** if and only if: $x_0 \in \mathcal{S} \rightarrow \phi(t, x_0) \in \mathcal{S},\ \forall t \geq 0$

Any trajectory starting in the set remains in inside it for all times



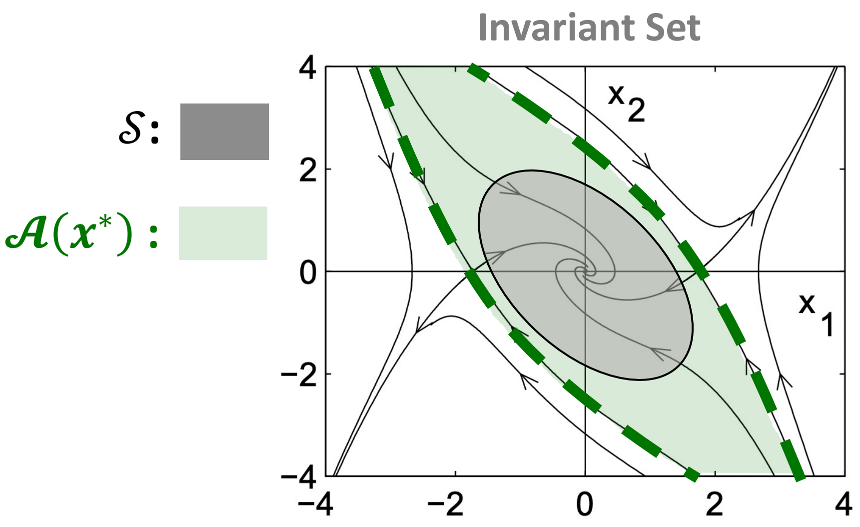
Source: K. Ghorbal, K. and A. Sogokon, Characterizing positively invariant sets: Inductive and topological methods. Journal of Symbolic Computation, 2022

Invariant sets: Merits

A set $\mathcal{S} \subseteq \mathbb{R}^d$ is **positively invariant** if and only if: $x_0 \in \mathcal{S} \to \phi(t, x_0) \in \mathcal{S},\ \forall t \geq 0$

Any trajectory starting in the set remains in inside it for all times

- **Invariant sets approximate regions of attraction**
  Compact invariant set $\mathcal{S}$, containing **only** $\{x^*\} = \Omega(f) \cap \mathcal{S}$ must be in the region of attraction $\mathcal{A}(x^*)$ ($\mathcal{S} \subset \mathcal{A}(x^*)$)



Invariant sets: Merits

A set $\mathcal{S} \subseteq \mathbb{R}^d$ is **positively invariant** if and only if: $x_0 \in \mathcal{S} \rightarrow \phi(t, x_0) \in \mathcal{S},\ \forall t \geq 0$

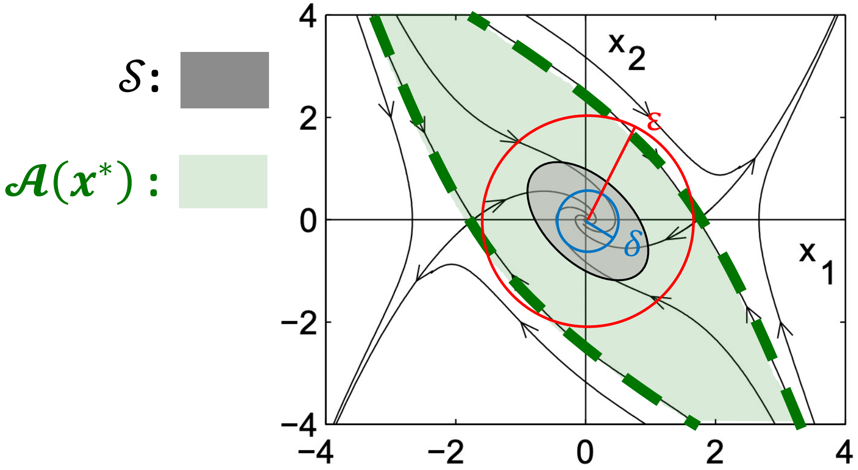
Any trajectory starting in the set remains in inside it for all times

- **Invariant sets approximate regions of attraction**
  Compact invariant set $\mathcal{S}$, containing **only** $\{x^*\} = \Omega(f) \cap \mathcal{S}$ must be in the region of attraction $\mathcal{A}(x^*)$ ($\mathcal{S} \subset \mathcal{A}(x^*)$)

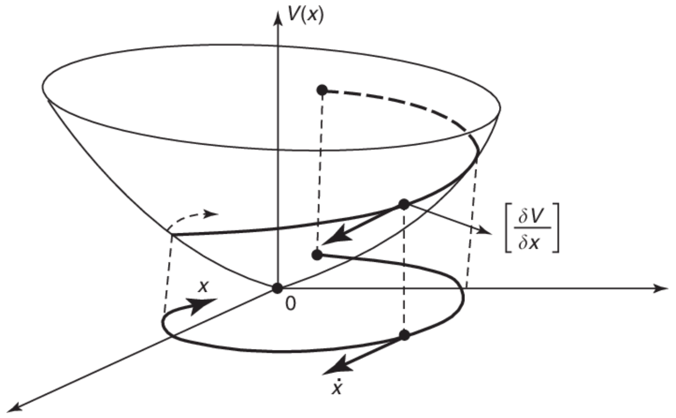
- **Invariant sets guarantee stability**
  Lyapunov stability: solutions starting "close enough" to the equilibrium (within a distance δ) remain "close enough" forever (within a distance ε)

- **Invariant sets further certify asymptotic stability via Lyapunov's direct method**
  Asymptotic stability: solutions that start close enough, remain close enough, and eventually converge to equilibrium.

Invariant Set



Lyapunov Functions



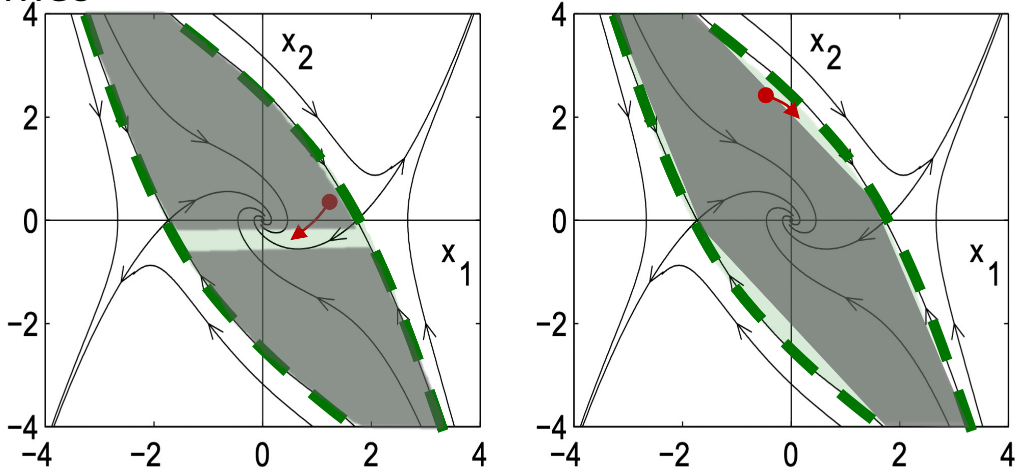
Invariant sets: Challenges

A set $\mathcal{S} \subseteq \mathbb{R}^d$ is **positively invariant** if and only if: $x_0 \in \mathcal{S} \to \phi(t, x_0) \in \mathcal{S},\ \forall t \geq 0$

Any trajectory starting in the set remains in inside it for all times

$\mathcal{S}$:

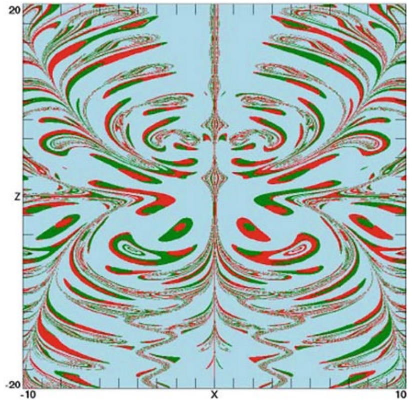
$\mathcal{A}(x^*)$:

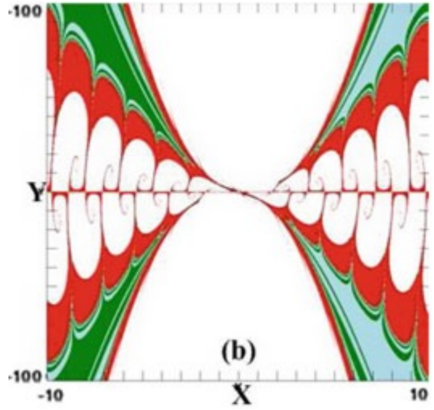


A not invariant trajectory:

- **$\mathcal{S}$ is topologically constrained**
  - If $\mathcal{S} \cap \Omega(f) = \{x^*\}$, then $\mathcal{S}$ is connected
- **$\mathcal{S}$ is geometrically constrained**
  - f should not point outwards for $x \in \partial\mathcal{S}$
- **$\mathcal{S}$ geometry can be wild**
  - $\mathcal{A}(\Omega(f))$ can be fractal

Basin of $\Omega(f)$





Outline

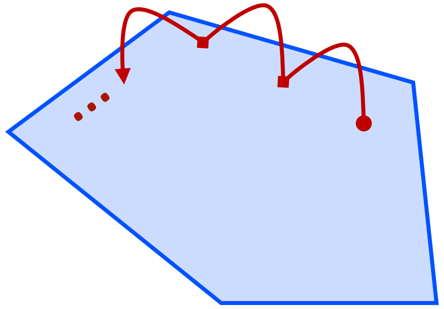
- **Relaxing Invariance: Merits and trade offs**
  - *Recurrent Sets:* Letting thing go and come back
- **Nonparametric Analysis via Recurrent Sets**
  - *Stability analysis:* Recurrent Lyapunov Functions (RLFs)
  - *Safety verification:* Recurrent Barrier Functions (RBFs)
- **Self-Improving via Nonparametric Control Policies**
  - Policy Improvement using Expert Demonstrations

Outline

- **Relaxing Invariance: Merits and trade offs**
  - *Recurrent Sets:* Letting thing go and come back
- **Nonparametric Analysis via Recurrent Sets**
  - *Stability analysis:* Recurrent Lyapunov Functions (RLFs)
  - *Safety verification:* Recurrent Barrier Functions (RBFs)
- **Self-Improving via Nonparametric Control Policies**
  - Policy Improvement using Expert Demonstrations

Recurrent sets: Letting things go, and come back

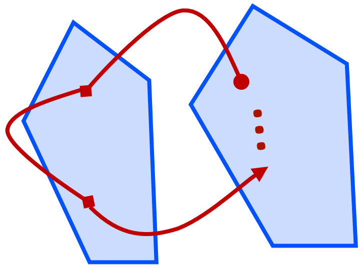
A set $\mathcal{R} \subseteq \mathbb{R}^d$ is **recurrent** if for any $x_0 \in \mathcal{R}$ and $t \geq 0$, $\exists t' \geq t$ s.t. $\phi(t', x_0) \in \mathcal{R}$.



Property of Recurrent Sets

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

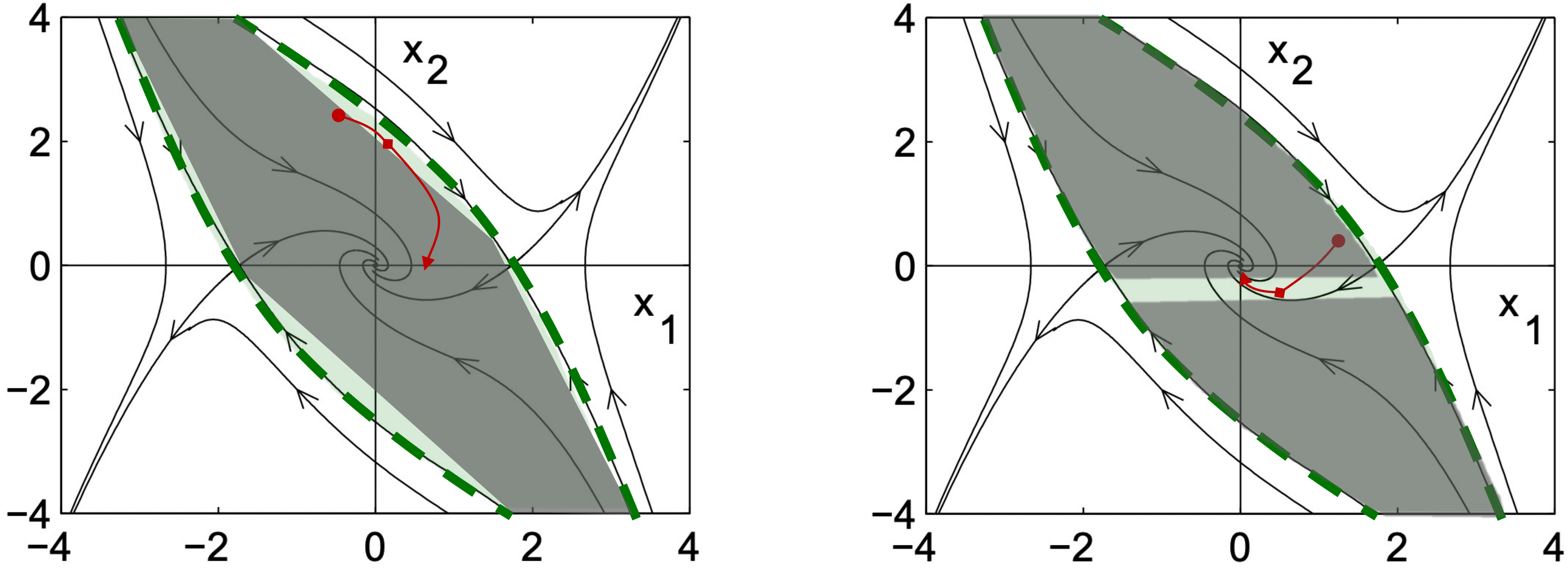
> Recurrent sets, while not invariant, guarantee that solutions that start in this set, will come back **infinitely often, forever!**



Recurrent sets: Letting things go, and come back

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

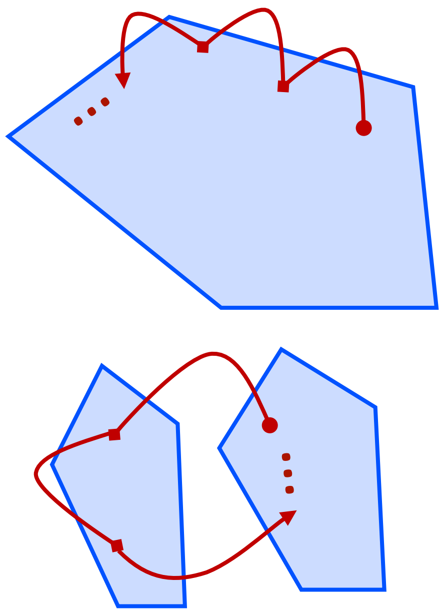
Previous two good inner approximations of $\mathcal{A}(x^*)$ are recurrent sets



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

Recurrent sets: Letting things go, and come back

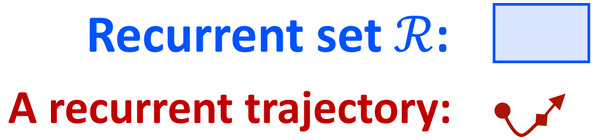
A set $\mathcal{R} \subseteq \mathbb{R}^d$ is **recurrent** if for any $x_0 \in \mathcal{R}$ and $t \geq 0$, $\exists t' \geq t$ s.t. $\phi(t', x_0) \in \mathcal{R}$.



Property of Recurrent Sets

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

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



Question: Can we use recurrent sets as functional substitutes of invariant sets?

Outline

- **Relaxing Invariance: Merits and trade offs**
  - *Recurrent Sets:* Letting thing go and come back
- **Nonparametric Analysis via Recurrent Sets**
  - *Stability analysis:* Recurrent Lyapunov Functions (RLFs)
  - *Safety verification:* Recurrent Barrier Functions (RBFs)
- **Self-Improving via Nonparametric Control Policies**
  - Policy Improvement using Expert Demonstrations

Roy Siegelmann

Yue Shen

Fernando Paganini

Nonparametric Stability Analysis

R. Siegelmann, Y. Shen, F. Paganini, and E. Mallada, “A recurrence-based direct method for stability analysis and GPU-based verification of non-monotonic Lyapunov functions”, CDC 2023

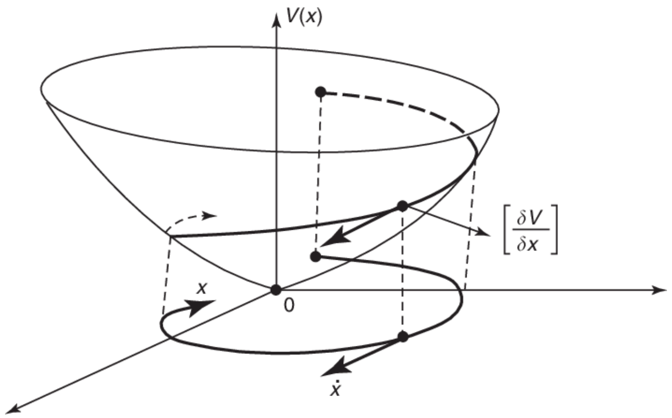
R. Siegelmann, Y. Shen, F. Paganini, and E. Mallada, “Recurrent Lyapunov Functions”, TAC 2025, submitted

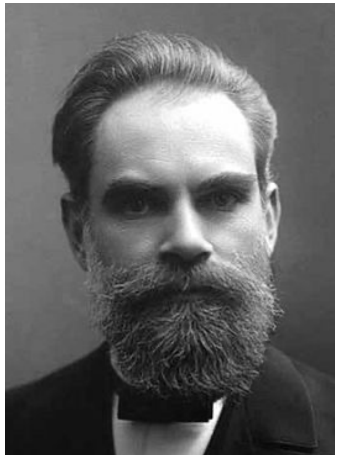
Lyapunov's Direct Method

Key idea: Make sub-level sets invariant to trap trajectories

Theorem [Lyapunov '1892]. *Given* $V: \mathbb{R}^d \to \mathbb{R}_{\geq 0}$, *with* $V(x) > 0, \forall x \in \mathbb{R}^d \backslash \{x^*\}$, *then:*

- $\dot{V} \leq 0 \to x^*$ *stable*
- $\dot{V} < 0 \to x^*$ *as. stable*





Challenge: Couples shape of V and vector field f

- Towards decoupling the $V - f$ geometry
  - Controlling regions where $\dot{V} \geq 0$ **[Karafyllis '09, Liu et al '20]**
  - Higher order conditions: $g(V^{(q)}, \dots, \dot{V}, V) \leq 0$ **[Butz '69, Gunderson '71, Ahmadi '06, Meigoli '12]**
  - Discretization approach: $V\big(x(T)\big) \leq V(x(0))$ **[Coron et al '94, Aeyels et. al '98, Karafyllis '12]**
  - Multiple Lyapunov Functions: $\{V_j : j \in [k]\}$ **[Ahmadi et al '14]**

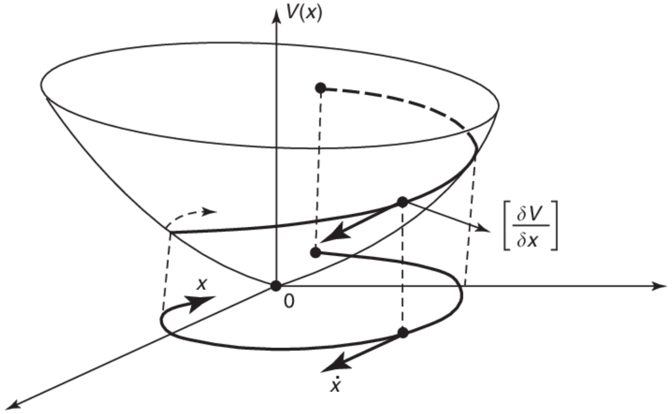
A Butz. Higher order derivatives of Lyapunov functions. IEEE Transactions on automatic control, 1969
Gunderson. A comparison lemma for higher order trajectory derivatives. Proceedings of the American Mathematical Society, 1971
Coron, Lionel Rosier. A relation between continuous time-varying and discontinuous feedback stabilization. J. Math. Syst., Estimation, Control, 1994
Aeyels, Peuteman. A new asymptotic stability criterion for nonlinear time-variant differential equations. IEEE Transactions on automatic control, 1998
Ahmadi. Non-monotonic Lyapunov functions for stability of nonlinear and switched systems: theory and computation, 2008
Karafyllis, Kravaris, Kalogerakis. Relaxed Lyapunov criteria for robust global stabilisation of non-linear systems. International Journal of Control, 2009
Meigoli, Nikravesh. Stability analysis of nonlinear systems using higher order derivatives of Lyapunov function candidates. Systems & Control Letters, 2012
Karafyllis. Can we prove stability by using a positive definite function with non sign-definite derivative? IMA Journal of Mathematical Control and Information, 2012
Ahmadi, Jungers, Parrilo, Roozbehani. Joint spectral radius and path-complete graph Lyapunov functions. SIAM Journal on Control and Optimization, 2014
Liu, Liberzon, Zharnitsky. Almost Lyapunov functions for nonlinear systems. Automatica, 2020

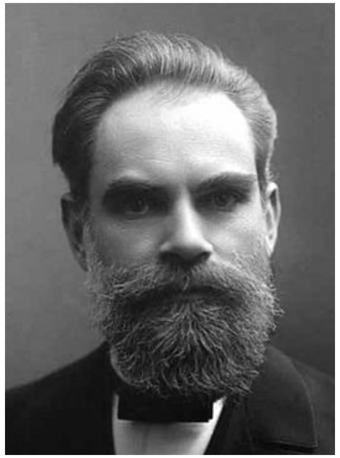
Lyapunov's Direct Method

Key idea: Make sub-level sets invariant to trap trajectories

Theorem [Lyapunov '1892]*. Given* $V: \mathbb{R}^d \rightarrow \mathbb{R}_{\geq 0}$*, with* $V(x) > 0, \forall x \in \mathbb{R}^d \backslash \{x^*\}$*, then:*

- $\dot{V} \leq 0 \rightarrow x^*$ *stable*
- $\dot{V} < 0 \rightarrow x^*$ *as. stable*





Challenge: Couples shape of V and vector field f

- Towards decoupling the $V - f$ geometry
  - Controlling regions where $\dot{V} \geq 0$ **[Karafyllis '09, Liu et al '20]**
  - Higher order conditions: $g(V^{(q)}, \ldots, \dot{V}, V) \leq 0$ **[Butz '69, Gunderson '71, Ahmadi '06, Meigoli '12]**
  - Discretization approach: $V\big(x(T)\big) \leq V(x(0))$ **[Coron et al '94, Aeyels et. al '98, Karafyllis '12]**
  - Multiple Lyapunov Functions: $\{V_j : j \in [k]\}$ **[Ahmadi et al '14]**

Question: Can we provide stability conditions based on recurrence?

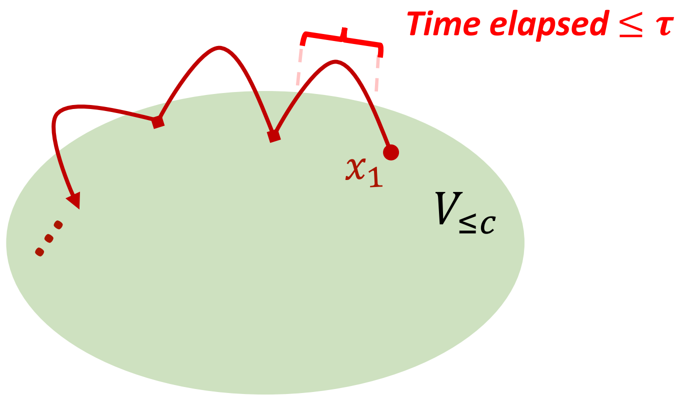
Recurrent Lyapunov Functions

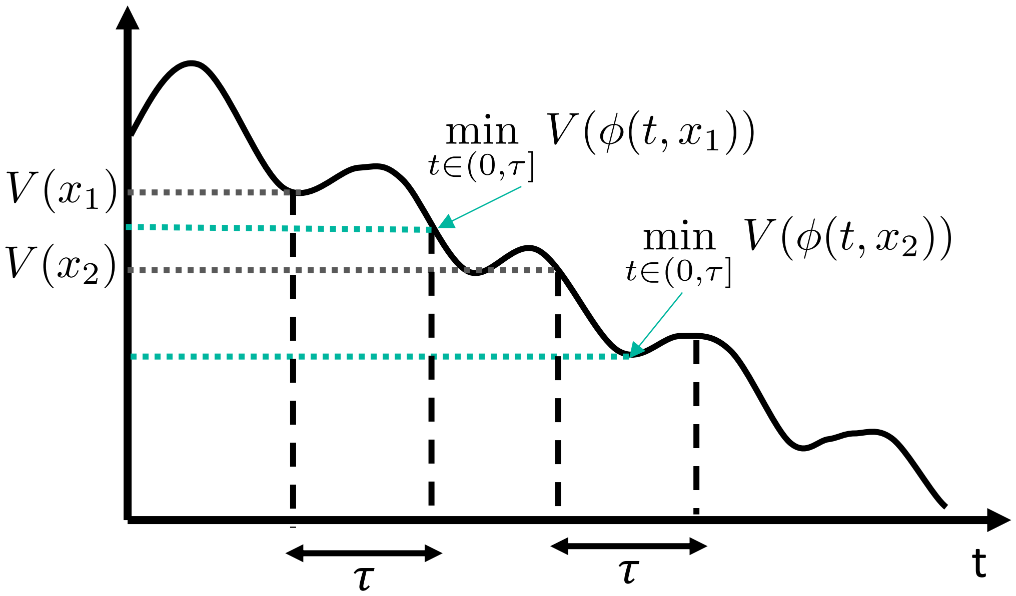
A continuous function $V\colon \mathbb{R}^d \to \mathbb{R}_+$ is a **Recurrent Lyapunov Function** if

$$\mathcal{L}_f^{(0,\tau]} V(x) := \min_{t\in(0,\tau]} V(\phi(t,x)) - V(x) \leq 0 \quad \forall x \in \mathbb{R}^d$$

Preliminaries:

- Sub-level sets $\{V(x) \leq c\}$ are $\boldsymbol{\tau}$-recurrent sets.





Definition: A set $\mathcal{R} \subseteq \mathbb{R}^d$ is **τ-recurrent** if for any $x_0 \in \mathcal{R}$ and $t \geq 0$, $\exists t' \in (t, t+\tau]$ s.t. $\phi(t', x_0) \in \mathcal{R}$.

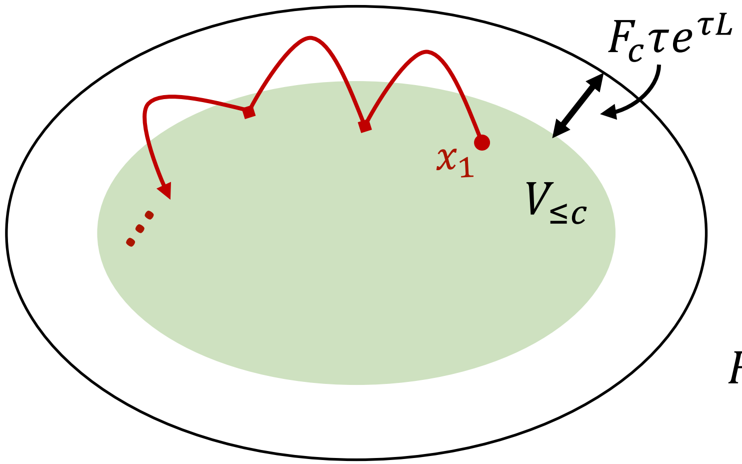
Recurrent Lyapunov Functions

A continuous function $V\colon \mathbb{R}^d \to \mathbb{R}_+$ is a **Recurrent Lyapunov Function** if

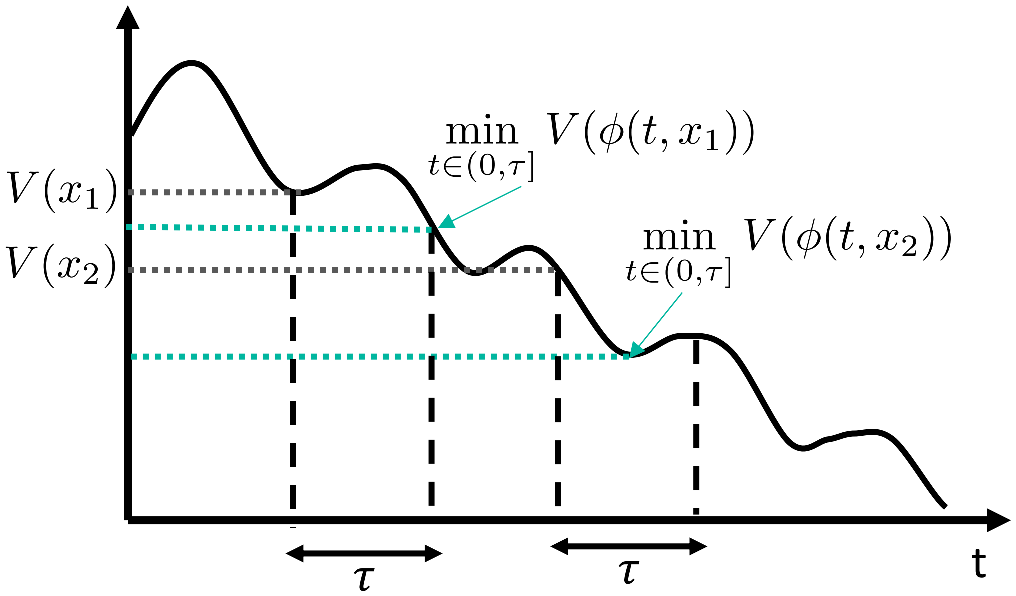
$$\mathcal{L}_f^{(0,\tau]} V(x) := \min_{t\in(0,\tau]} V(\phi(t,x)) - V(x) \leq 0 \quad \forall x \in \mathbb{R}^d$$

Preliminaries:

- Sub-level sets $\{V(x) \leq c\}$ are $\boldsymbol{\tau}$-recurrent sets.
- When f is L-Lipschitz, one can trap trajectories.



$F_c = \max_{x\in V_{\leq c}} ||f(x)||$



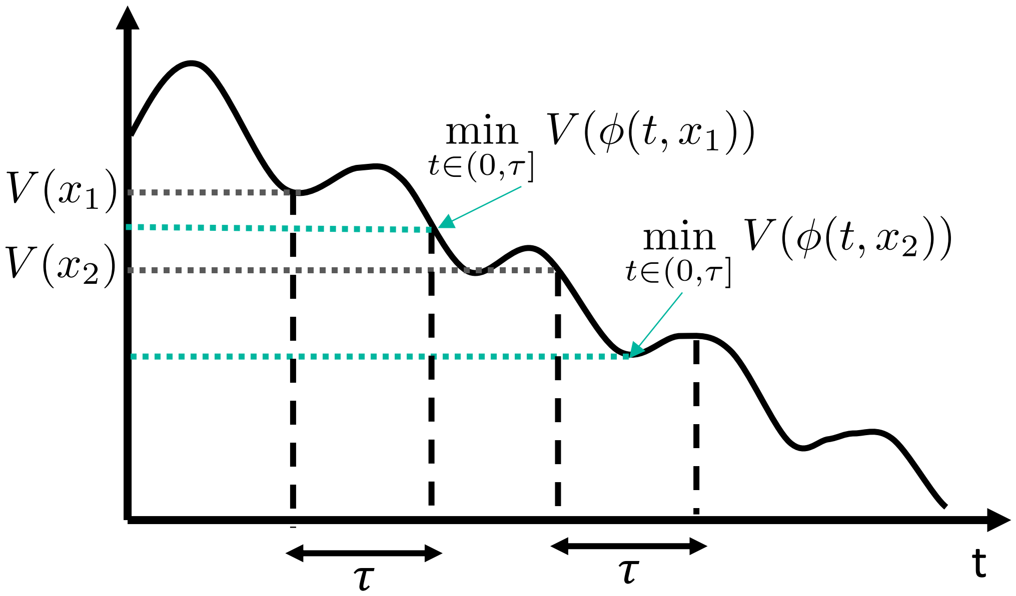
Recurrent Lyapunov Functions

A continuous function $V\colon \mathbb{R}^d \to \mathbb{R}_+$ is a **Recurrent Lyapunov Function** if

$$\mathcal{L}_f^{(0,\tau]}V(x) := \min_{t\in(0,\tau]} V(\phi(t,x)) - V(x) \leq 0 \quad \forall x \in \mathbb{R}^d$$

Theorem [CDC 23]: *Let* $V\colon \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ *be a Recurrent Lyapunov Function and let* f *be L-Lipschitz*

- *Then, the equilibrium* x^* *is stable.*



Siegelmann, Shen, Paganini, M, *A recurrence-based direct method for stability analysis and GPU-based verification of non-monotonic Lyapunov functions*, **CDC 2023**

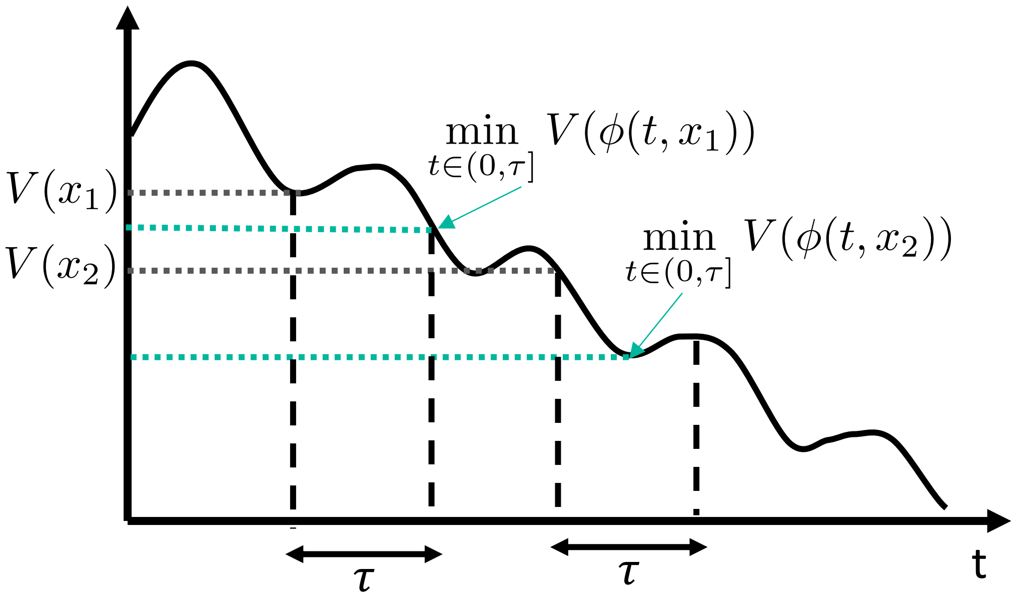
Recurrent Lyapunov Functions

A continuous function $V\colon \mathbb{R}^d \to \mathbb{R}_+$ is a **Recurrent Lyapunov Function** if

$$\mathcal{L}_f^{(0,\tau]} V(x) := \min_{t\in(0,\tau]} V(\phi(t,x)) - V(x) < 0 \quad \forall x \in \mathbb{R}^d$$

Theorem [CDC 23]: *Let $V\colon \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ be a Recurrent Lyapunov Function and let f be L-Lipschitz*

- *Then, the equilibrium x^* is stable.*
- *Further, if the **inequality is strict**, then x^* is asymptotically stable!*



Siegelmann, Shen, Paganini, M, *A recurrence-based direct method for stability analysis and GPU-based verification of non-monotonic Lyapunov functions*, **CDC 2023**

Exponential Stability Analysis

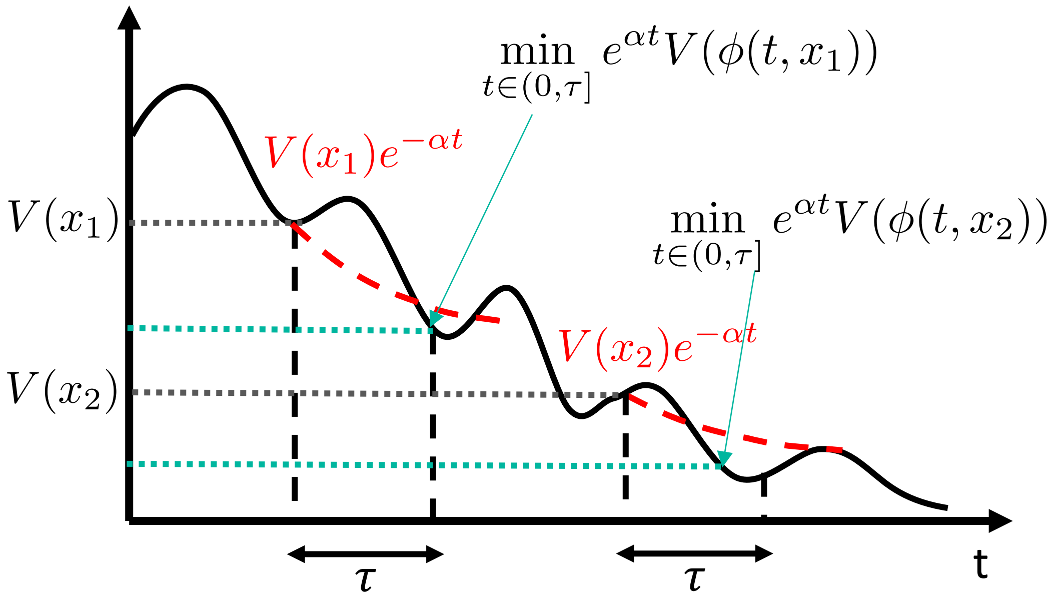
The function $V\colon \mathbb{R}^d \to \mathbb{R}_+$ is ***α*-Exponential Recurrent Lyapunov Function** if

$$\mathcal{L}_{f,\alpha}^{(0,\tau]} V(x) := \min_{t\in(0,\tau]} e^{\alpha t} V(\phi(t,x)) - V(x) \le 0 \quad \forall x \in \mathbb{R}^d$$

Theorem [CDC 23]: *Let* $V\colon \mathbb{R}^d \to \mathbb{R}_{\ge 0}$ *satisfy*

$$\alpha_1 \|x - x^*\| \le V(x) \le \alpha_2 \|x - x^*\|.$$

Then, if V *is α-Exponential Recurrent Lyapunov Function,* x^* *is α-exponentially stable.*



Siegelmann, Shen, Paganini, M, *A recurrence-based direct method for stability analysis and GPU-based verification of non-monotonic Lyapunov functions,* **CDC 2023**

Norm-based Converse Theorem

Theorem: Assume x^* is λ-exponentially stable: $\exists\, K, \lambda > 0$ such that:

$$\left|\left|\phi(t,x) - x^*\right|\right| \leq K e^{-\lambda t}\left|\left|x - x^*\right|\right|, \qquad \forall x \in \mathbb{R}^d.$$

Then, $V(x) = \left|\left|x - x^*\right|\right|$ *is* α*-Exponential Recurrent Lyapunov Function , i.e.,*

$$\min_{t\in(0,\tau]} e^{\alpha t}\left|\left|\phi(t,x) - x^*\right|\right| - \left|\left|x - x^*\right|\right| \leq 0, \qquad \forall x \in \mathbb{R}^d,$$

whenever $\alpha < \lambda$ *and* $\tau \geq \frac{1}{\lambda - \alpha}\ln K.$

Remarks:

- The rate α must be strictly smaller than the rate of convergence λ (trading off optimality).
- *Any norm is a Lyapunov function!*

Question: Is the struggle for its search over?

Nonparametric Verification of Exponential Stability

Proposition [CDC 23]: *Let* $||\cdot||$ *be any norm and* $x^* = 0$*. Then, whenever*

$$\min_{t\in(0,\tau]} e^{\alpha t}\left(||\phi(x,t)|| + re^{Lt}\right) \le ||x|| - r$$

for all y *with* $||y - x|| \le r$

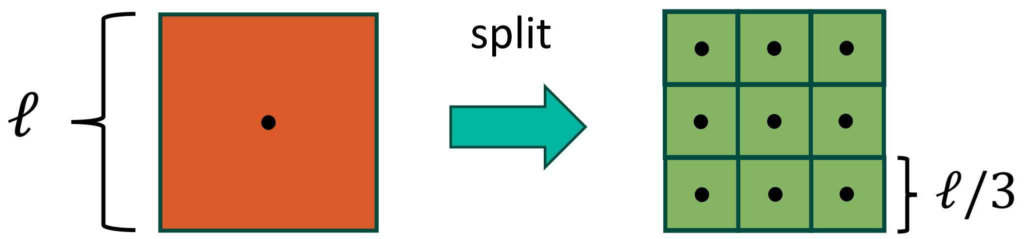
$$\min_{t\in(0,\tau]} e^{\alpha t}||\phi(y,t)|| \le ||y||$$

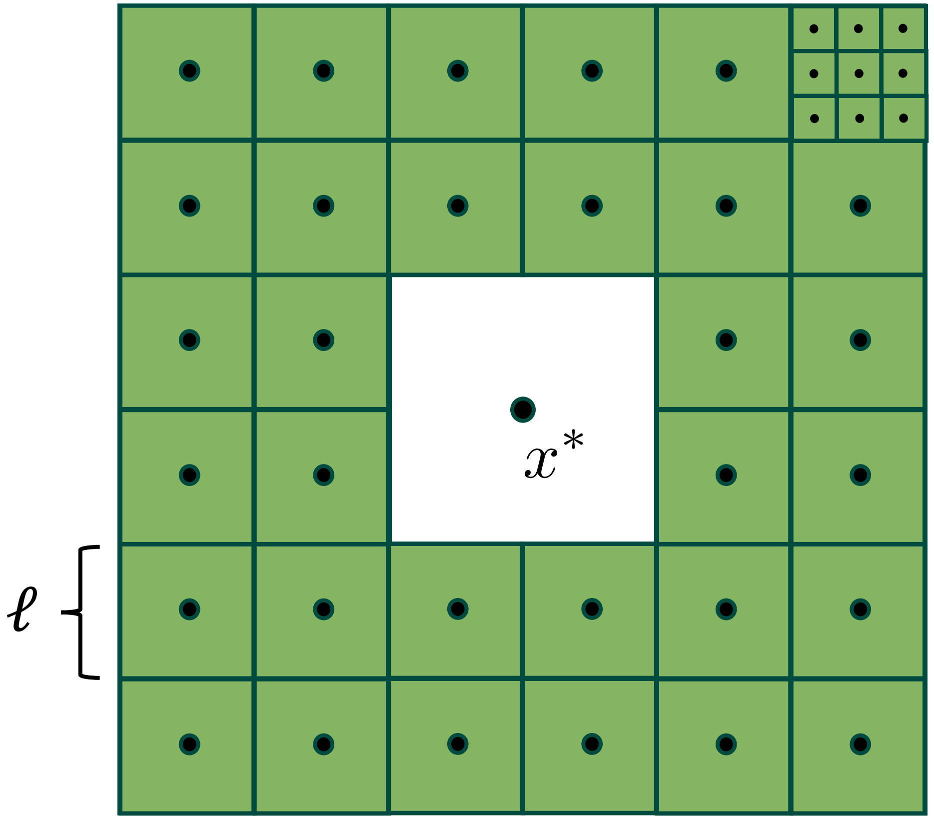
Remarks:

- Only requires a trajectory of length τ
- Trades off between **radius** $\boldsymbol{r}$ and verified **performance** $\boldsymbol{\alpha}$
- Amenable for parallel computations **using GPUs**

Nonparametric Stability Verification via GPUs

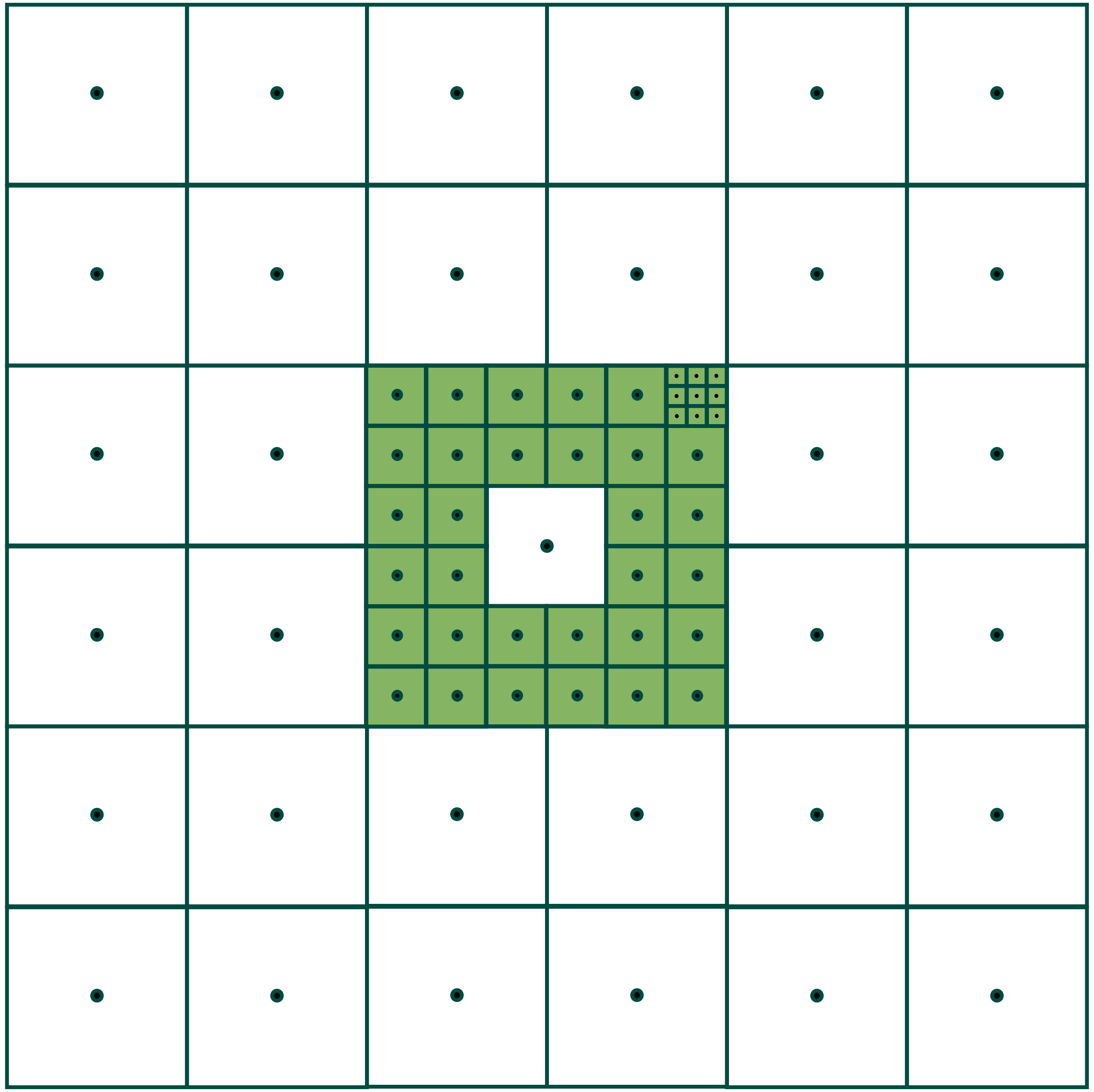
- **Basic Algorithm:**
  - Consider $V(x) = ||x - x^*||_\infty$
  - Build a grid of hypercubes surrounding x^*
  - Test grid center points:
    - Simulate trajectories of length τ
    - Find α s.t. the verified radius is $r \geq \ell/2$
  - Hypercube **not verified**, **split in 3^d** parts
  - Repeat testing of new points





Nonparametric Stability Verification via GPUs

- **Basic Algorithm:**
  - Consider $V(x) = ||x - x^*||_\infty$
  - Build a grid of hypercubes surrounding x^*
  - Test grid center points:
    - Simulate trajectories of length τ
    - Find α s.t. the verified radius is $r \geq \ell/2$
  - Hypercube **not verified**, **split in $\mathbf{3^d}$** parts
  - Repeat testing of new points
  - **Exponentially expand** to outer layer
  - Repeat testing in new layer



Nonparametric Stability Verification via GPUs

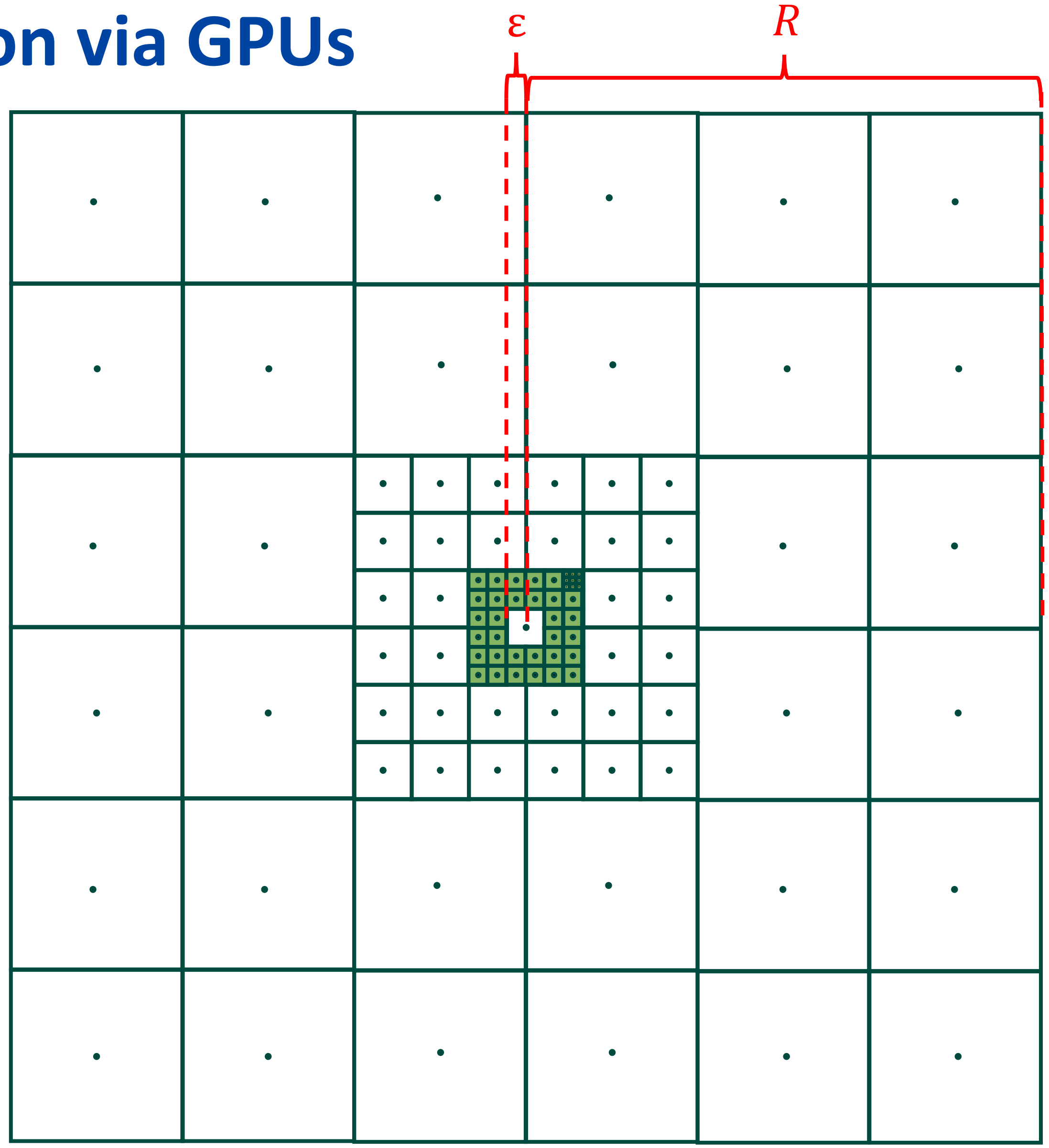
- **Basic Algorithm:**
  - Consider $V(x) = ||x - x^*||_\infty$
  - Build a grid of hypercubes surrounding x^*
  - Test grid center points:
    - Simulate trajectories of length τ
    - Find α s.t. the verified radius is $r \geq \ell/2$
  - Hypercube **not verified**, **split in 3^d** parts
  - Repeat testing of new points
  - **Exponentially expand** to outer layer
  - Repeat testing in new layer

Q: How many samples are needed?

If x^* is λ-exp. stable

$$\mathcal{O}\left(q^{-d}\log\left(\frac{R}{\varepsilon}\right)\right)$$

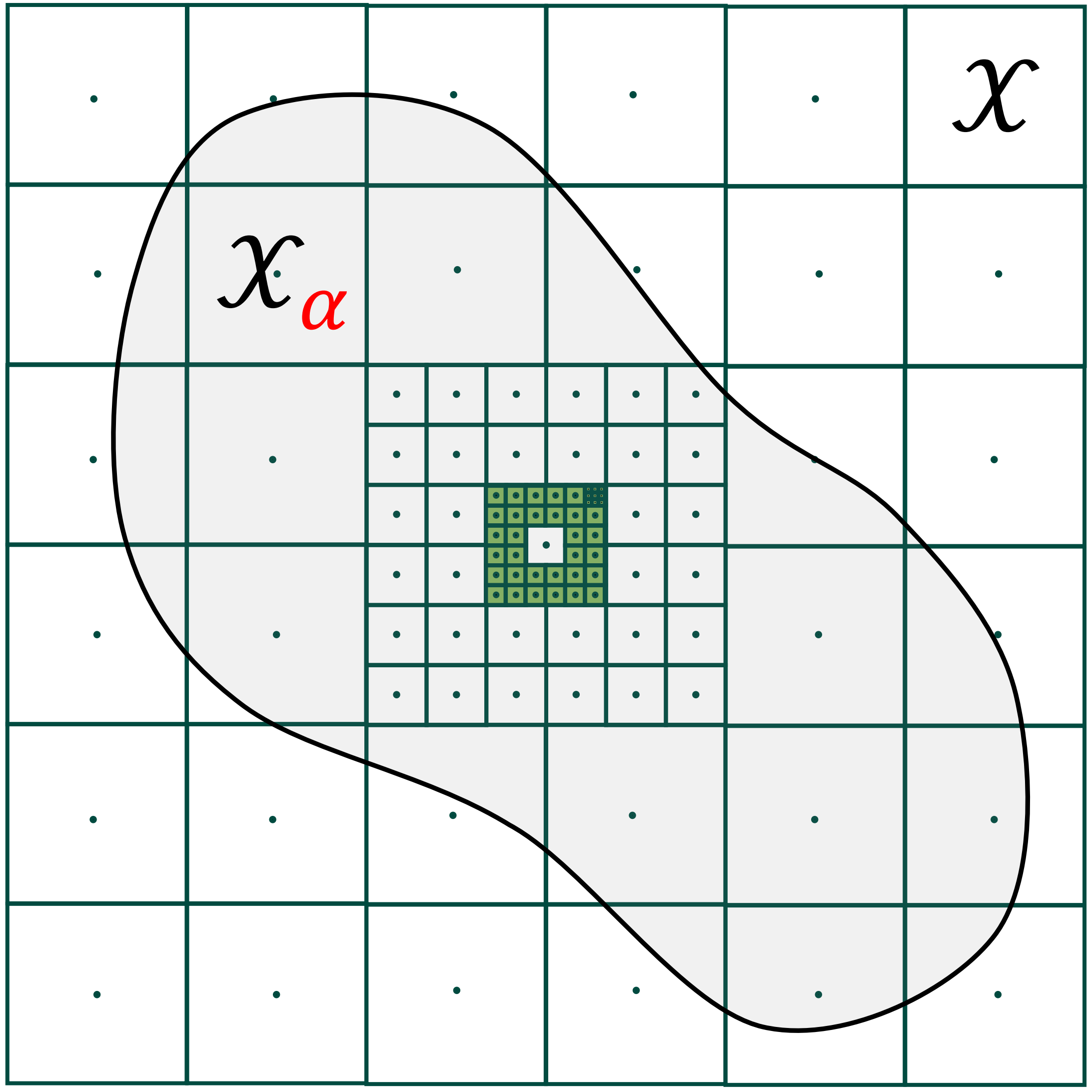
with $q = \frac{1-Ke^{(\alpha-\lambda)\tau}}{1+e^{(L+\alpha)\tau}} < 1.$



Nonparametric Stability Verification via GPUs

- **Basic Algorithm:**
  - Consider $V(x) = ||x - x^*||_\infty$
  - Build a grid of hypercubes surrounding x^*
  - Test grid center points:
    - Simulate trajectories of length τ
    - Find α s.t. the verified radius is $r \geq \ell/2$
  - Hypercube **not verified**, **split in 3^d** parts
  - Repeat testing of new points
  - **Exponentially expand** to outer layer
  - Repeat testing in new layer

- **Two Alg. Variations:**
  - Alg. 1: Find largest $\alpha_{\max}$ for region $\mathcal{X}$
  - Alg. 2: Find region $\mathcal{X}_\alpha$ for given α



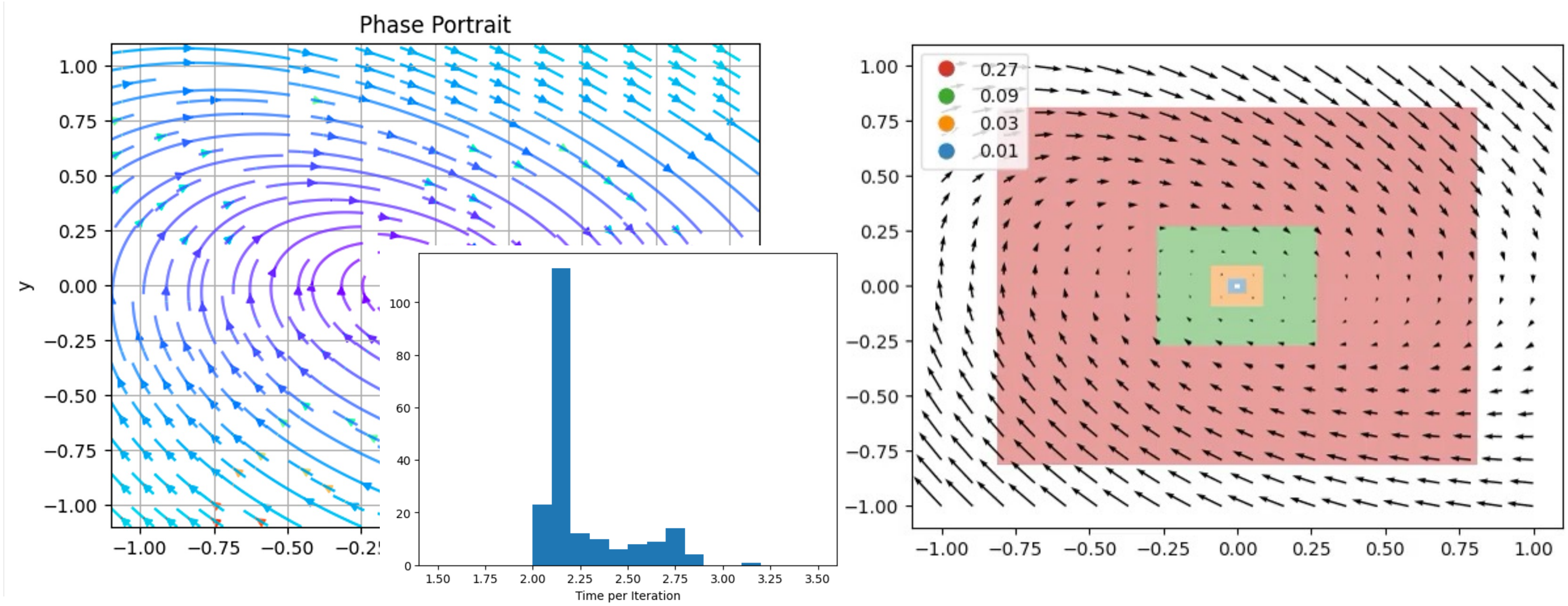
Numerical Illustration – Find Best α

Consider the 2-d non-linear system: $\dot{x} = \begin{bmatrix} 0 & 2 \\ -1 & -1 \end{bmatrix} x + B \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix}$

with $B_{ij} \sim \mathcal{N}(0, \sigma^2)$

$\sigma = 0.3$

$\alpha_{\max} = 0.470$

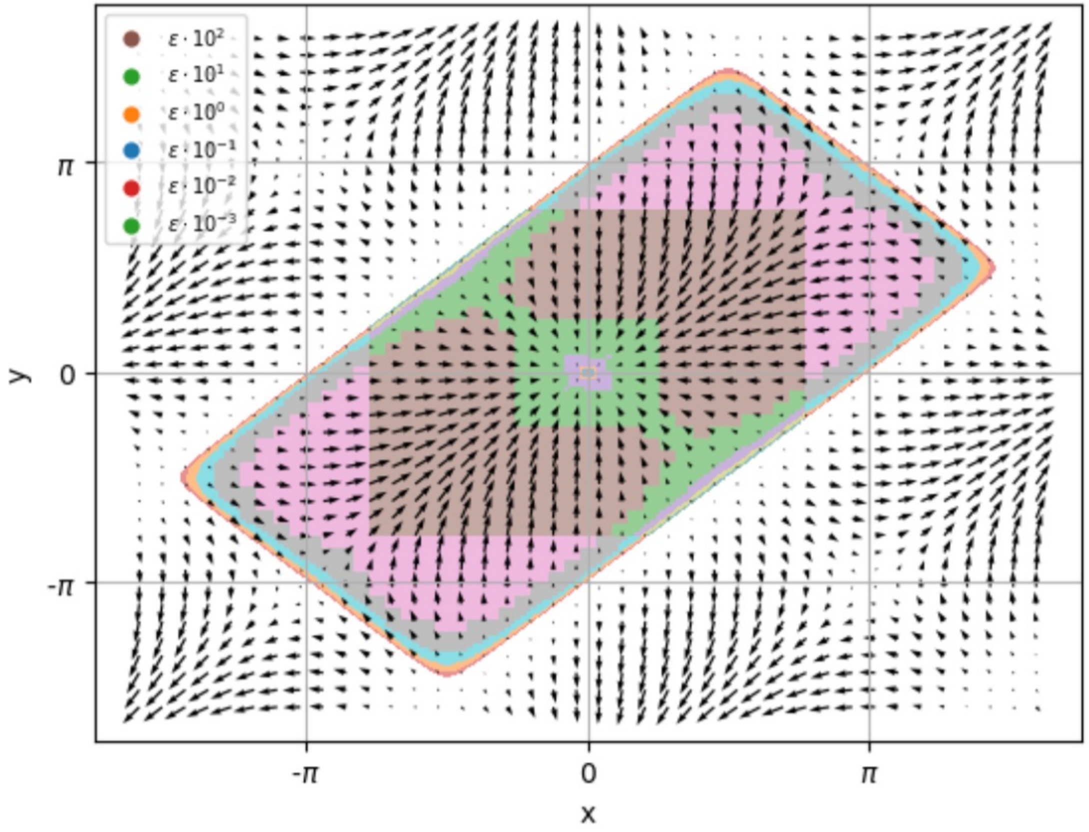


Numerical Illustration – Find region $\mathcal{X}_\alpha$

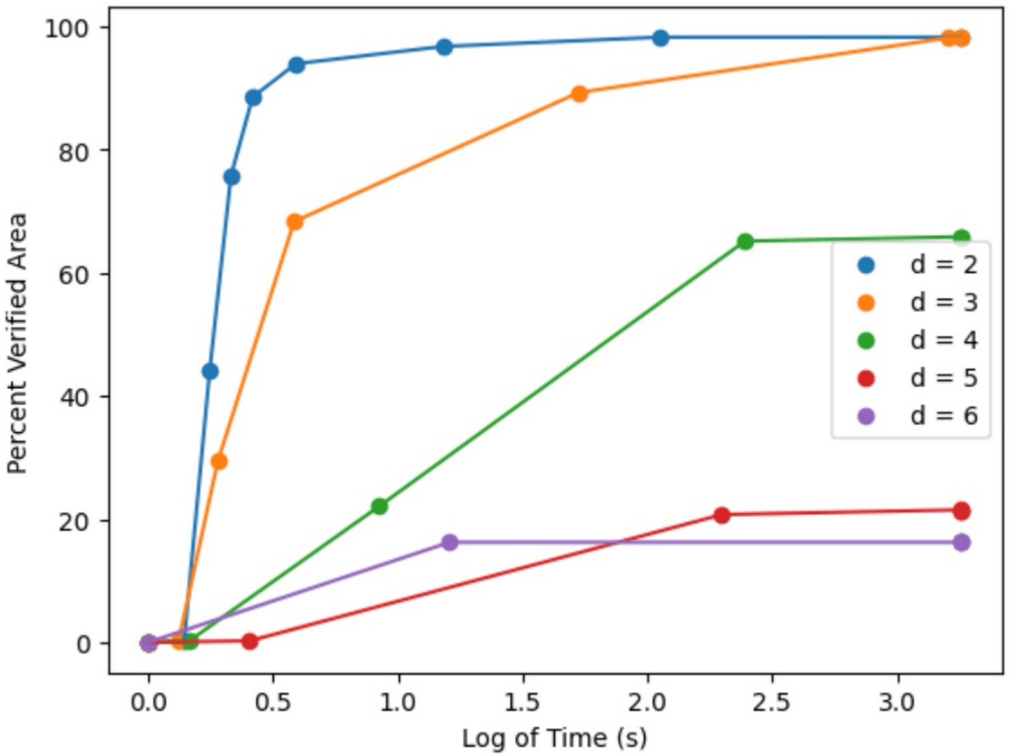
Consider the system of n Kuramoto oscillators:

$$\dot{\theta}_i = \frac{k}{n}\sum_{j=1}^{n} \sin(\theta_j - \theta_i)$$

Parameters: $n = 3$ and $\alpha = 1$



System dimension: $d = n - 1$



Outline

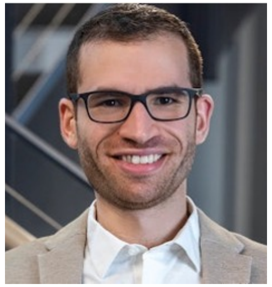
- **Relaxing Invariance: Merits and trade offs**
  - *Recurrent Sets:* Letting thing go and come back
- **Nonparametric Analysis via Recurrent Sets**
  - *Stability analysis:* Recurrent Lyapunov Functions (RLFs)
  - *Safety verification:* Recurrent Barrier Functions (RBFs)
- **Self-Improving via Nonparametric Control Policies**
  - Policy Improvement using Expert Demonstrations

Outline

- **Relaxing Invariance: Merits and trade offs**
  - *Recurrent Sets:* Letting thing go and come back
- **Nonparametric Analysis via Recurrent Sets**
  - *Stability analysis:* Recurrent Lyapunov Functions (RLFs)
  - *Safety verification:* Recurrent Barrier Functions (RBFs)
- **Self-Improving via Nonparametric Control Policies**
  - Policy Improvement using Expert Demonstrations

Yue Shen

Johns Hopkins University



Hussein Sibai

WashU

Nonparametric Safety Verification using Recurrence

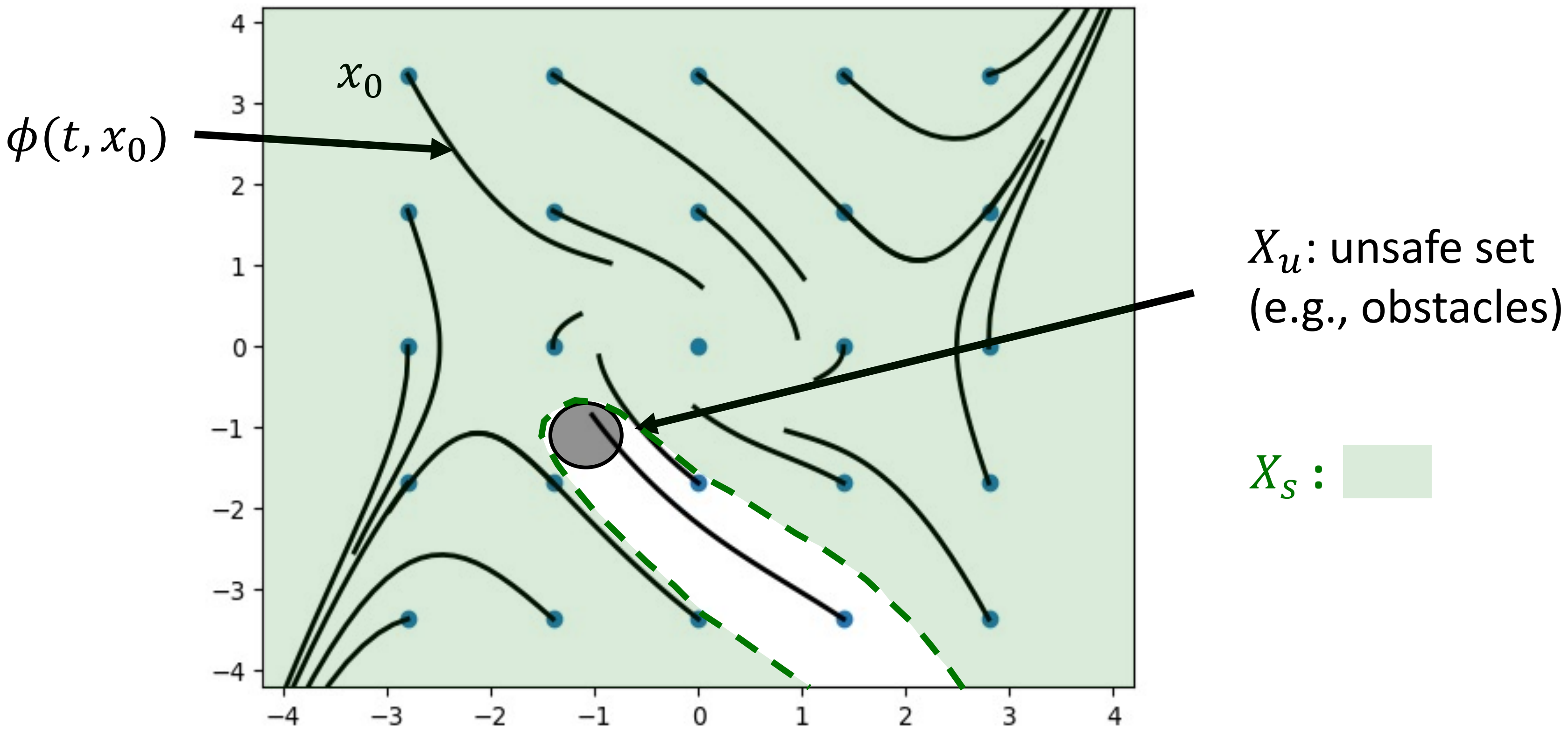
Y. Shen, H. Sibai, E. Mallada, "Generalized Barrier Functions: Integral Conditions and Recurrent Relaxations", in 60th Allerton Conference on Communication, Control, and Computing 2024

Safety in Dynamical Systems

Consider the continuous-time dynamical system: $\dot{x} = f(x)$

- $\phi(t, x_0)$: solution at time t starting from x_0
- X_u: set of unsafe states

Goal: Find the safe set $\mathcal{X}_s := \{x_0 \in \mathbb{R}^d | \phi(t, x_0) \notin \mathcal{X}_u, \forall t \geq 0\}$



Safety in Dynamical Systems via Invariant Sets

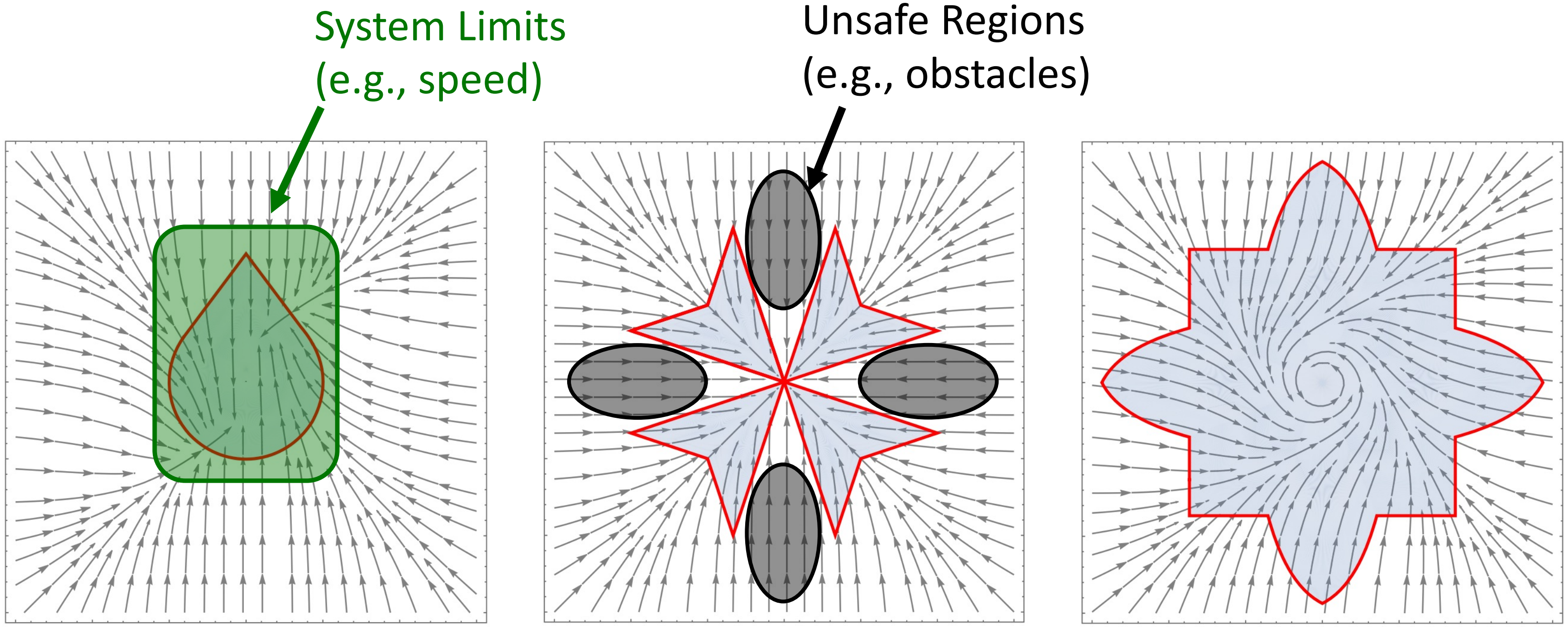
Consider the continuous-time dynamical system: $\dot{x} = f(x)$

- $\phi(t, x_0)$: solution at time t starting from x_0
- X_u: set of unsafe states

Goal: Find the safe set $\mathcal{X}_s := \{x_0 \in \mathbb{R}^d | \phi(t, x_0) \notin \mathcal{X}_u, \forall t \geq 0\}$

General Approach: Use invariant sets!

A set $\mathcal{S} \subseteq \mathbb{R}^d$ is **invariant** if and only if: $x_0 \in \mathcal{S} \rightarrow \phi(t, x_0) \in \mathcal{S}, \ \forall t \geq 0$



Certifying Safety using Barrier Functions

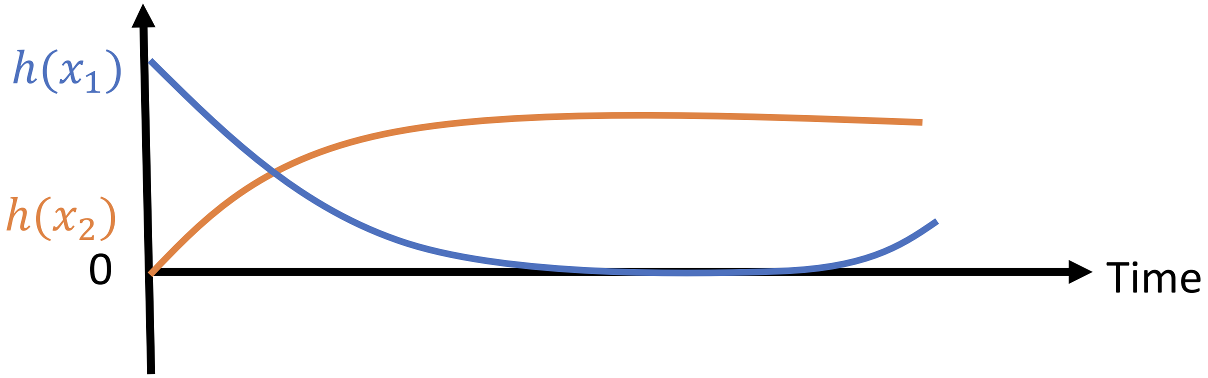
Mitio Nagumo

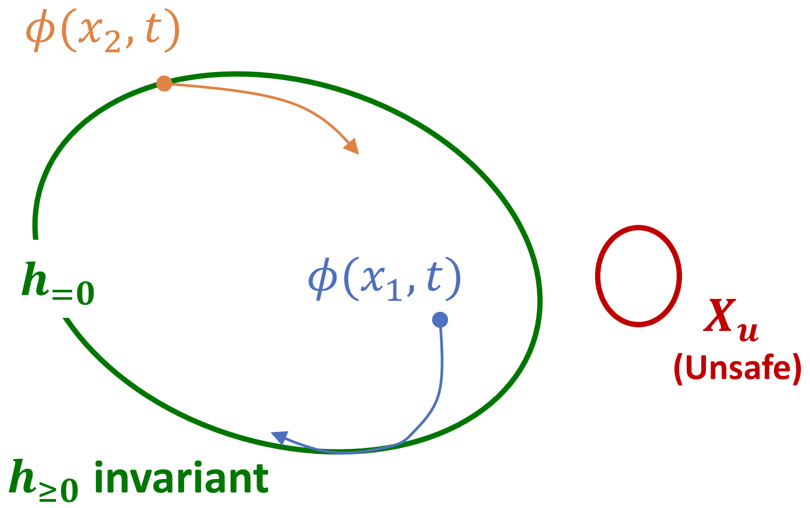
Theorem - Nagumo's Barrier Functions [Nagumo '42] **:**

Let $h: \mathbb{R}^{\mathrm{d}} \to \mathbb{R}$ be differentiable, with 0 being a *regular value.*
Then h is a Nagumo's Barrier Function (NBF) satisfying:

$$L_f h(x) := \lim_{t \to 0} \frac{h(\phi(t,x)) - h(x)}{t} \geq 0, \quad \forall x \in h_{=0},$$

if and only if $h_{\geq 0} := \{x \in \mathbb{R}^d \,|\, h(x) \geq 0\}$ *is invariant.*





Then $h_{\geq 0}$ is a safe set whenever $h_{\geq 0} \cap X_u = \emptyset$

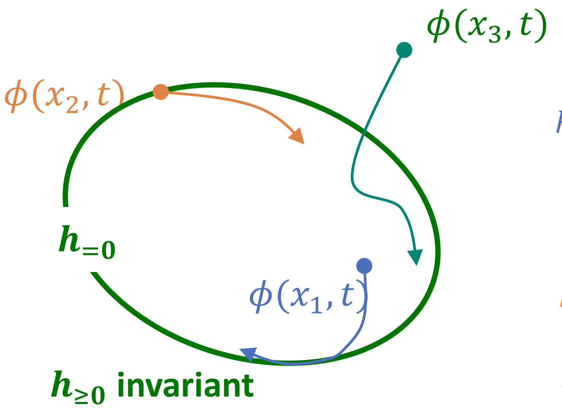
M. Nagumo, "Über die lage der integralkurven gewöhnlicher differentialgleichungen," Proceedings of the Physico-Mathematical Society of Japan 1942

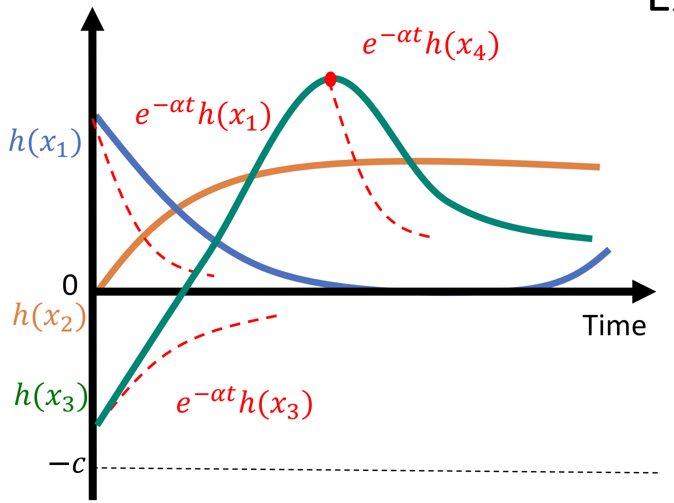
Shaping Behavior using Barrier Functions (BFs)

Barrier functions provide a flexible framework to shape the behavior of trajectories

Zeroing Barrier Function

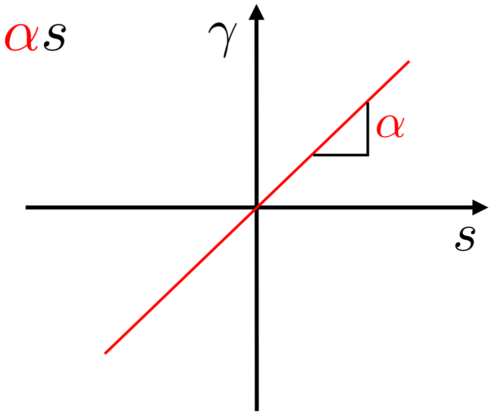
$$L_f h(x) \geq -\gamma\left(h(x)\right), \quad \forall x \in h_{\geq -c}$$





Extended Class $\mathcal{K}_e$:

- $\gamma \in \mathcal{K}_e$ iff $\gamma'(s) \geq 0$ and $\gamma(0) = 0$
- Example: $\gamma_\alpha(s) = \alpha s$



Other: Exponential BFs (EBFs), Minimal BFs (MBFs), Control BFs (CBFs), High Order CBFs (HOCBFs), …

S. Prajna, A. Jadbabaie. *Safety Verification of Hybrid Systems Using Barrier Certificates.* HSCC 2004

P. Wieland, F. Allgöwer. *Constructive safety using control barrier functions*. IFAC Proceedings Volumes 2007

A. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, P. Tabuada. *Control barrier functions: Theory and applications*. IEEE ECC 2019

R. Konda, A. Ames, S. Coogan. *Characterizing safety: Minimal control barrier functions from scalar comparison systems.* IEEE L-CSS 2020

W. Xiao, C. Belta. *High-order control barrier functions*. IEEE TAC 2021

Shaping Behavior using Barrier Functions (BFs)

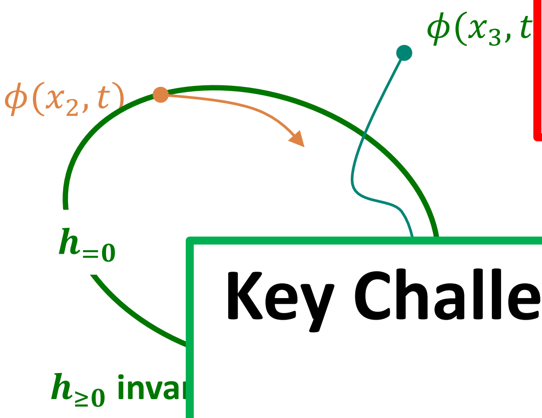
Barrier functions provide a flexible framework to shape the behavior of trajectories

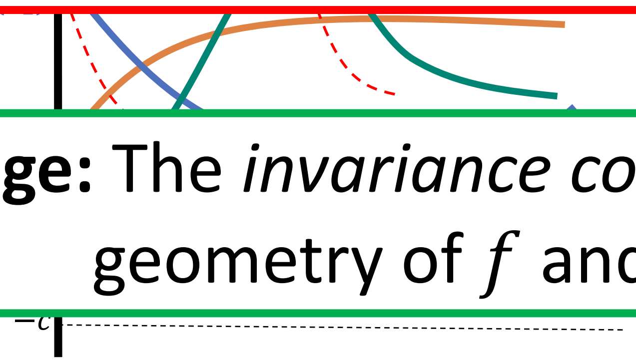
Zeroing Barrier Function

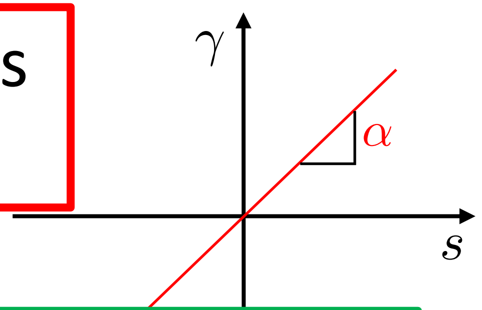
$$L_f h(x) \geq -\gamma\left(h(x)\right), \quad \forall x \in h_{\geq -c}$$

Extended Class $\mathcal{K}_e$:

- $\gamma \in \mathcal{K}_e$ iff $\gamma'(s) \geq 0$ and $\gamma(0) = 0$







Problem: Finding Barrier Functions is usually difficult

Key Challenge: The *invariance condition* on $h_{\geq 0}$ couples the geometry of f and the set $h_{\geq 0}$

Other: Exponential BFs (ZBFs), Minimal BFs (MBFs), Control BFs (CBFs), High Order CBFs (HOCBFs), …

S. Prajna, A. Jadbabaie. *Safety Verification of Hybrid Systems Using Barrier Certificates.* HSCC 2004

P. Wieland, F. Allgöwer. *Constructive safety using control barrier functions*. IFAC Proceedings Volumes 2007

A. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, P. Tabuada. *Control barrier functions: Theory and applications*. IEEE ECC 2019

R. Konda, A. Ames, S. Coogan. *Characterizing safety: Minimal control barrier functions from scalar comparison systems.* IEEE L-CSS 2020

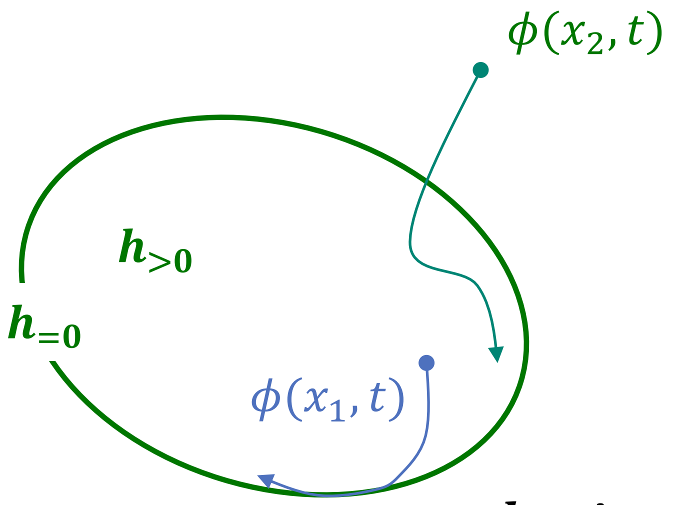
W. Xiao, C. Belta. *High-order control barrier functions*. IEEE TAC 2021

Recurrent Barrier Functions

Barrier Function:
Let h *be differentiable,* $\gamma \in \mathcal{K}_e$, *and*

$$L_f h(x) \geq -\gamma\left(h(x)\right)$$

then, $h_{\geq 0}$ is invariant



$h_{\geq 0}$ **invariant**

Recurrent Barrier Functions

Barrier Function:
Let h *be differentiable,* $\gamma \in \mathcal{K}_e$, *and*

$$L_f h(x) + \gamma\left(h(x)\right) \geq 0$$

then, $h_{\geq 0}$ is invariant

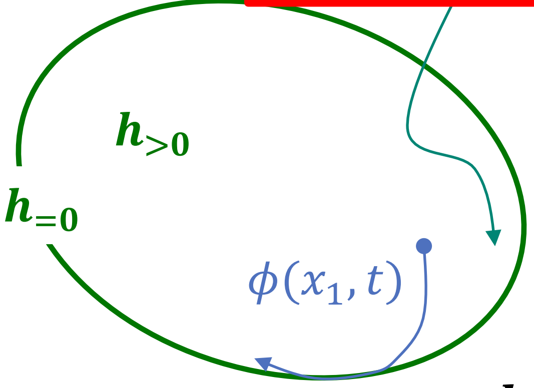
As $\tau \to 0$ ⇆ By definition

Recurrent Barrier Function:
Let h *be continuous,* $\gamma \in \mathcal{K}_e$, *and*

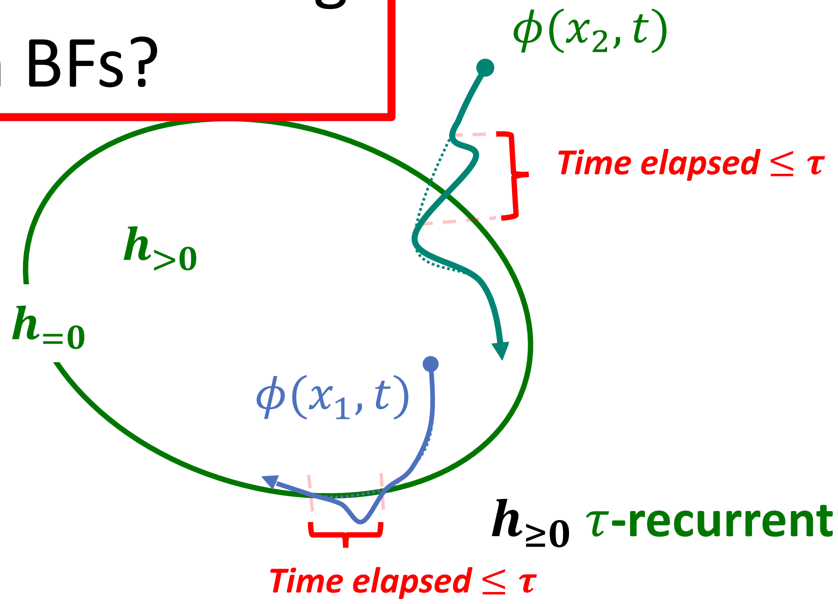
$$\max_{t\in(0,\tau]} h(\phi(x,t)) + \int_0^t \gamma(h(\phi(x,s))ds \geq h(x)$$

then, $h_{\geq 0}$ is ***τ*-recurrent**

Question: Do we gain anything from relaxing the invariance condition in BFs?



$h_{\geq 0}$ **invariant**



$h_{\geq 0}$ τ**-recurrent**

Assessing Safety via Recurrent BFs

Claim 1: Signed norms are Recurrent BFs!

Let h be a Zeroing BF, with $\gamma_{\underline{\alpha},\bar{\alpha}} \in \mathcal{K}_e$ given by

$$\gamma_{\underline{\alpha},\bar{\alpha}}(s) = \begin{cases} \bar{\alpha}s, & s \geq 0 \\ \underline{\alpha}s, & s < 0 \end{cases}$$

Then, for any set S with $h_{\geq 0} \subseteq \mathcal{S} \subseteq h_{\geq -c}$, the function:

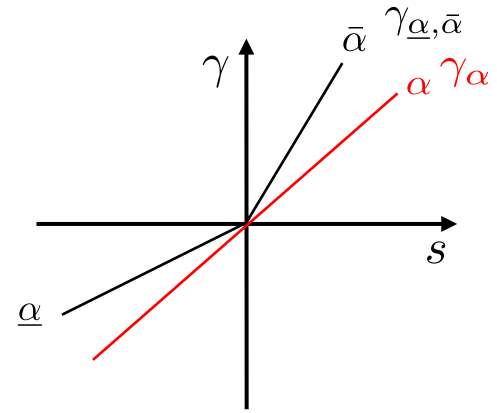
$$\hat{h}(x) := -\mathrm{sd}(x, \mathcal{S})$$

is a **Recurrent BF** with $\gamma_\alpha = \alpha s$, with $\underline{\alpha} < \alpha < \bar{\alpha}$

Claim 2: Safety verification with RBFs

If $\hat{h} = -\mathrm{sd}(x, S)$ is an RBF, then the set S is a safe set whenever:

$$S \cap \mathcal{R}_{[-\tau,0]}(\mathcal{X}_u) = \emptyset$$

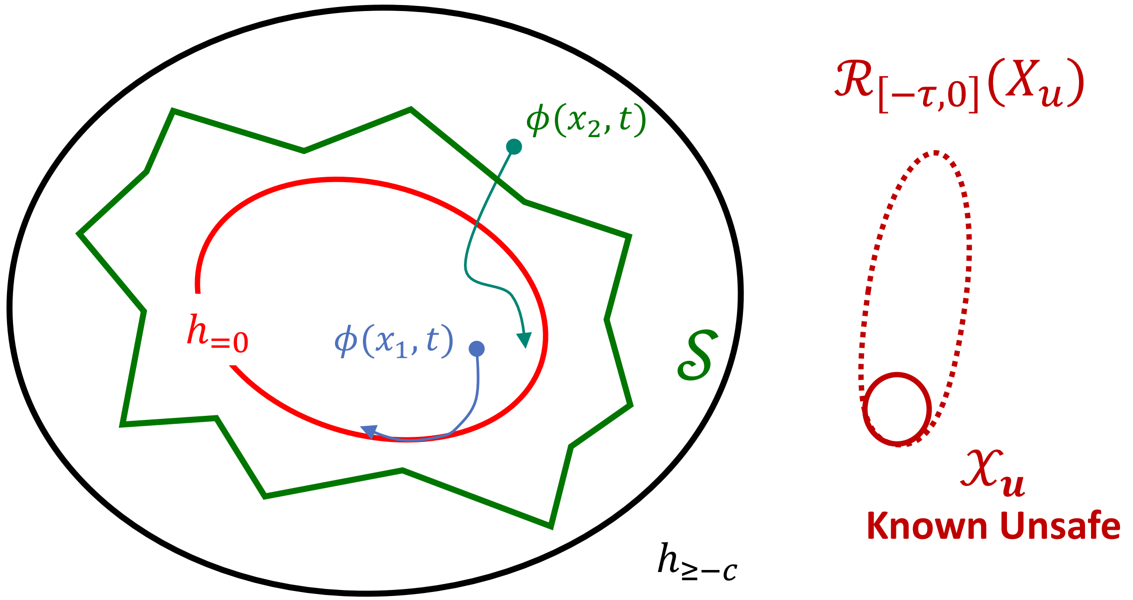


BF:

$$L_f h(x) + \gamma_{\underline{\alpha},\bar{\alpha}}(h(x)) \geq 0$$

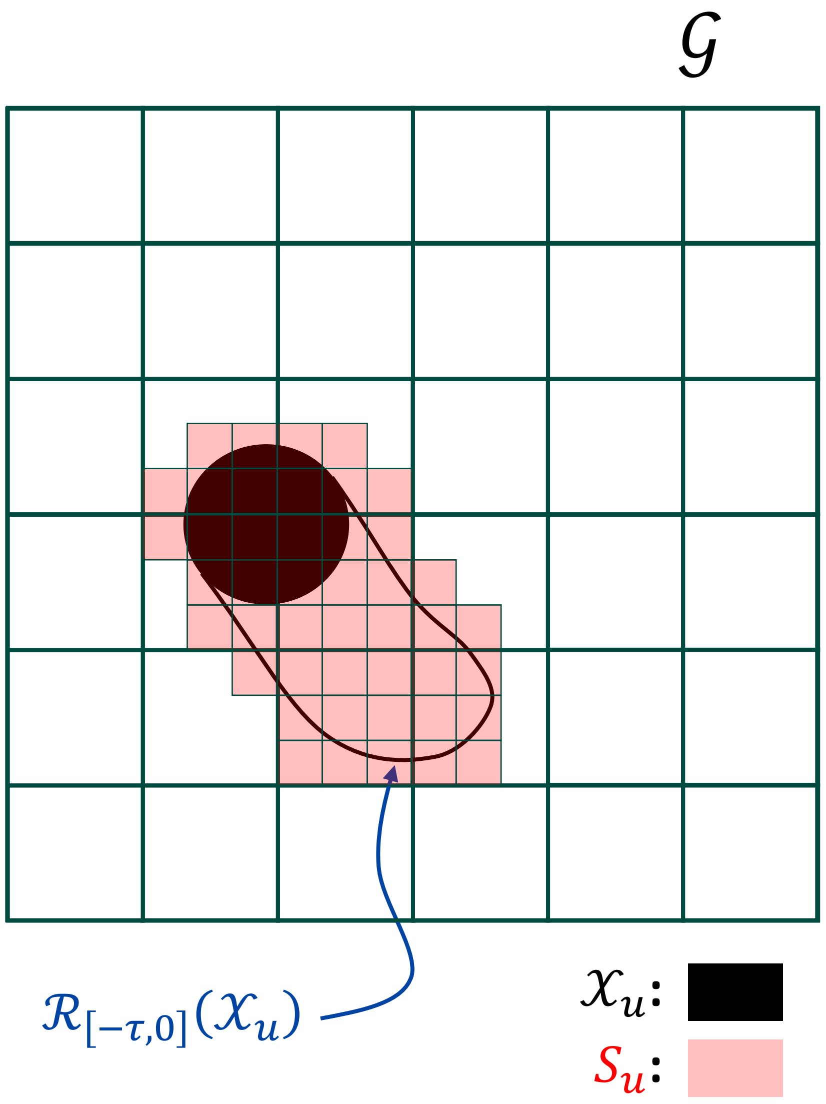
RBF:

$$\max_{t \in (0,\tau]} \hat{h}(\phi(x,t)) + \int_0^t \gamma_\alpha(\hat{h}(\phi(x,s))ds \geq \hat{h}(x)$$



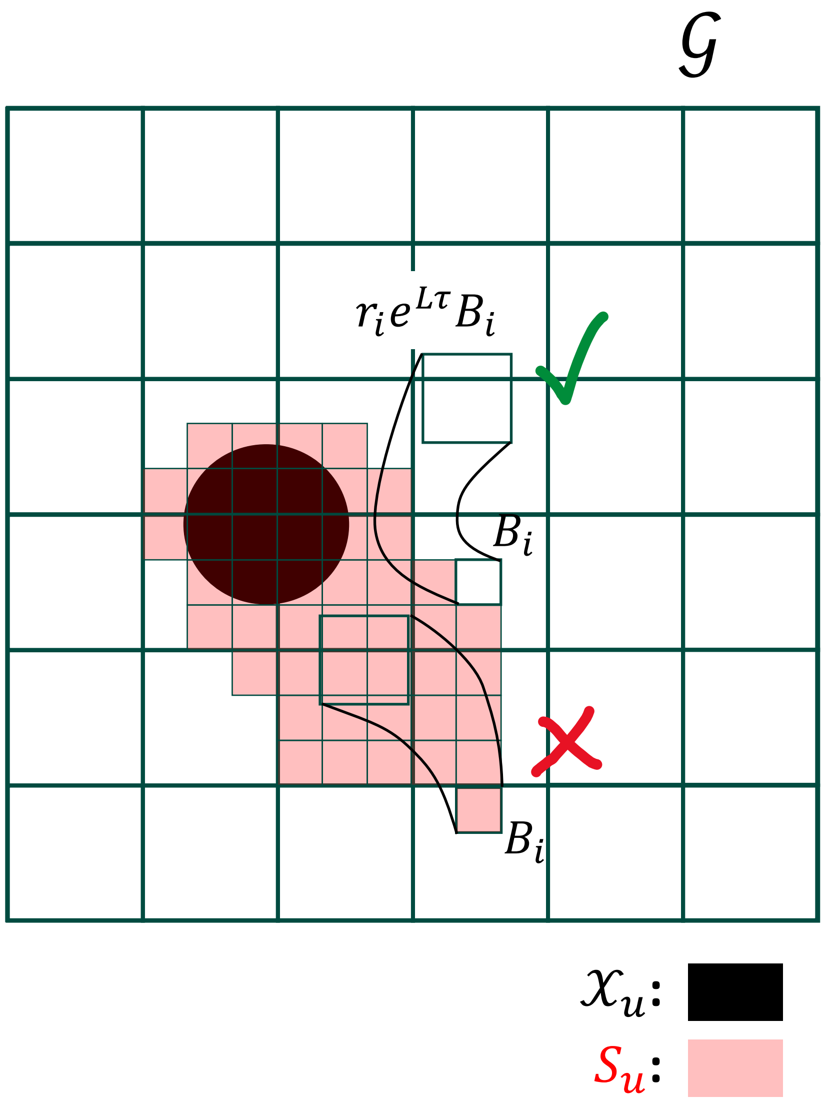
Basic Algorithm

- Given unsafe region $\mathcal{X}_u$, precision $r_{\min}$
- Build initial grid of hypercubes $\mathcal{G} = \{B_i := B_{r_i}(x_i)\}$
- **Stage 1:** τ −Backward reachability
  - Find $S_u = \cup_i B_i$ such that: $\mathcal{R}_{[-\tau,0]}(\mathcal{X}_u) \subset S_u$
- **Stage 2:** Check RBF on $h(x) = -\mathrm{sd}(x, (S_u)^c)$



Basic Algorithm

- Given unsafe region $\mathcal{X}_u$, precision $r_{\min}$
- Build initial grid of hypercubes $\mathcal{G} = \{B_i := B_{r_i}(x_i)\}$
- **Stage 1:** τ −Backward reachability
  - Find $S_u = \cup_i B_i$ such that: $\mathcal{R}_{[-\tau,0]}(\mathcal{X}_u) \subset S_u$
- **Stage 2:** Check RBF on $h(x) = -\mathrm{sd}(x, (S_u)^c)$
  - For $B_i \in \mathcal{G}$, while $\mathcal{G}$ not empty:
    - If: B_i satisfies RBF condition, **continue**
    - Else if: B_i can never satisfy RBF condition, **add B_i to S_u**
    - Else: **refine grid**

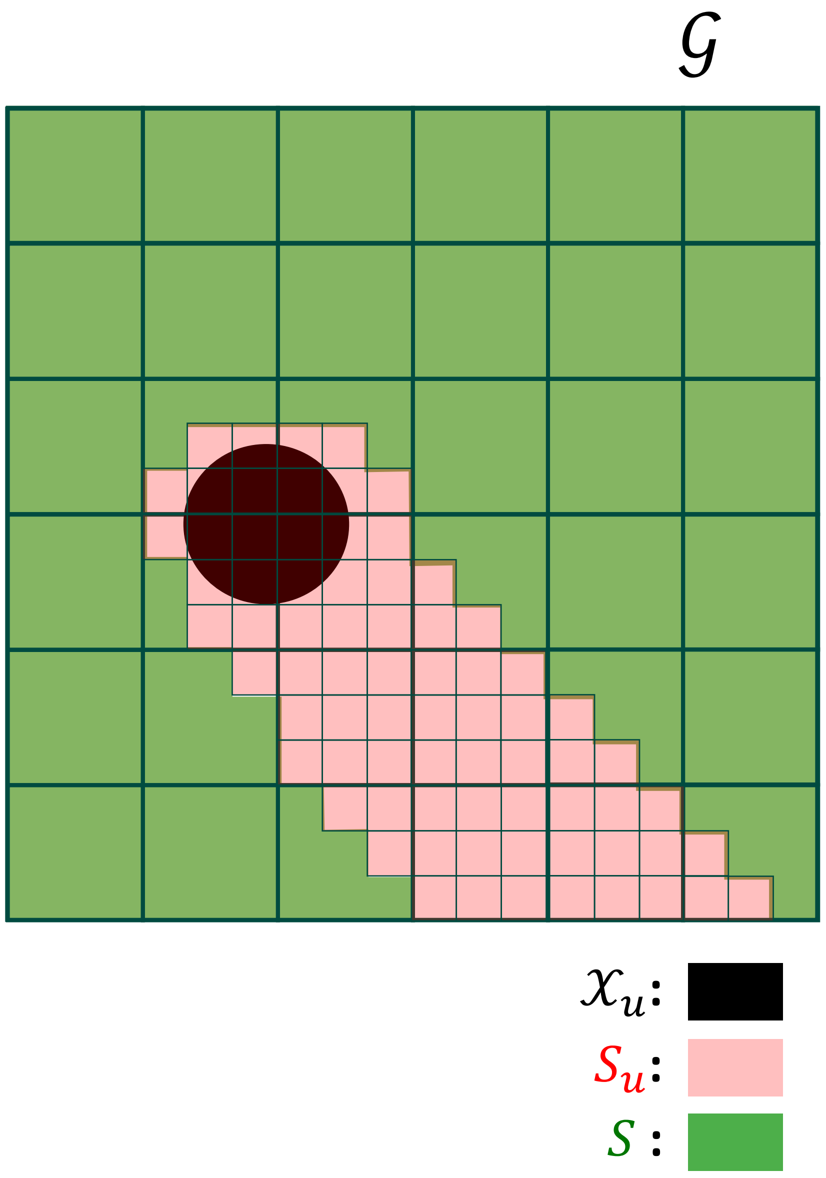


Basic Algorithm

- Given unsafe region $\mathcal{X}_u$, precision $r_{\min}$
- Build initial grid of hypercubes $\mathcal{G} = \{B_i := B_{r_i}(x_i)\}$
- **Stage 1:** τ −Backward reachability
  - Find $S_u = \cup_i B_i$ such that: $\mathcal{R}_{[-\tau,0]}(\mathcal{X}_u) \subset S_u$
- **Stage 2:** Check RBF on $h(x) = -\mathrm{sd}(x, (S_u)^c)$
  - For $B_i \in \mathcal{G}$, while $\mathcal{G}$ not empty:
    - If: B_i satisfies RBF condition, **continue**
    - Else if: B_i can never satisfy RBF condition, **add B_i to** S_u
    - Else: **refine grid**
  - **Finish when:**
    - all points satisfy RBF condition, or precision $r_{\min}$ is reached

Claim:

- The set $S = (S_u)^c$ satisfies: $S \cap \mathcal{R}_{[-\tau,0]}(\mathcal{X}_u)$
- The function $h(x) = -\mathrm{sd}(x, S)$ is an RBF



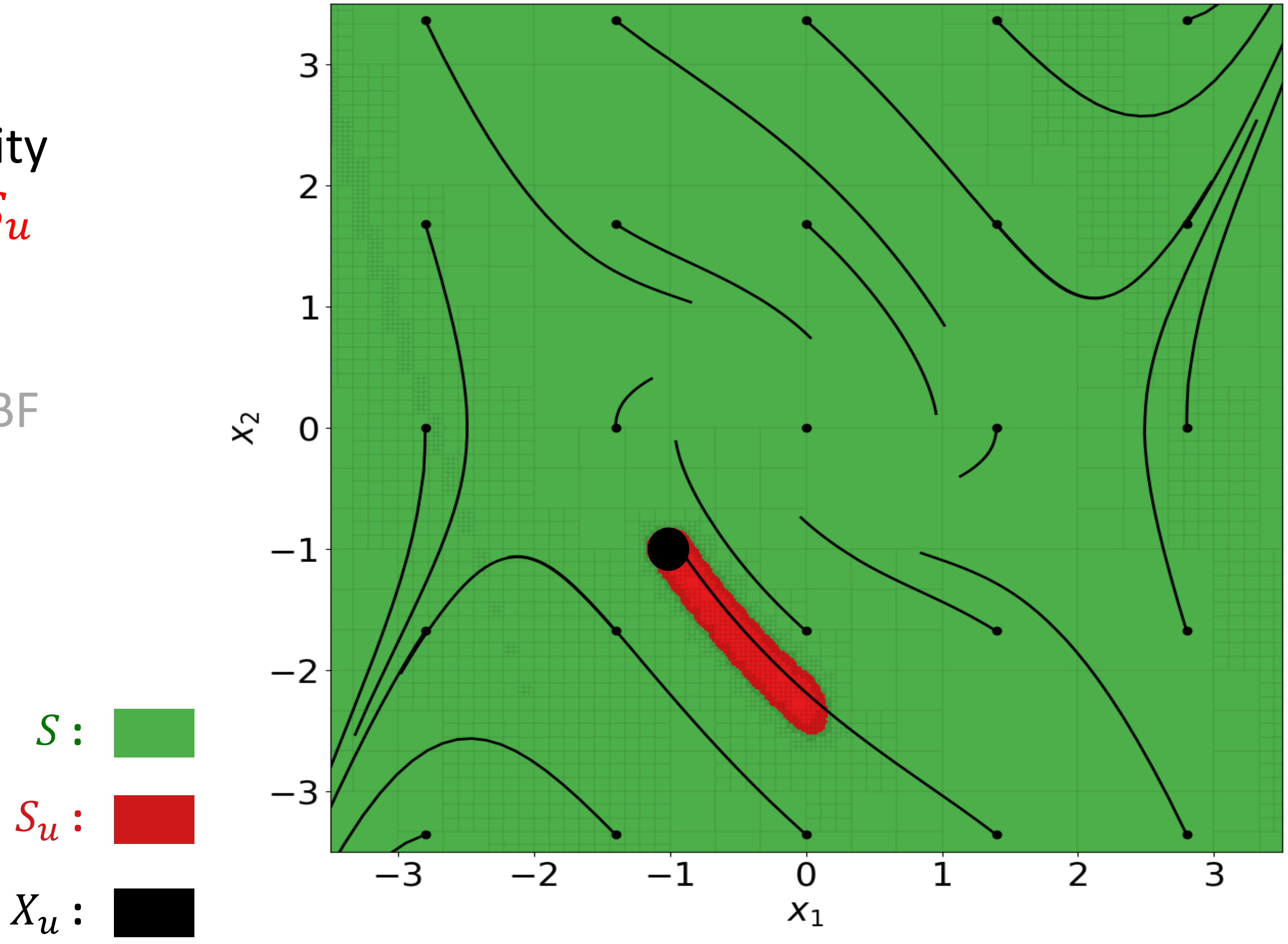
Nonparametric Safety Verification – Stage 1

Stage 1: τ −Backward reachability

- Find S_u with $\mathcal{R}_{[-\tau,0]}(\mathcal{X}_u) \subset S_u$

Stage 2: RBF condition

- Check $h(x) = -\mathrm{sd}(x, S)$ is RBF



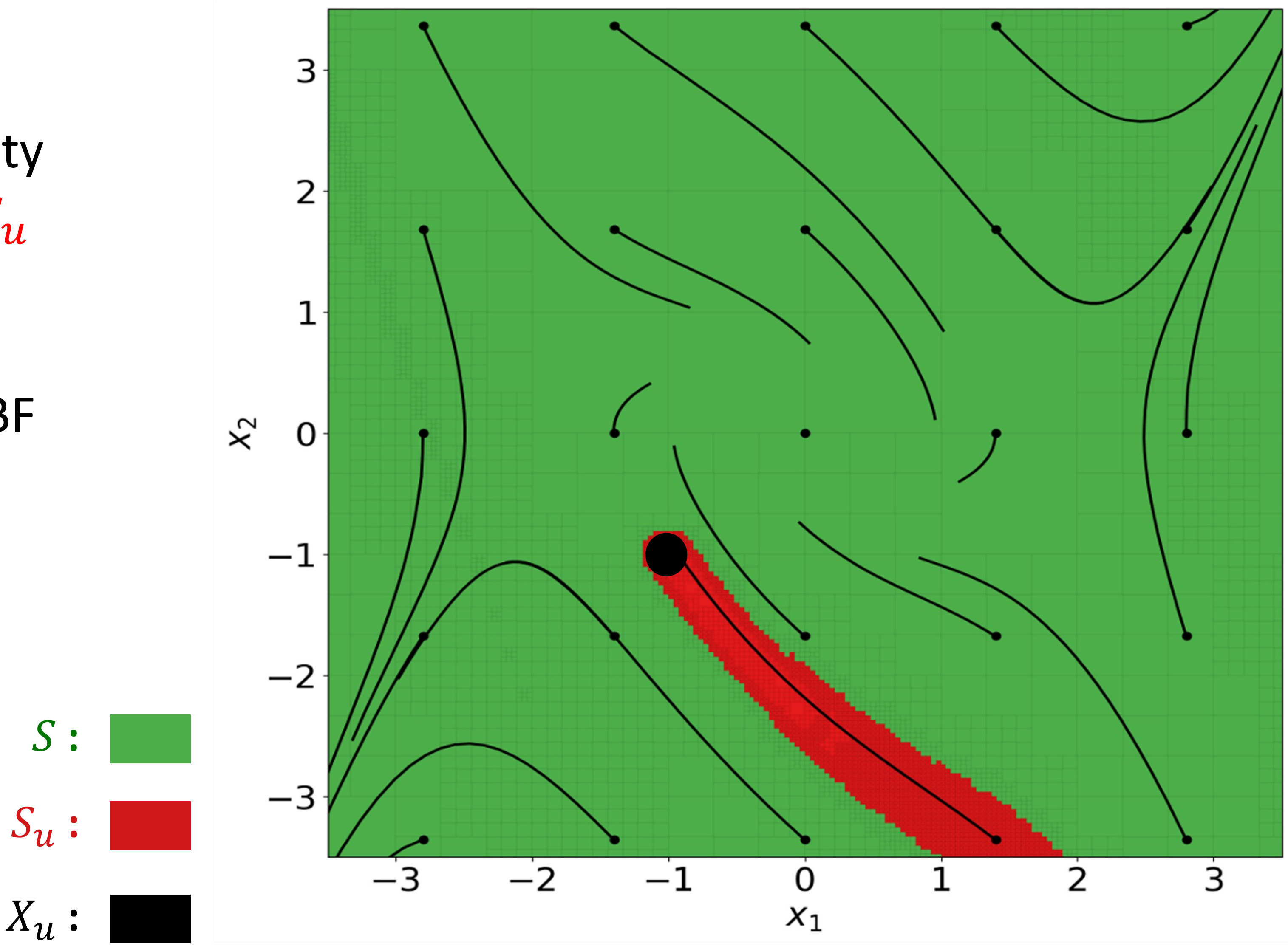
Nonparametric Safety Verification – Stage 1

Stage 1: τ –Backward reachability

- Find S_u with $\mathcal{R}_{[-\tau,0]}(\mathcal{X}_u) \subset S_u$

Stage 2: RBF condition

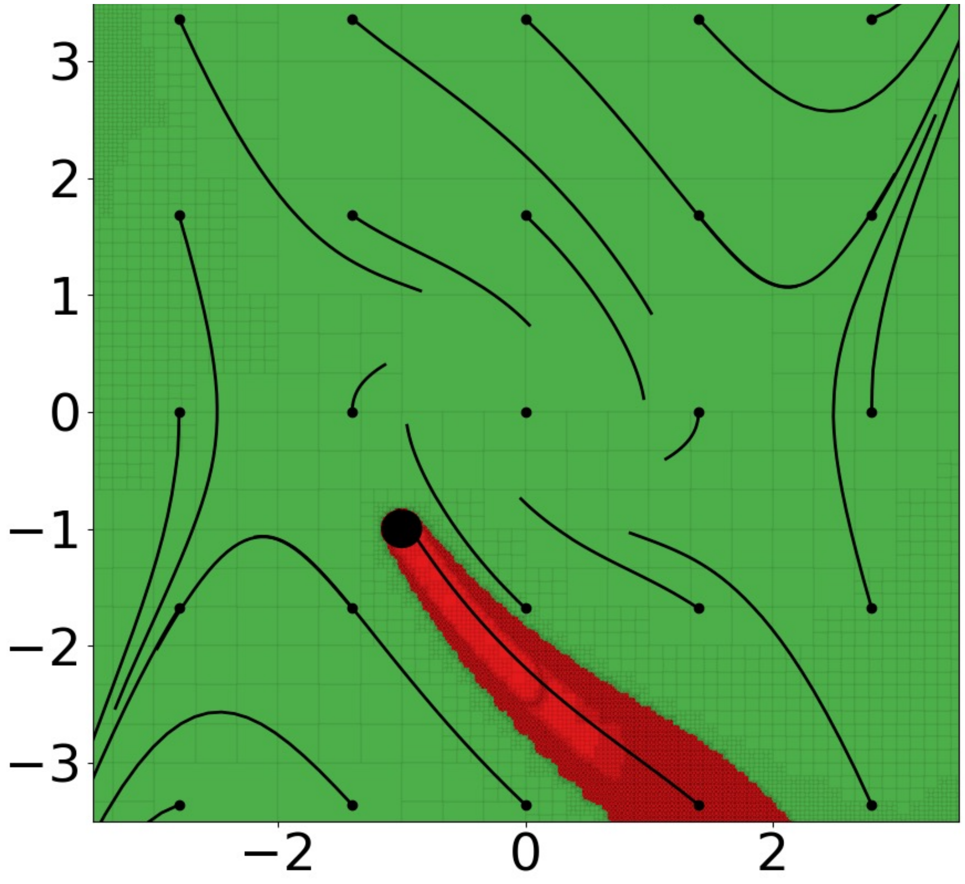
- Check $h(x) = -\text{sd}(x, S)$ is RBF



Numerical Validation: Reachability vs Recurrence

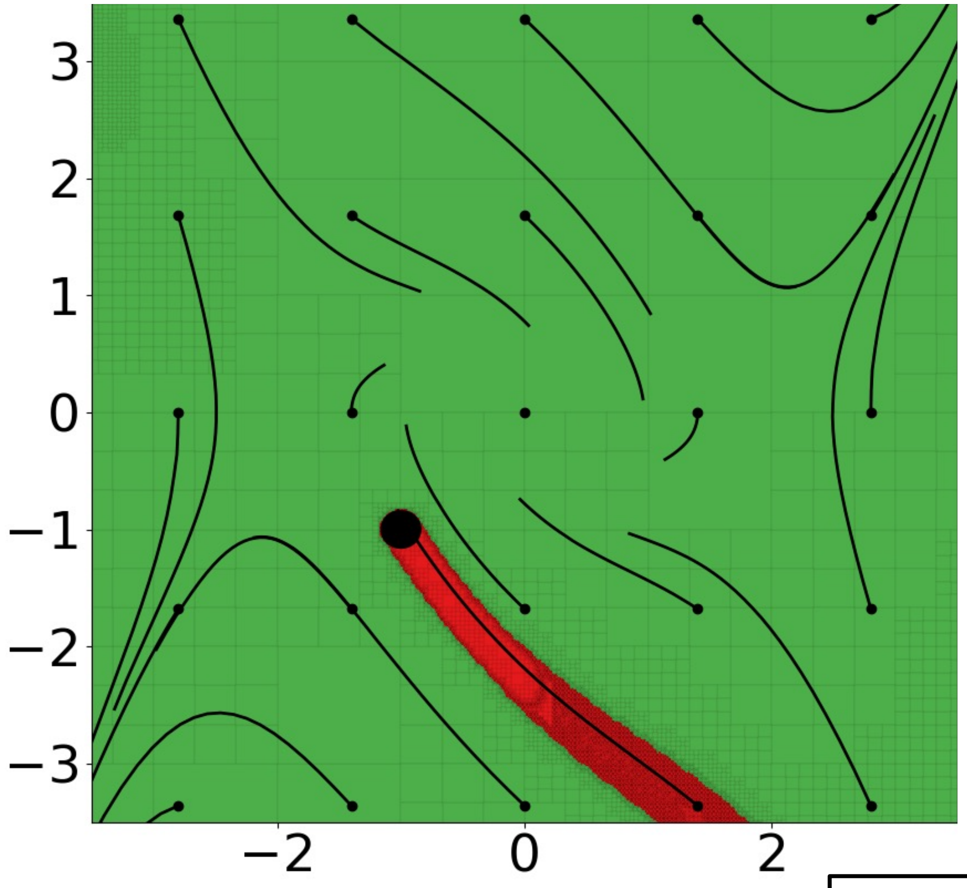
$r_{\min} = 0.01$

Reachability



running time: **354.12s**
97.4% safe set covered

Nagumo's RBF

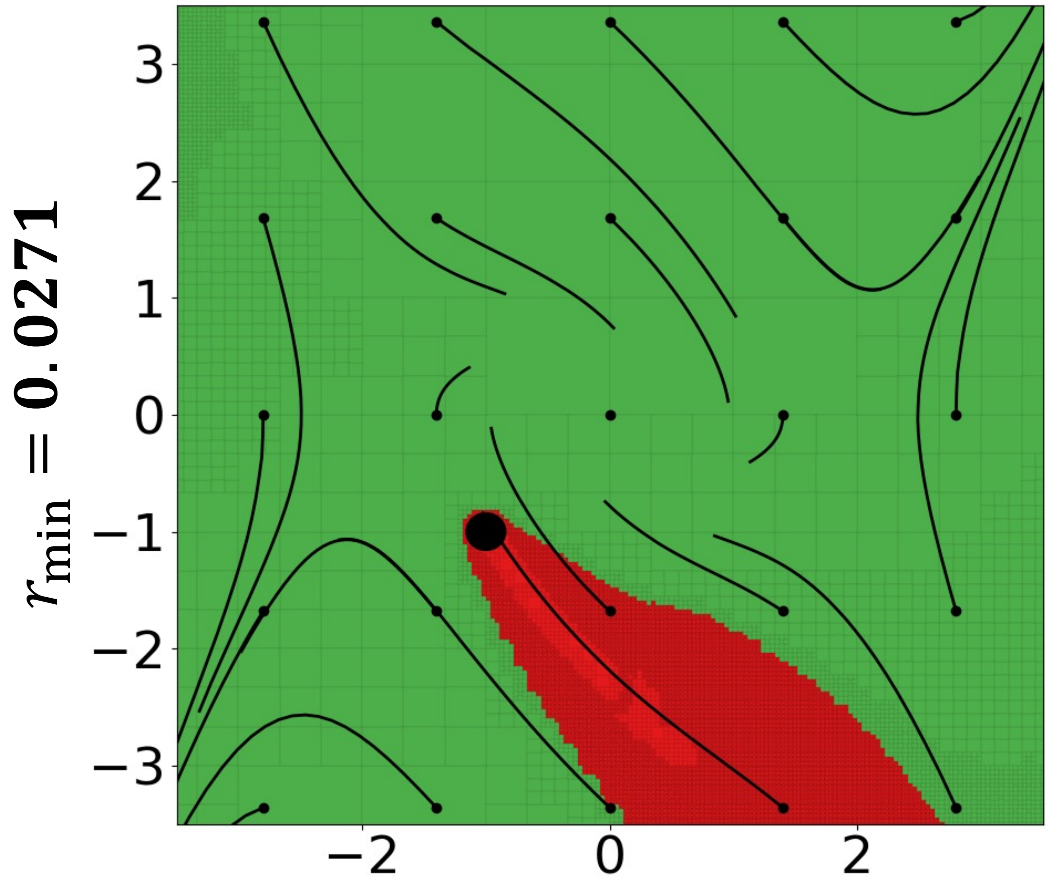


running time: **20.68s**
99.6% safe set covered

~17x faster
+2.2% more area

Numerical Validation: Reachability vs Recurrence

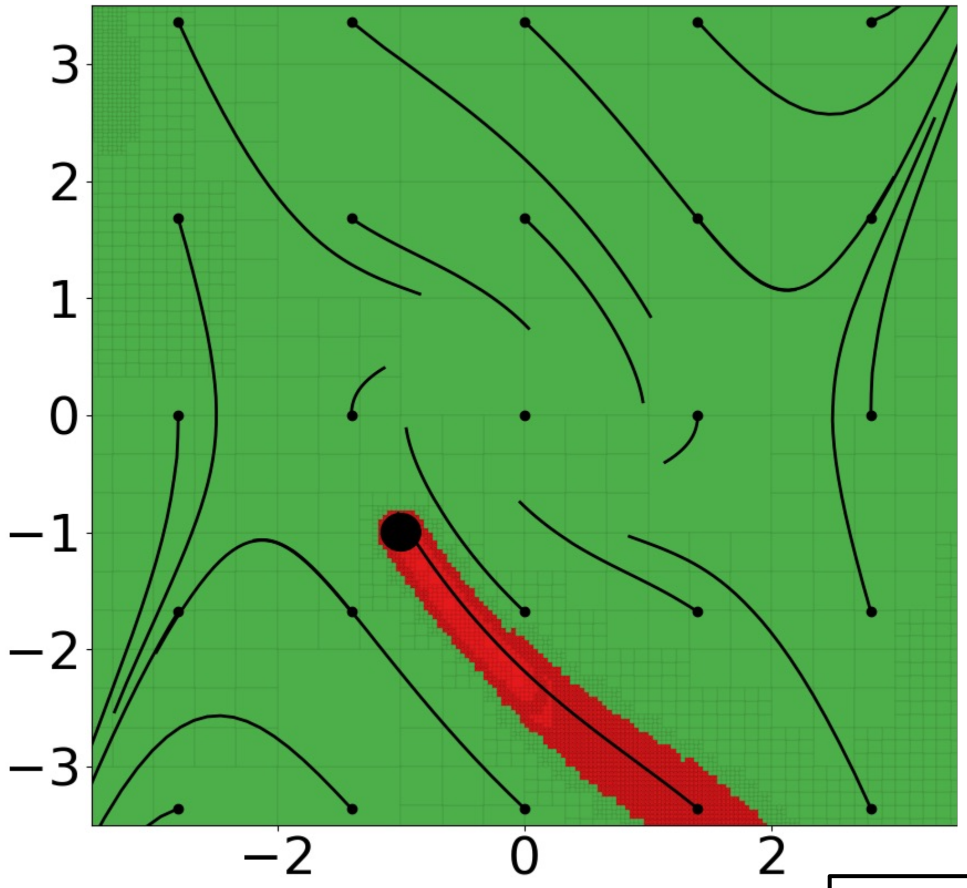
Reachability



running time: **23.15s**

92.2% safe set covered

Nagumo's RBF

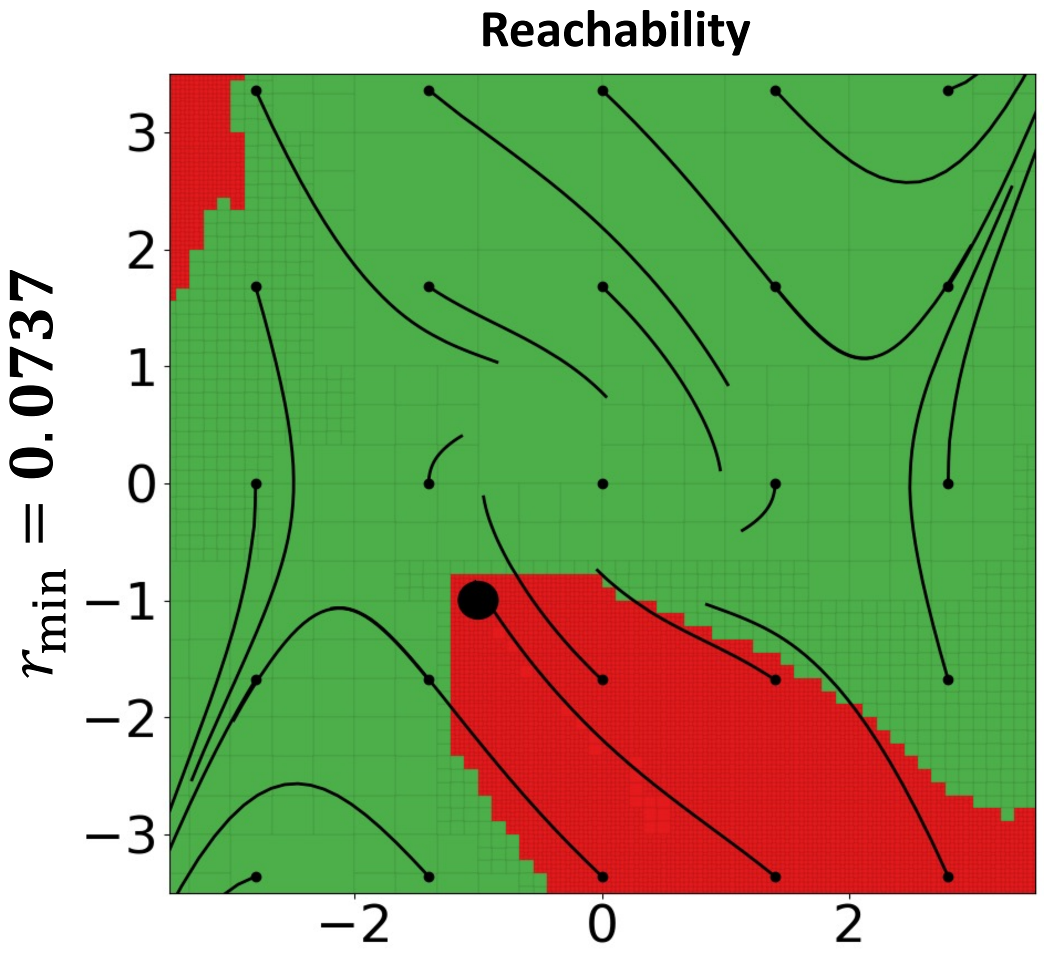


running time: **3.81s**

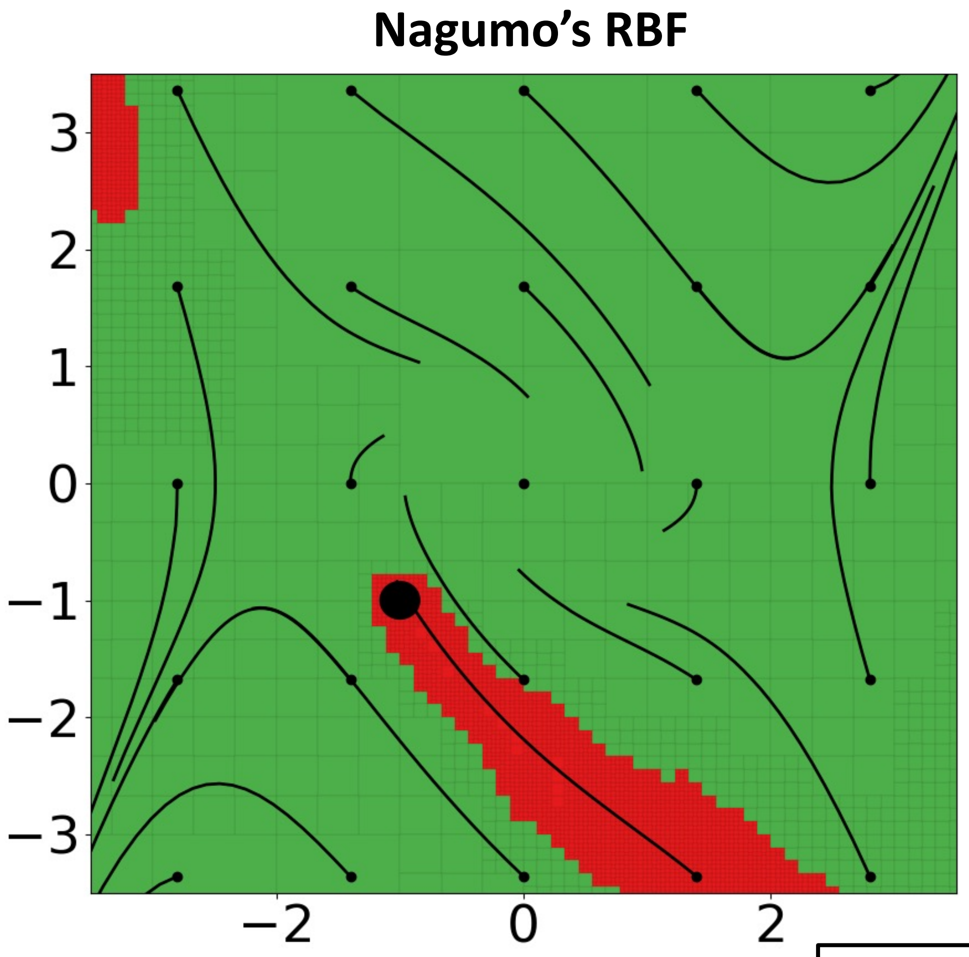
98.6% safe set covered

~6x faster
+6.4% more area

Numerical Validation: Reachability vs Recurrence



running time: **3.01s**
83.3% safe set covered

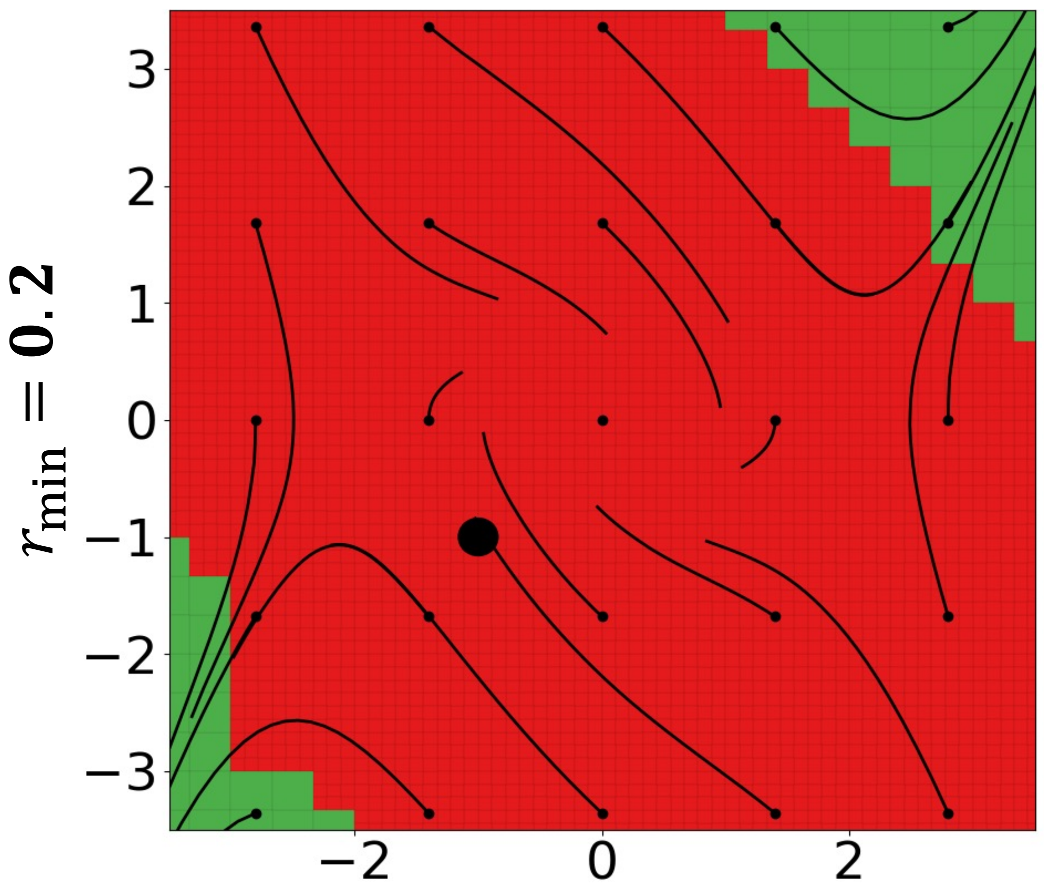


running time: **1.56s**
94.6% safe set covered

∼2x faster
+11.2% more area

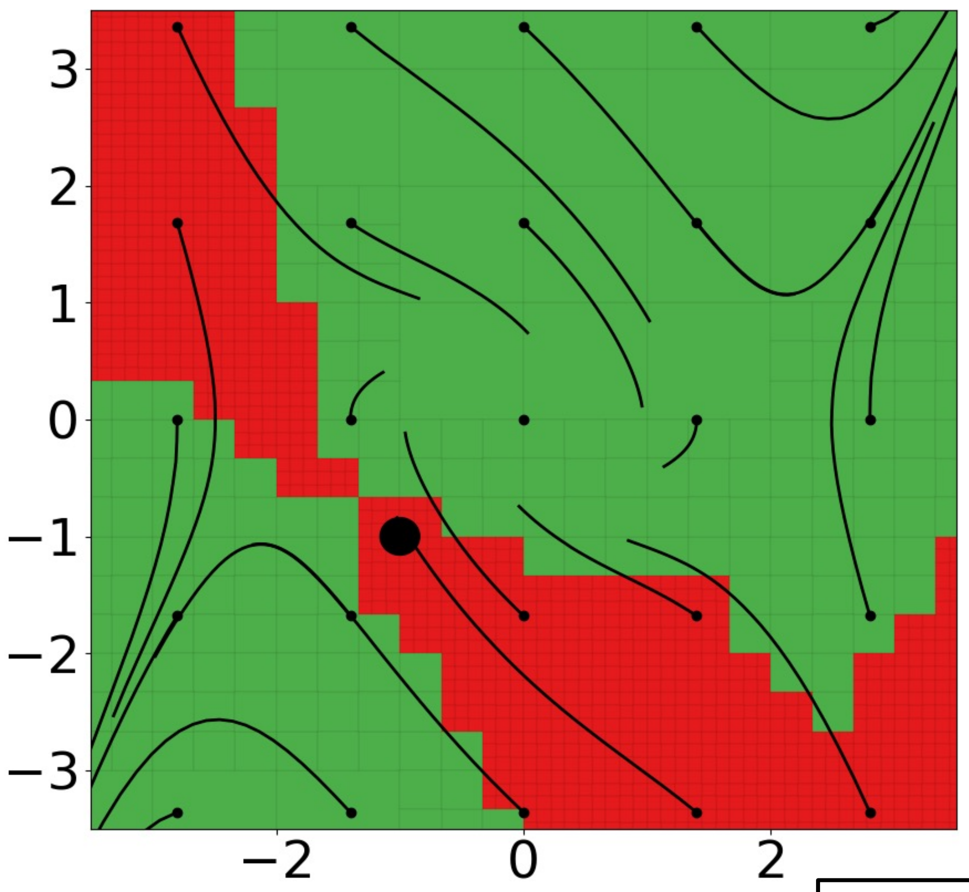
Numerical Validation: Reachability vs Recurrence

Reachability



running time: **1.86s**
9.3% safe set covered

Nagumo's RBF



running time: **0.31s**
71.2% safe set covered

6x faster
+61.9% more area

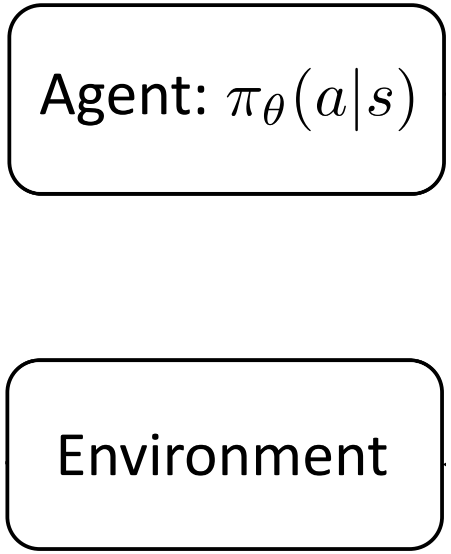
Outline

- **Relaxing Invariance: Merits and trade offs**
  - *Recurrent Sets:* Letting thing go and come back
- **Nonparametric Analysis via Recurrent Sets**
  - *Stability analysis:* Recurrent Lyapunov Functions (RLFs)
  - *Safety verification:* Recurrent Barrier functions (RBFs)
- **Self-Improving via Nonparametric Control Policies**
  - Policy Improvement using Expert Demonstrations

Outline

- **Relaxing Invariance: Merits and trade offs**
  - *Recurrent Sets:* Letting thing go and come back
- **Nonparametric Analysis via Recurrent Sets**
  - *Stability analysis:* Recurrent Lyapunov Functions (RLFs)
  - *Safety verification:* Recurrent Barrier functions (RBFs)
- **Self-Improving via Nonparametric Control Policies**
  - Policy Improvement using Expert Demonstrations

Reinforcement Learning



- **Agent:** at time t
  - Receives state s_t and reward r_t
  - Performs action a_t

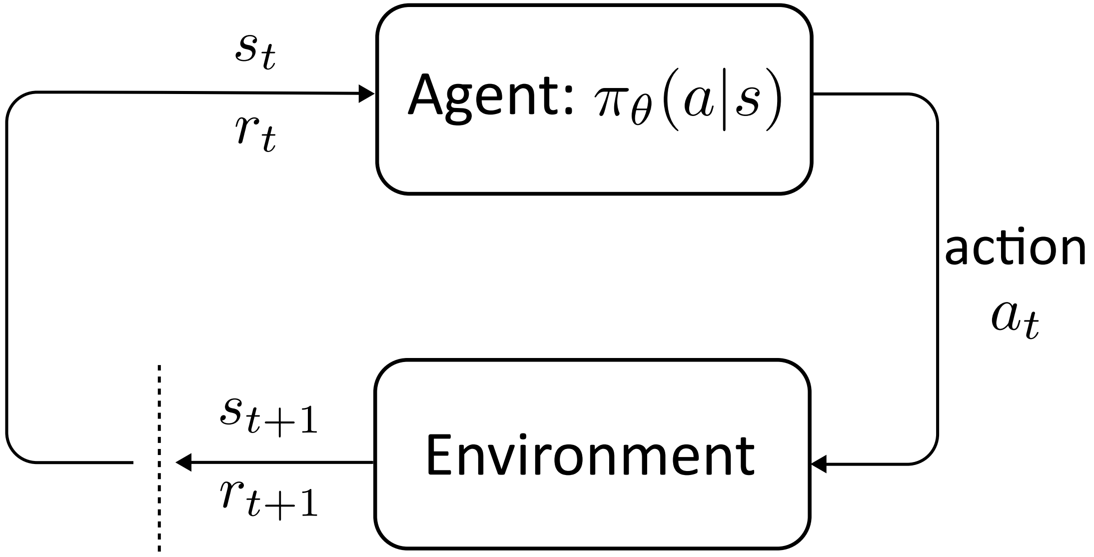
- **Environment:**
  - Receives action a_t
  - Provides state s_{t+1} and reward r_{t+1}

- **Goal:** Find a policy π_θ that maximizes

$$\max_\theta J(\theta) := E_{\pi_\theta, s_0 \sim \rho}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

- **RL Language:**
  - **Value function:** $$V^{\pi_\theta}(s_t) := E_{\pi_\theta}\left[\sum_{t'=t}^{\infty} \gamma^{t'-t} r(s_{t'}, a_{t'})\right]$$

Reinforcement Learning



- **Agent:** at time t
  - Receives state s_t and reward r_t
  - Performs action a_t

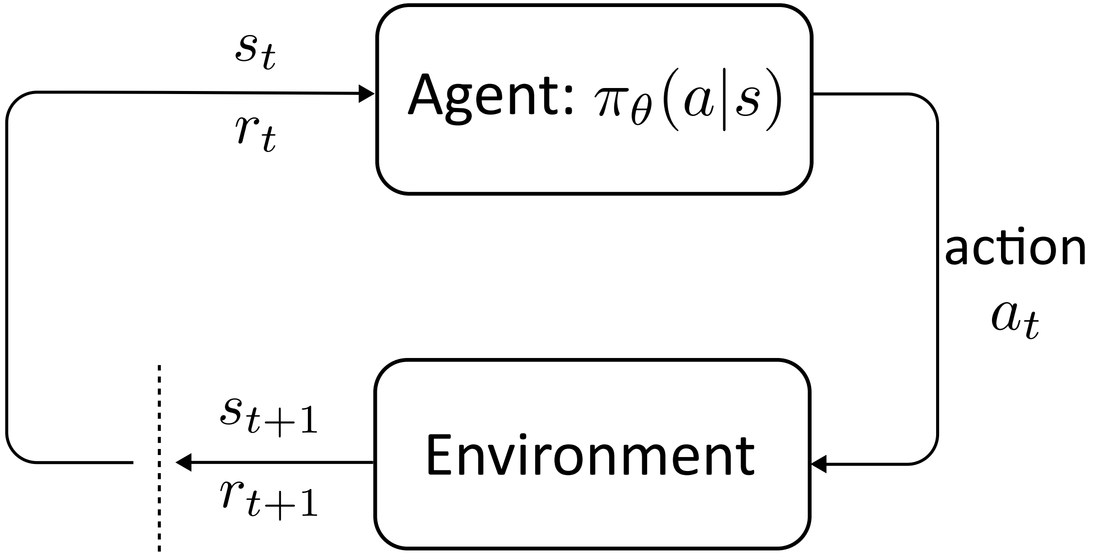
- **Environment:**
  - Receives action a_t
  - Provides state s_{t+1} and reward r_{t+1}

- **Goal:** Find a policy π_θ that maximizes

$$\max_\theta J(\theta) := E_{s_0\sim\rho}\left[V^{\pi_\theta}(s_0)\right]$$

- **RL Language:**
  - **Value function:** $V^{\pi_\theta}(s_t) := E_{\pi_\theta}\left[\sum_{t'=t}^{\infty}\gamma^{t'-t}r(s_{t'},a_{t'})\right]$
  - **Action value function:** $Q^{\pi_\theta}(s_t,a_t) := E_{\pi_\theta}\left[\sum_{t'=t}^{\infty}\gamma^{t'-t}r(s_{t'},a_{t'})\right]$

Reinforcement Learning



- **Agent:** at time t
  - Receives state s_t and reward r_t
  - Performs action a_t
- **Environment:**
  - Receives action a_t
  - Provides state s_{t+1} and reward r_{t+1}
- **Goal:** Find a policy π_θ that maximizes

$$\max_\theta J(\theta) := E_{s_0\sim\rho, a_0\sim\pi_\theta(s_0)}\left[Q^{\pi_\theta}(s_0, a_0)\right]$$

- **RL Language:**
  - **Value function:** $$V^{\pi_\theta}(s_t) := E_{\pi_\theta}\left[\sum_{t'=t}^{\infty} \gamma^{t'-t} r(s_{t'}, a_{t'})\right]$$
  - **Action value function:** $$Q^{\pi_\theta}(s_t, a_t) := E_{\pi_\theta}\left[\sum_{t'=t}^{\infty} \gamma^{t'-t} r(s_{t'}, a_{t'})\right]$$

Classical policy improvement works in discrete spaces

"Policy improvement" is a fundamental building block of classical RL

Policy iteration = Policy evaluation + Policy improvement

Policy evaluation

- Given π, evaluate it to find $Q^\pi(\cdot,\cdot)$

- Can evaluate "separately" for each (s,a)
- Can store Q in a table

Policy improvement

- Given $Q^\pi(\cdot,\cdot)$, define: $\pi' : \mathcal{S} \to \mathcal{A} : \quad \pi'(s) \in \operatorname*{argmax}_{a \in \mathcal{A}} Q^\pi(s,a)$
- Then:

$$V^{\pi'}(s) \geq V^\pi(s) \quad \forall s \in \mathcal{S}$$

- Given s, maximize an array of size $|\mathcal{A}|$

Rinse and repeat until $V^{\pi'} \equiv V^\pi \implies \pi = \pi' = \pi^\star$

Policy Optimization in Continuous Action Spaces

$$\max_{\theta} J(\theta)$$

Based on Policy Gradient:

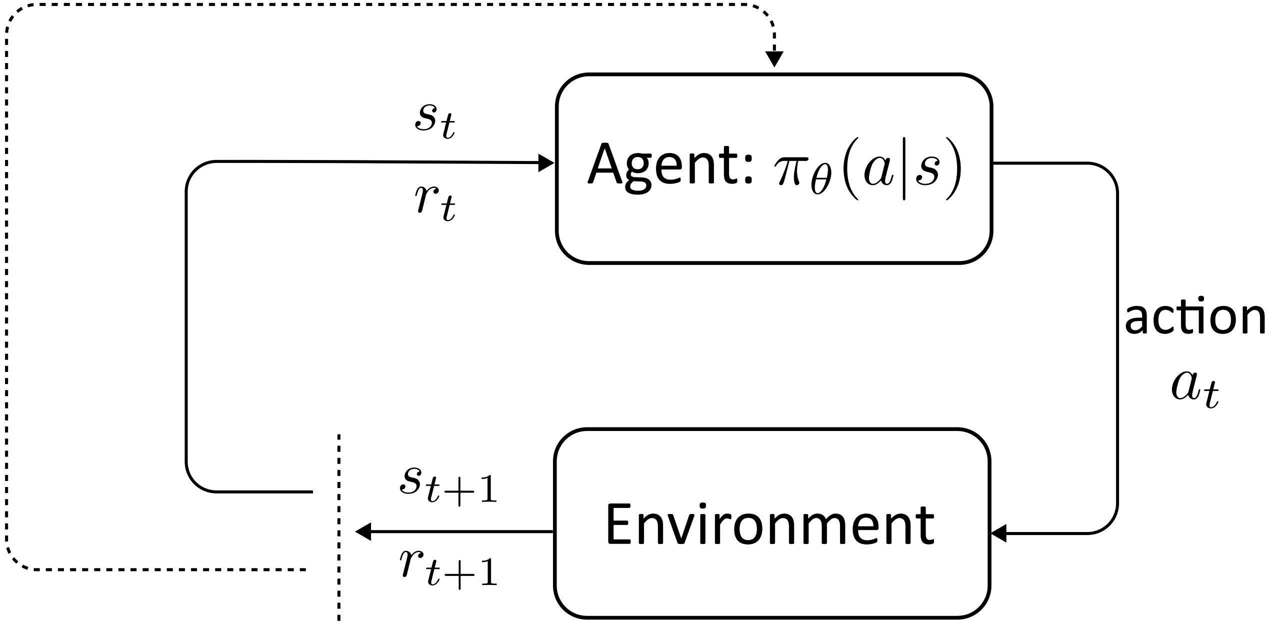
- Use experience to approximate $\nabla_\theta J(\theta) \approx \hat{g}$

$$\hat{g} := \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t^{(i)} \mid s_t^{(i)}) \hat{R}_t^{(i)}$$

$$\hat{R}_t^{(i)} = \sum_{k=t}^{T} \gamma^{k-t} r_k^{(i)}$$

cumulative return

update: $\theta_{k+1} = \theta_k + \eta \, \hat{g}$

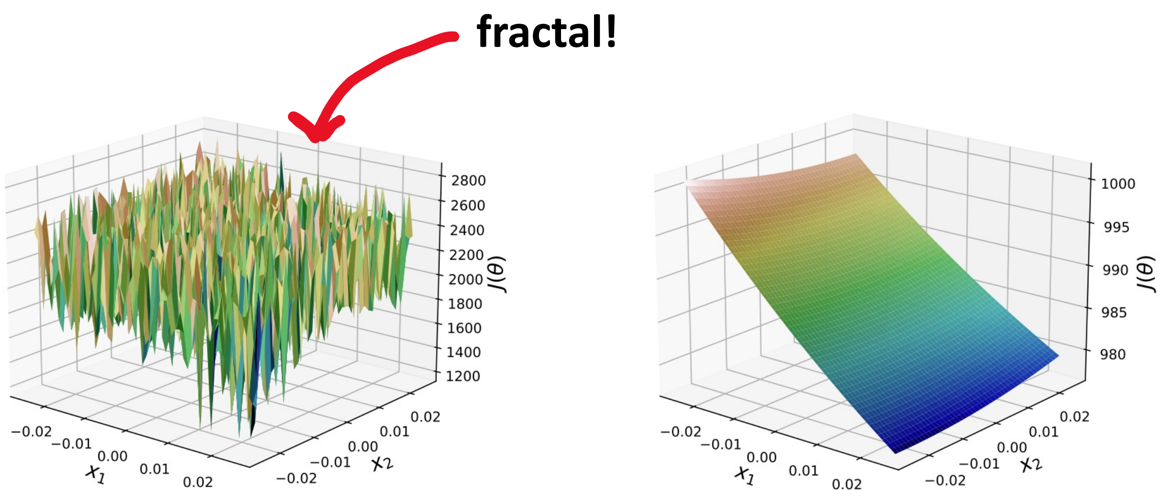


Many Challenges:

- Estimation variance
- Non-smoothness
- Fractal landscape
- Mollification
- …

Fundamental challenges of Policy Optimization

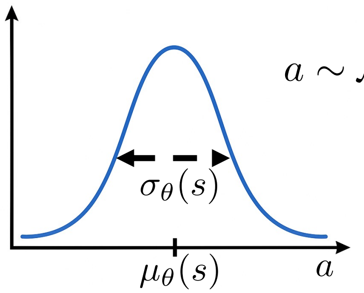
Challenge: Fractal Optimization Landscapes



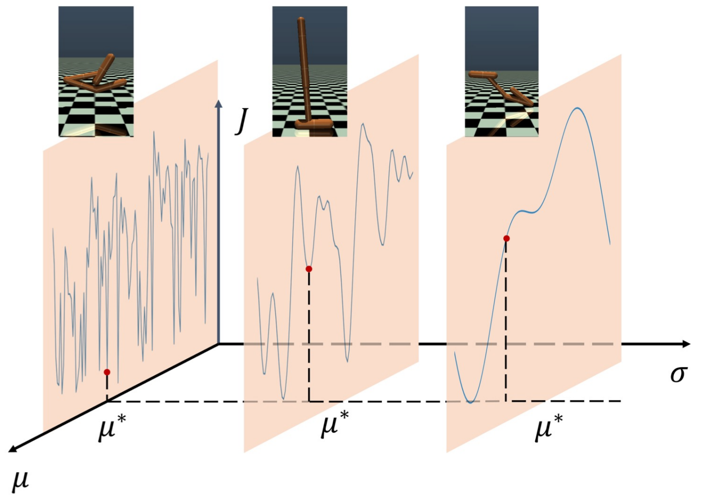
- **Goal:** $\max_\theta \quad J(\theta) := E_{\pi_\theta, s_0\sim\rho}\left[\sum_{t=0}^{\infty} \gamma^t r_t(s_t, a_t)\right]$
- **Approach:** $\theta_{k+1} = \theta_k + \eta\, \hat{\nabla}_\theta J(\theta)$

Challenge: Mollification of Policy Gradient

- **Goal:** $\max_\theta J(\theta) := E_{s_0\sim\rho, a_0\sim\pi_\theta(s_0)}\left[Q^{\pi_\theta}(s_0, a_0)\right]$
- **Policy:**



$a \sim \mathcal{N}(\mu_\theta(s), \sigma_\theta^2 I)$



Tao Wang, Sylvia Hebert, Sicun Gao, Fractal landscapes in policy optimization, NeurIPS 23
Tao Wang, Sylvia Hebert, Sicun Gao, Mollification effects of policy gradient, ICML 24

Agustin Castellano Johns Hopkins University

Sohrab Rezaei Johns Hopkins University

Jared Markowitz Johns Hopkins Applied Physics Laboratory

Enrique Mallada Johns Hopkins University

Nonparametric policy improvement in continuous action spaces

A. Castellano, S. Rezaei, J. Markovitz, and E. Mallada, Nonparametric Policy Improvement for Continuous Action Spaces via Expert Demonstrations, 2025, submitted to Reinforcement Learning Conference.

Problem Setup

Goal: find optimal policy

$$\max_{\theta} J(\theta) := E_{s_0 \sim \rho, a_0 \sim \pi_\theta(s_0)} \left[Q^{\pi_\theta}(s_0, a_0) \right]$$

Problem Setup

Goal: find optimal **nonparametric** policy

$$\max_{\mathcal{D}} J(\pi_{\mathcal{D}}) := E_{s_0\sim\rho, a_0\sim\pi_{\mathcal{D}}(s_0)}\left[Q^{\pi_{\mathcal{D}}}(s_0, a_0)\right]$$

Data set: $\mathcal{D} = \{(s_i, a_i, Q_i)\}_{i=1}^{|\mathcal{D}|}$ $\quad Q_i := \sum_t \gamma^t r(s_t, a_t)$

Assumptions:

Optimal $Q^\star$ **is smooth:** $|Q^\star(s,a) - Q^\star(s',a')| \leq L\big(d_{\mathcal{S}}(s,s') + d_{\mathcal{A}}(a,a')\big)$

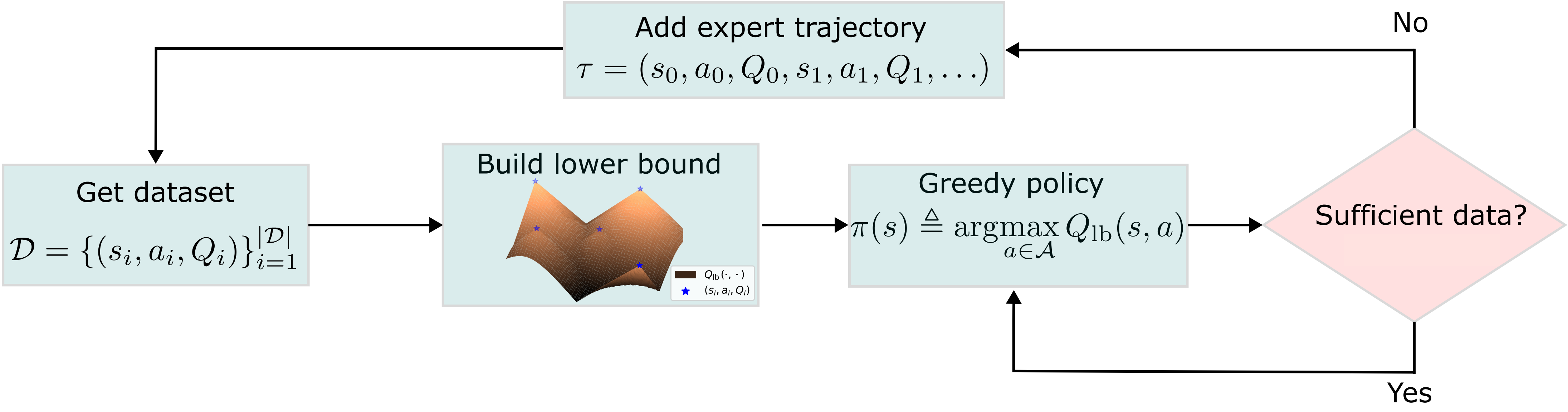
Deterministic dynamics: $s_{t+1} = f(s_t, a_t)$

Expert data: we have $\mathcal{D} = \{(s_i, a_i, Q_i)\}_{i=1}^{|\mathcal{D}|}$, where $a_i = \pi^\star(s_i); Q_i = Q^\star(s_i, a_i)$

Expert data: we have $\mathcal{D} = \{(s_i, a_i, Q_i)\}_{i=1}^{|\mathcal{D}|}$, where $a_i = \pi^\star(s_i); Q_i = Q^\star(s_i, a_i)$

1. **How** can we use these transitions to learn a nonparametric policy?
2. **What** guarantees can we get when we add more transitions?
3. **Where** should we add transitions to improve performance?

Overview of our method



1. How can we use these transitions to learn a nonparametric policy?

Building bounds & Nonparametric Policy

Expert data: we have $\mathcal{D} = \{(s_i, a_i, Q_i)\}_{i=1}^{|\mathcal{D}|}$, where $a_i = \pi^\star(s_i); Q_i = Q^\star(s_i, a_i)$

- Use the data to define **lower** bounds on optimal values:

$$V_{\text{lb}}(s) \triangleq \max_{1\le i\le|\mathcal{D}|} \{Q_i - L \cdot d_{\mathcal{S}}(s, s_i)\} \qquad Q_{\text{lb}}(s,a) \triangleq \max_{1\le i\le|\mathcal{D}|} \{Q_i - L \cdot \left(d_{\mathcal{S}}(s, s_i) + d_{\mathcal{A}}(a, a_i)\right)\}$$

- **Nonparametric Policy:**

$$\pi(s) \triangleq \operatorname*{argmax}_{a\in\mathcal{A}} Q_{\text{lb}}(s,a) = a_{i'}$$

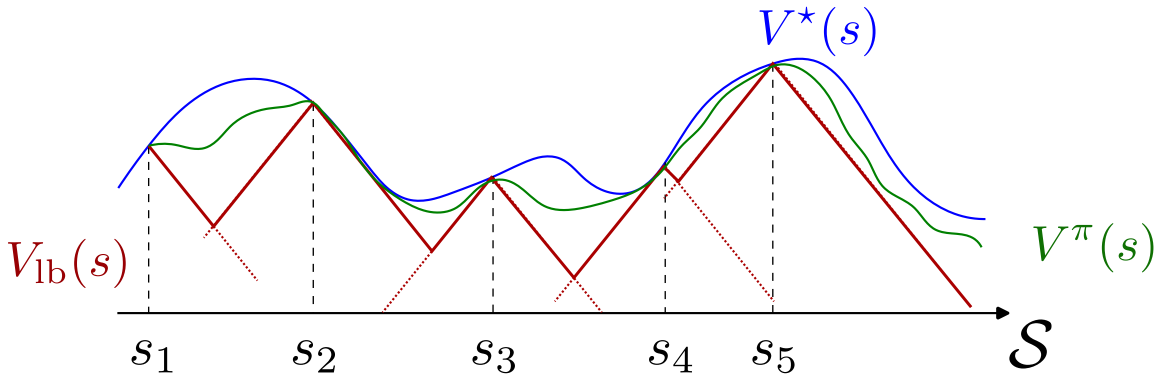
- **Remark:** Note argmax always gives actions in dataset $(s_{i'}, a_{i'}, Q_{i'})$
- **Question:** What can we say about $V^\pi(s)$?

Nonparametric policy *improves* over lower bound

Policy Evaluation:

- Nonparametric π satisfies $\forall s \in \mathcal{S}$:

$$V_{\text{lb}}(s) \le V^{\pi}(s) \le V^{\star}(s)$$



Policy Improvement:

- Given data sets $\mathcal{D}, \mathcal{D}'$ with $\mathcal{D} \subset \mathcal{D}'$

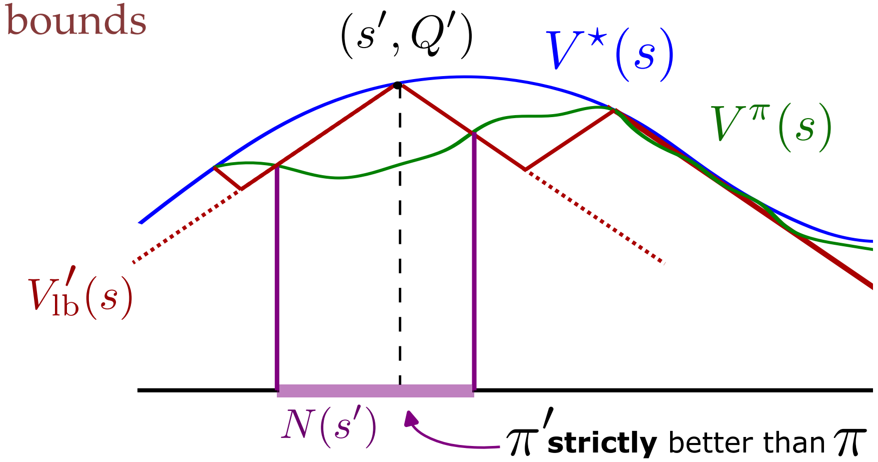
$$V_{\text{lb}}(s) \le V'_{\text{lb}}(s) \quad \forall s \in \mathcal{S}$$

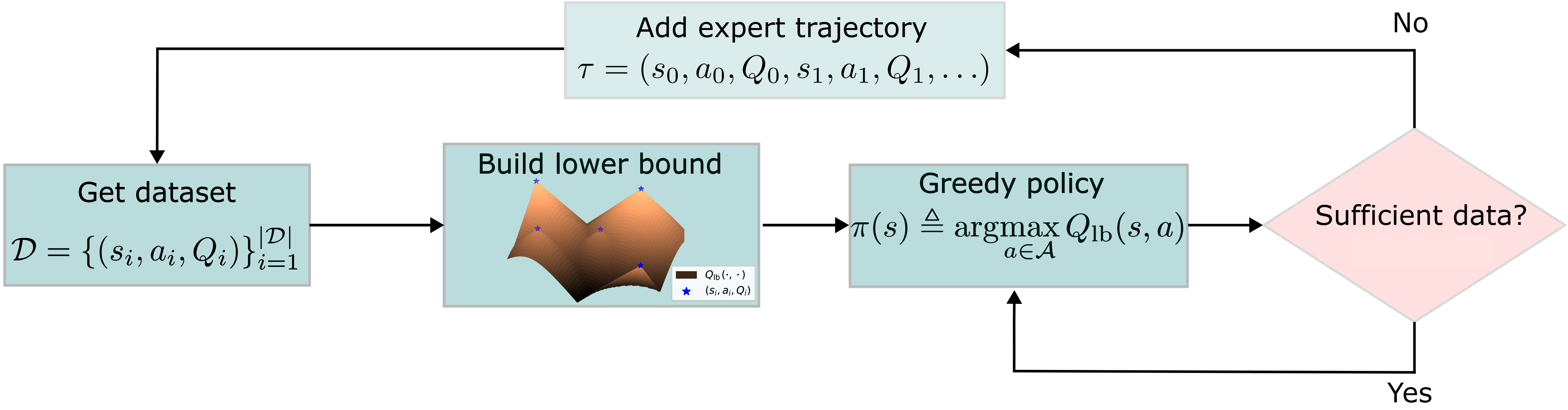
More data = better lower bounds

$$V^{\pi}(s') \le V^{\pi'}(s') \quad \forall s' \in \mathcal{D}' \backslash \mathcal{D}$$

Improvement on added points

- **Strict** on neighbors of new data: $\forall s \in N(s')$



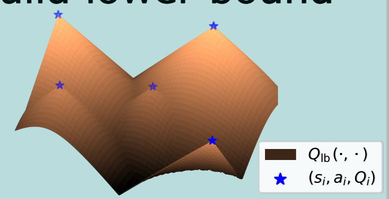

Add expert trajectory
$\tau = (s_0, a_0, Q_0, s_1, a_1, Q_1, \ldots)$
No
Get dataset
$\mathcal{D} = \{(s_i, a_i, Q_i)\}_{i=1}^{|\mathcal{D}|}$
Build lower bound
$Q_{\mathrm{lb}}(\cdot, \cdot)$
(s_i, a_i, Q_i)
Greedy policy
$\pi(s) \triangleq \operatorname{argmax}_{a \in \mathcal{A}} Q_{\mathrm{lb}}(s, a)$
Sufficient data?
Yes

1. **How** to learn a policy?

Get dataset

$\mathcal{D} = \{(s_i, a_i, Q_i)\}_{i=1}^{|\mathcal{D}|}$

→ Build lower bound



→ Greedy policy

$\pi(s) \triangleq \operatorname{argmax}_{a\in\mathcal{A}} Q_{\rm lb}(s,a)$

→ Policy Evaluation

$V_{\rm lb}(s) \leq V^\pi(s)$

2. **What** guarantees with more transitions?

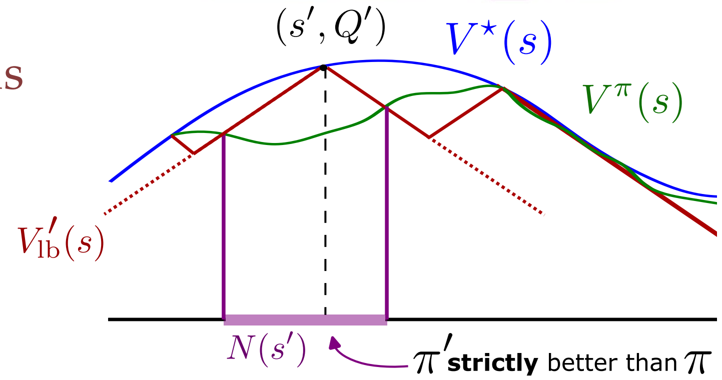
$$V_{\rm lb}(s) \leq V'_{\rm lb}(s) \qquad \forall s \in \mathcal{S}$$

More data = better lower bounds

$$V^\pi(s') \leq V^{\pi'}(s') \qquad \forall s' \in \mathcal{D}'\backslash\mathcal{D}$$

Improvement on added points

strict improvement



3. **Where** to add transitions?

- Only *where* **sufficient improvement** is guaranteed: $\Delta(s) := V_{\rm ub}(s) - V_{\rm lb}(s) > \varepsilon$

$$V_{\rm lb}(s) \leq V^\pi(s) \leq V^\star(s) \leq V_{\rm ub}(s)$$

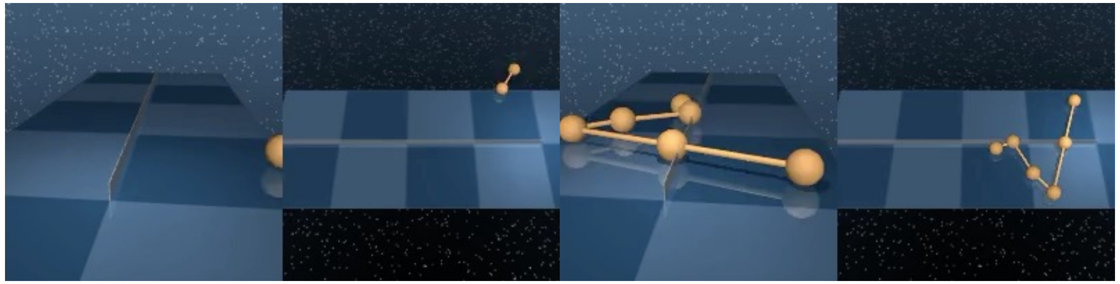
Experiments

Number of balls

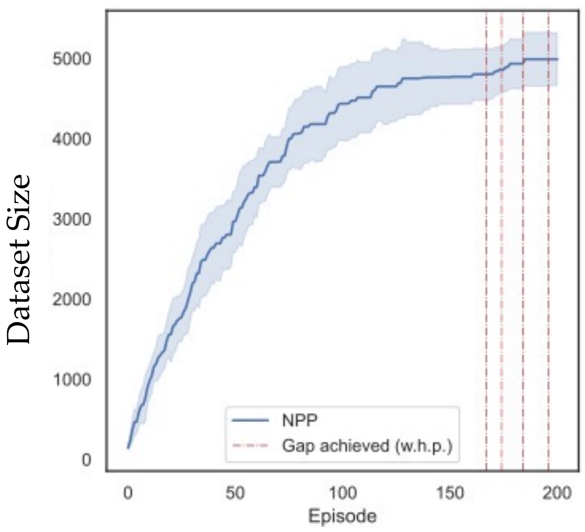
1st m actuated

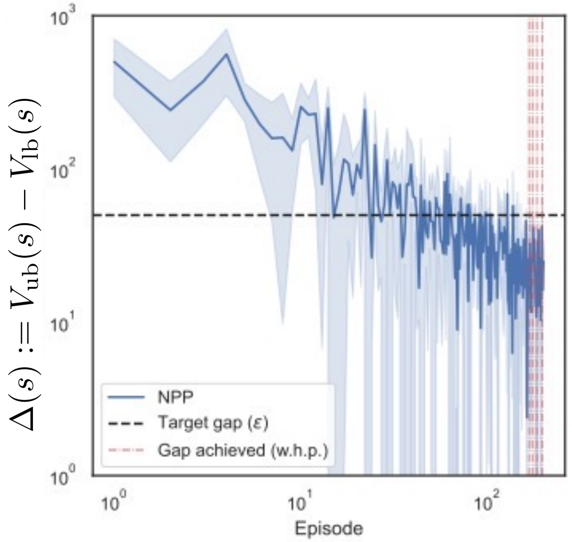
- We use the `lqr_n_m` environments from DeepMind's Control Suite

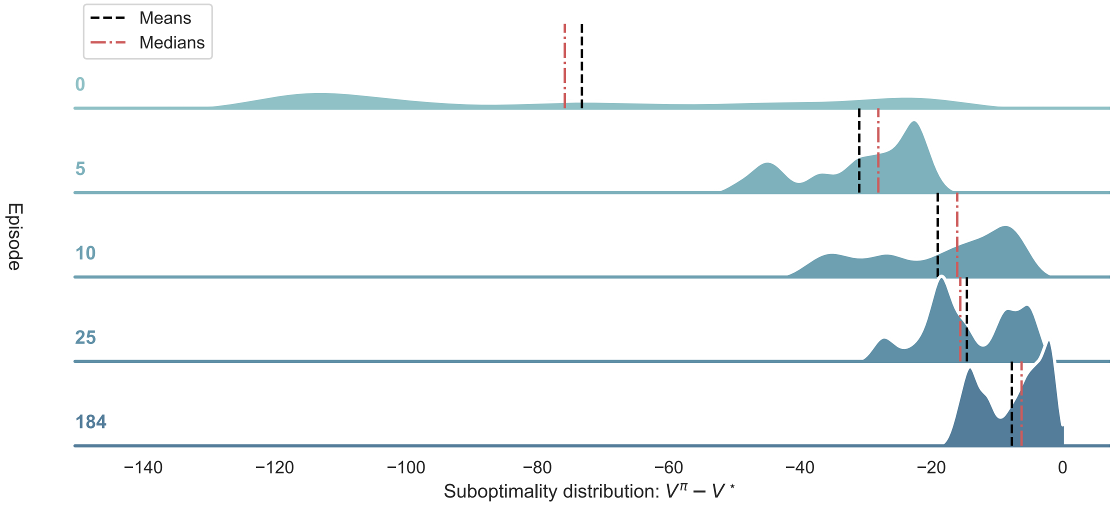
`lqr_2_1` `lqr_6_2`



- Results on **`lqr_2_1`**:

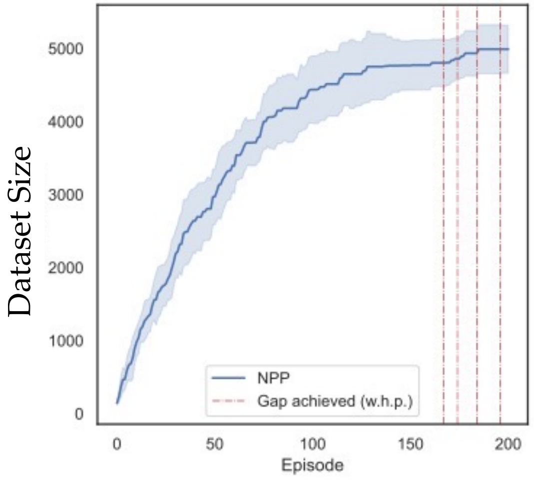


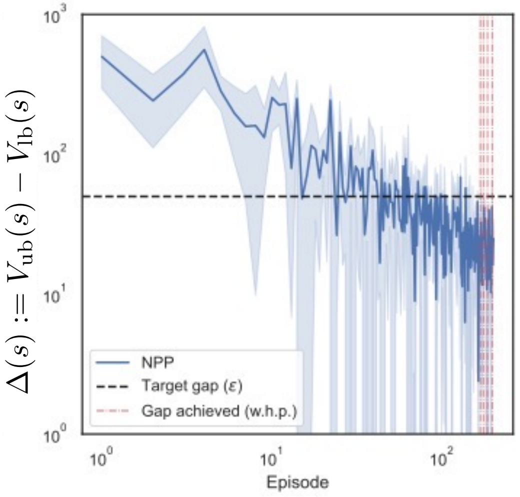


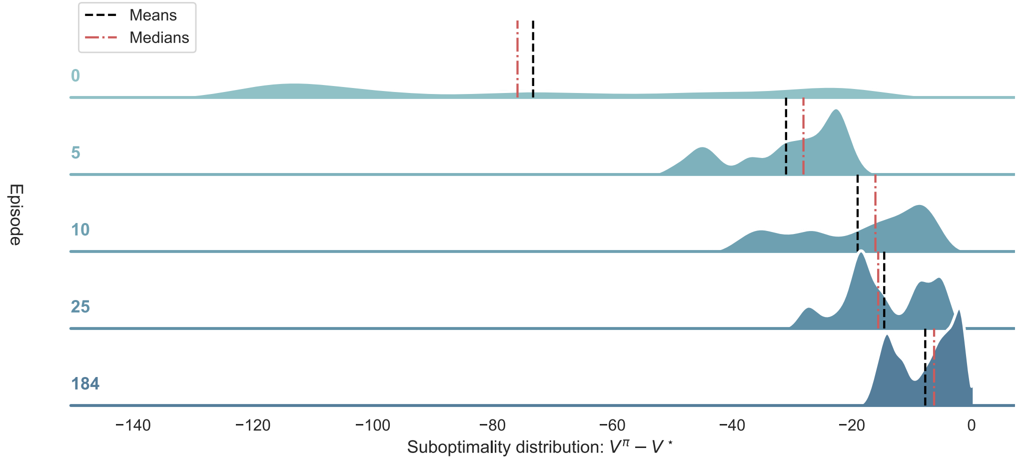


Experiments

- We use the `lqr_n_m` environments from DeepMind's Control Suite
- Results on **`lqr_2_1`**:



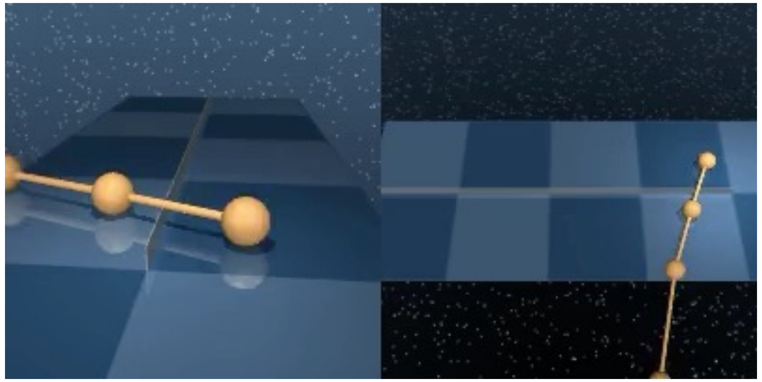




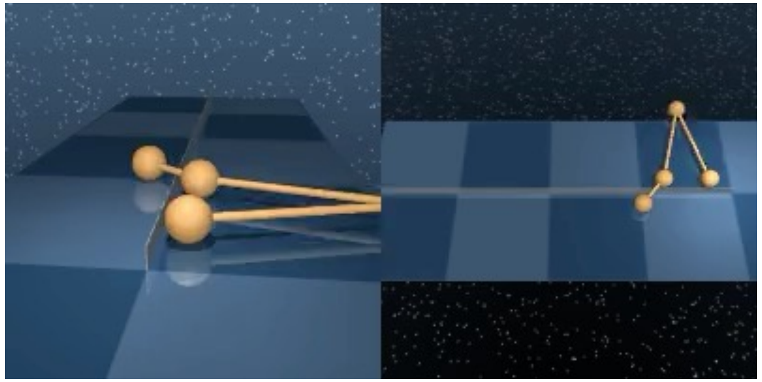
- **Remarks:**
  - **Incremental learning:** No catastrophic forgetting, or oscillations
  - Improvement across the entire state space (not in expectation)
  - Only valuable data is added (harder to find at times passes)

Incremental Learning

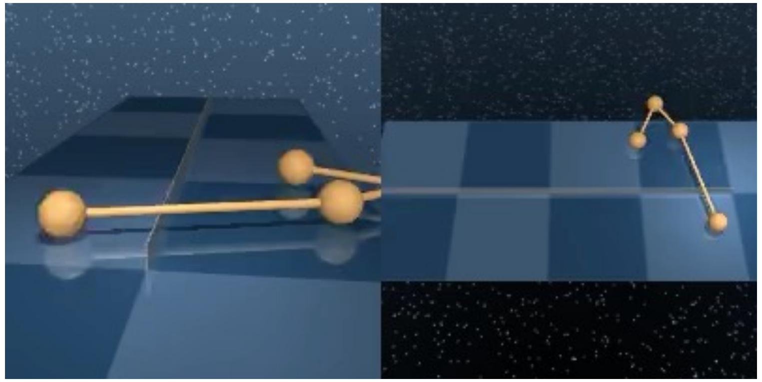
after 10 episode…



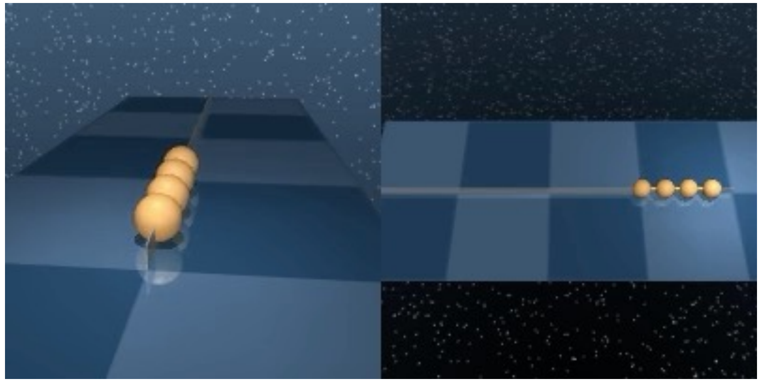
after 100 episode…



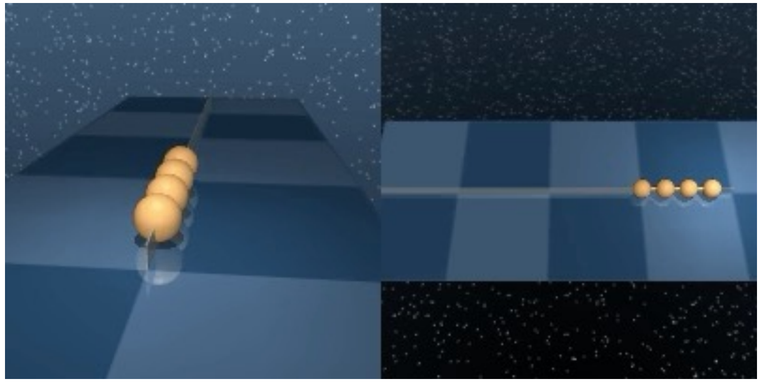
after 1000 episodes…



after 30K+



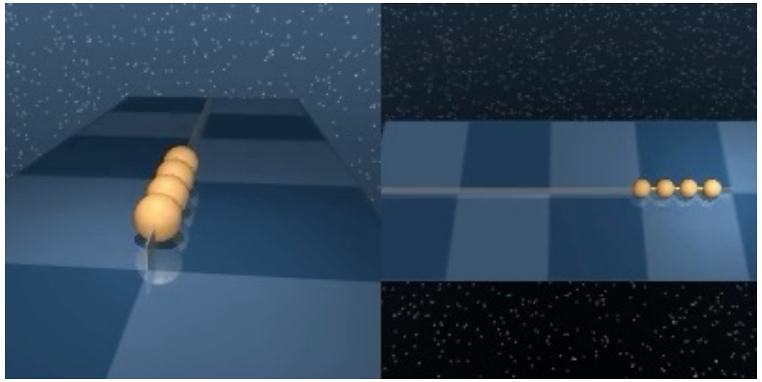
optimal control

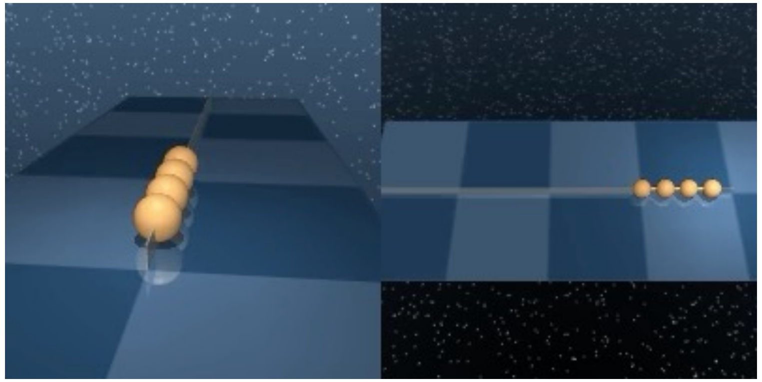


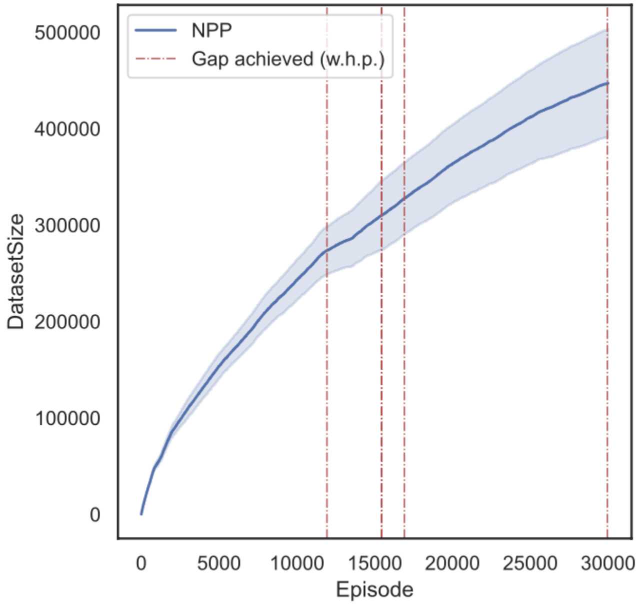
Incremental Learning

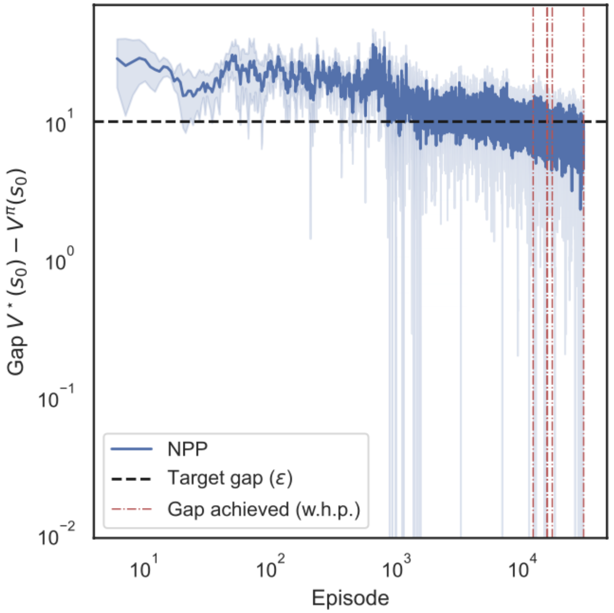
after 30K+

optimal control







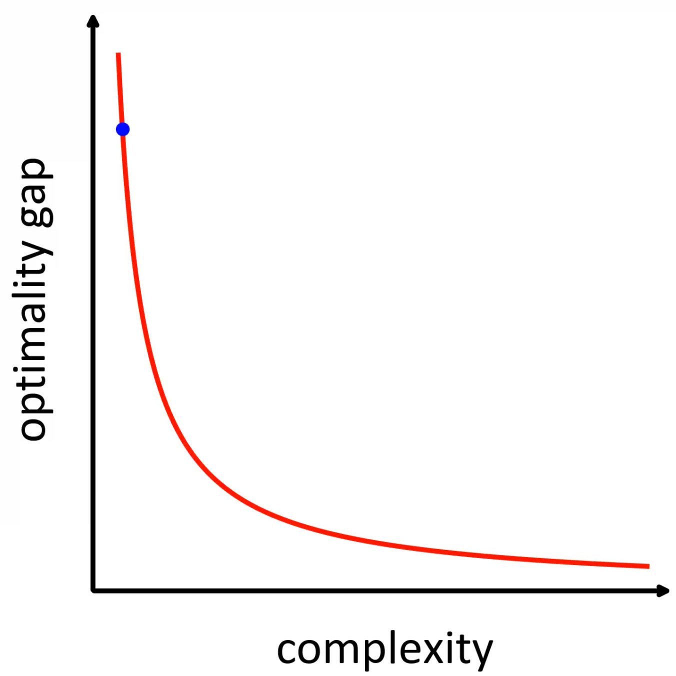


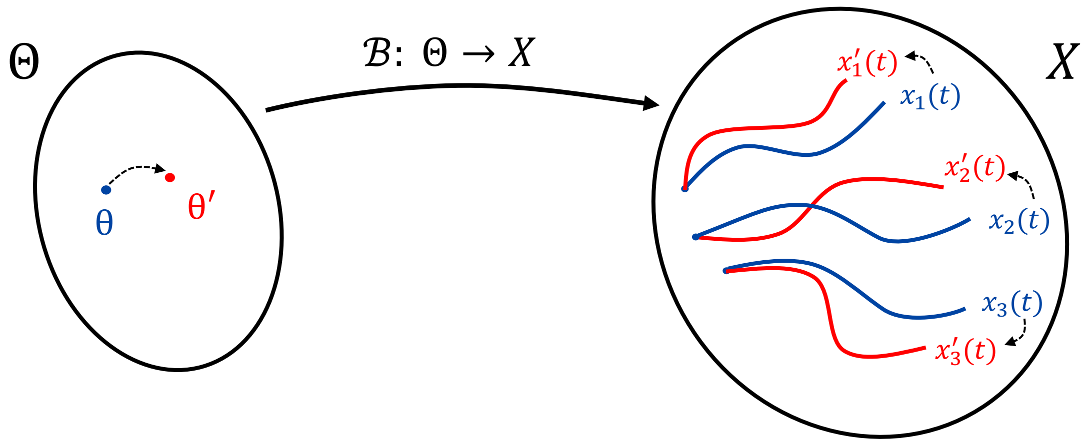
Research Goals

- To develop analysis and design methods that *trade off complexity and performance*.

- To allow for *continual improvement*, without the need for redesign, retune, or retrain

- To design control policies with controlled sensitivity to parameter changes





Conclusions and Future work

- **Takeaways**
  - Proposed a **relaxed notion of invariance: recurrence.**
  - **Nonparametric theory for dynamical systems analysis** leading to:
    - General **Lyapunov** and **Barrier Function** conditions **satisfied by any norm!**
    - Algorithms that are **parallelizable and progressive/sequential.**
  - **Nonparametric policies:** Guaranteed improvement with each demonstration.

- **Ongoing work**
  - **Recurrence:** Information theoretical lower bounds of control recurrence sets
  - **Lyapunov/CBF Theory:** Generalize other Lyapunov notions, Control Lyapunov Functions, Control Barrier Functions, Contraction
  - **Nonparametric policies (NP):** NP policy iteration, enforcing safety and stability using NP, exploring alternative inductive biases (beyond Lipschitz)

Thanks!

Related Publications:

[CDC 23] Siegelmann, Shen, Paganini, M, *A recurrence-based direct method for stability analysis and GPU-based verification of non-monotonic Lyapunov functions*, **CDC 2023**, **TAC submitted**
[HSCC 24] Sibai, M, *Recurrence of nonlinear control systems: Entropy and bit rates,* **HSCC, 2024, NAHS under review**
[Allerton 24] Shen, Sibai, M, *Generalized Barrier Functions: Integral conditions and recurrent relaxations,* **Allerton 2024**
[RLC 25] Castellano, Rezaei, Markovitz, and M, Nonparametric Policy Improvement for Continuous Action Spaces via Expert Demonstrations, **RLC 2025**

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