

# A Geometric Approach for Learning Reach Sets

**Shadi Haddad**

Department of Applied Mathematics  
University of California, Santa Cruz

[shhaddad@ucsc.edu](mailto:shhaddad@ucsc.edu)

Committee: Profs. Qi Gong, Hongyun Wang, Abhishek Halder (Advisor)

**Dissertation Defense, December 06, 2023**

# Overview

## Part 1: Analytical and Semi-analytical Computation of Reach Sets:

L-CSS 2024; TAC 2023; SVAA 2023; L-CSS 2022; ACC 2022, 2020

Integrators with time invariant input set

Integrators with time varying input set

Controllable LTI systems

Differentially flat nonlinear systems

## Part 2: Data-Driven Learning of Compact Sets:

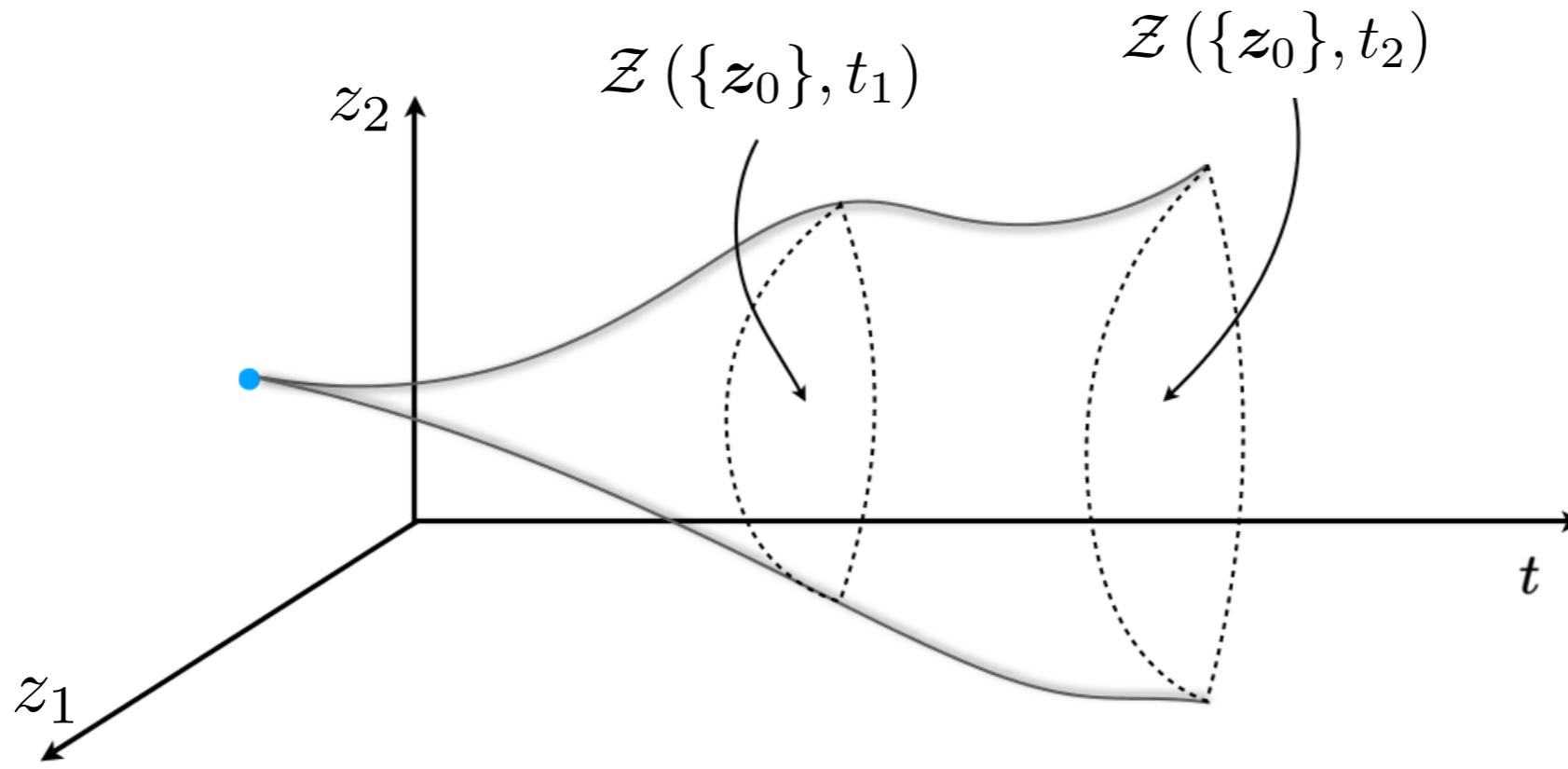
ACC 2023; TCST 2022; CPS IoT 2021; L-CSS 2020

Region of attraction

Maximal control invariant set

Reach sets of neural networks, nonlinear control systems

# Reach set: Definition



**Controlled dynamics**

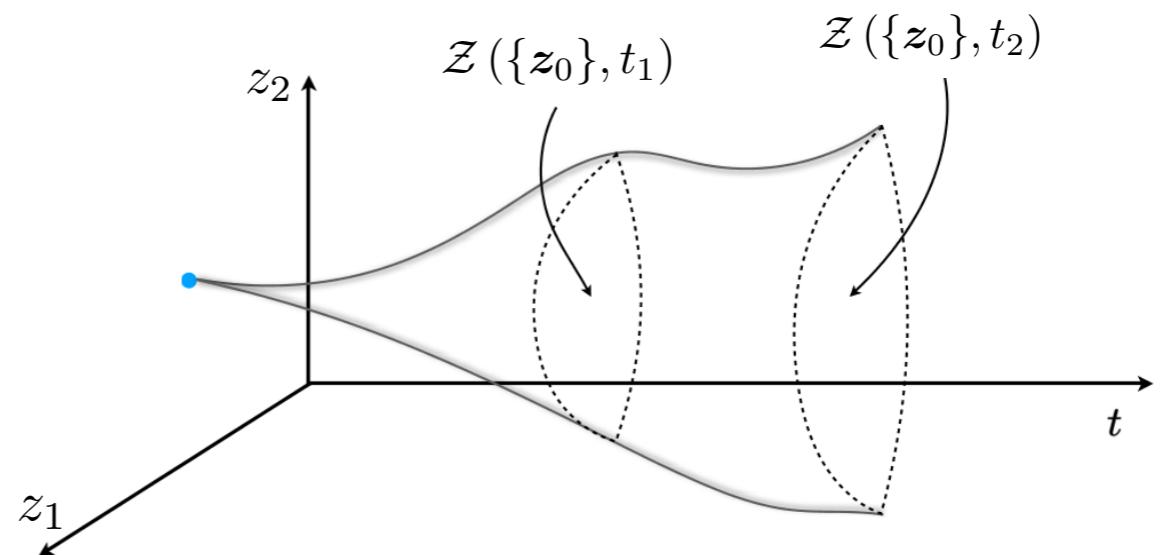
$$\dot{z} = f(z, v), \quad z(t=0) \in \mathbb{R}^{d_z}, \quad v \in \mathcal{V} \subset \mathbb{R}^m$$

**Forward reach set at time  $t$**

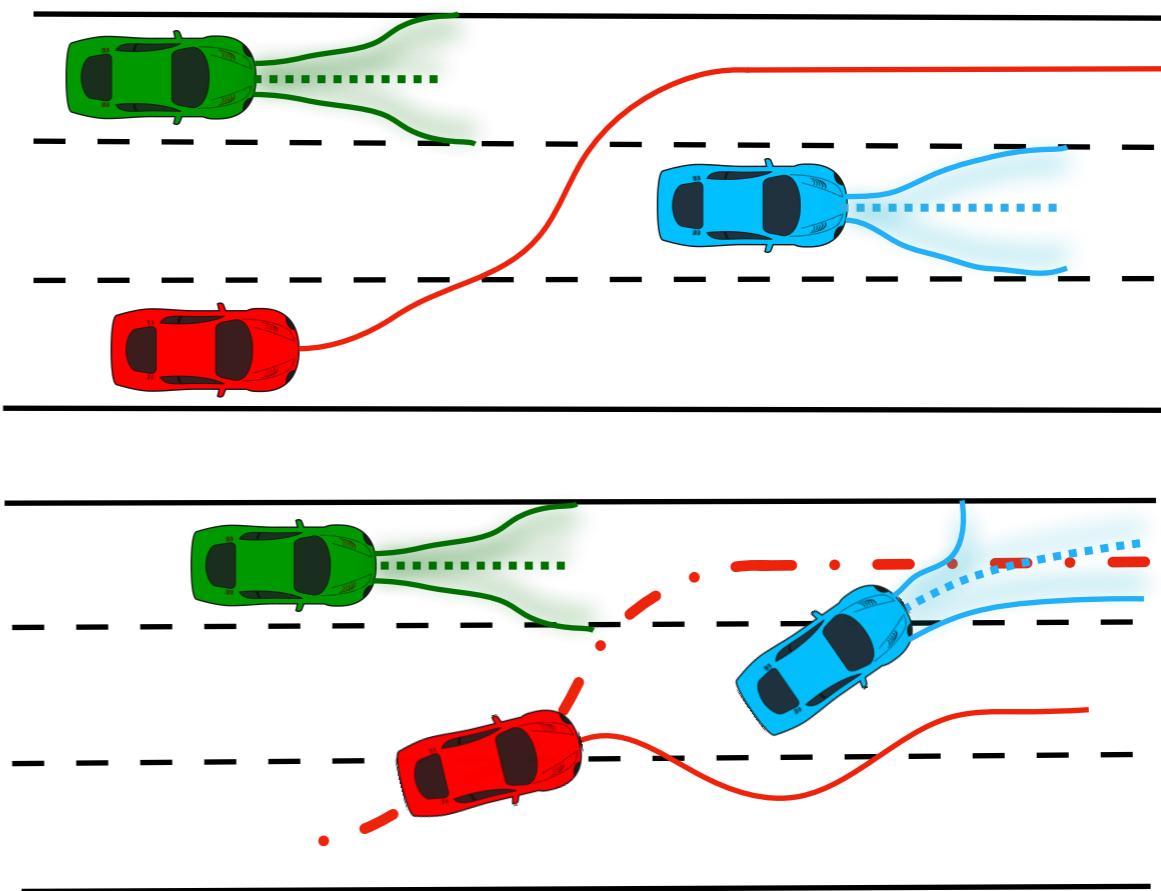
$$\mathcal{Z}_t := \bigcup_{\text{measurable } v(\cdot) \in \mathcal{V}} \left\{ z(t) \in \mathbb{R}^{d_z} \mid \dot{z} = f(z, v), \quad z(t=0) \in \mathbb{R}^{d_z}, \right. \\ \left. v(\tau) \in \mathcal{V} \text{ for all } 0 \leq \tau \leq t \right\}.$$

# Reach set: Applications

Predicting the states of an uncertain system

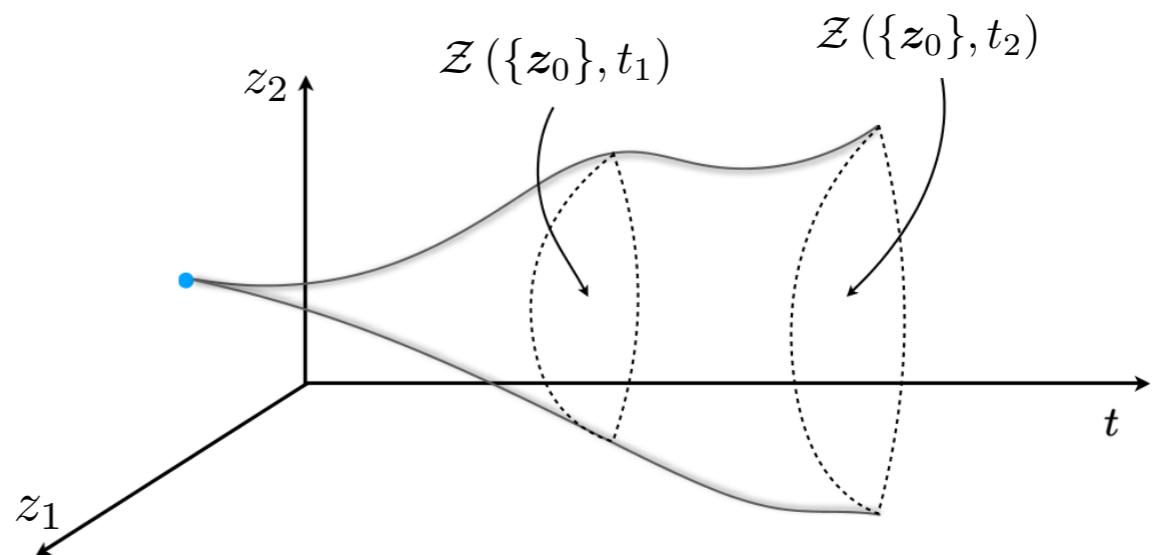


Safety critical applications such as motion planning & collision warning systems

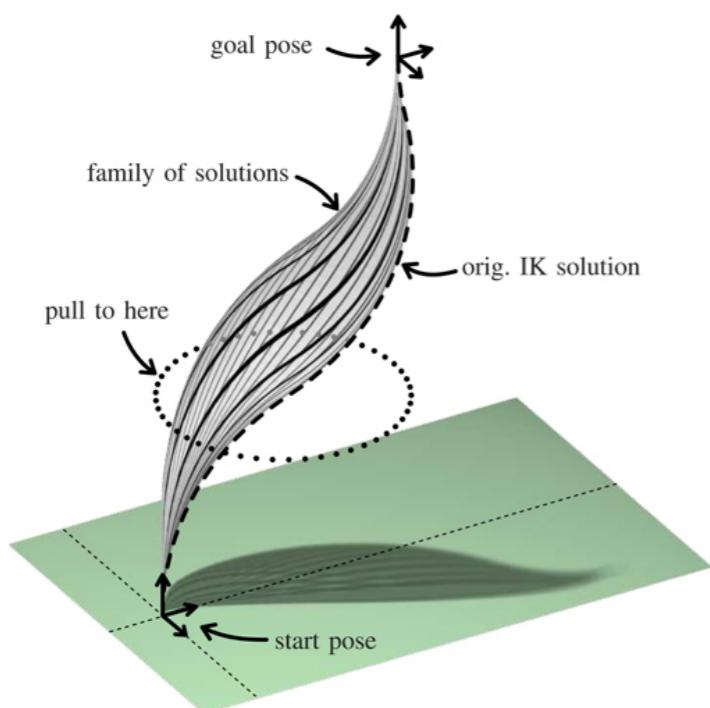


# Reach set: Applications

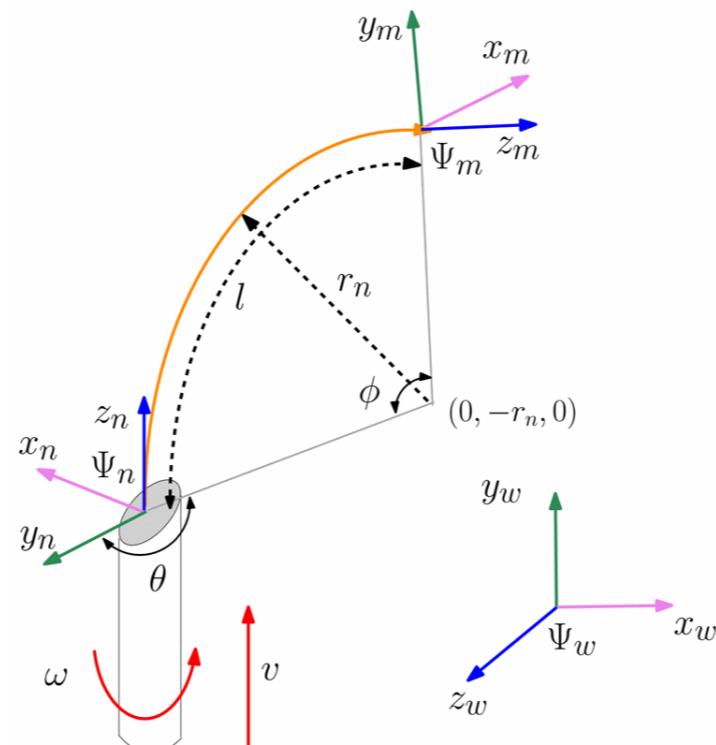
Predicting the states of an uncertain system



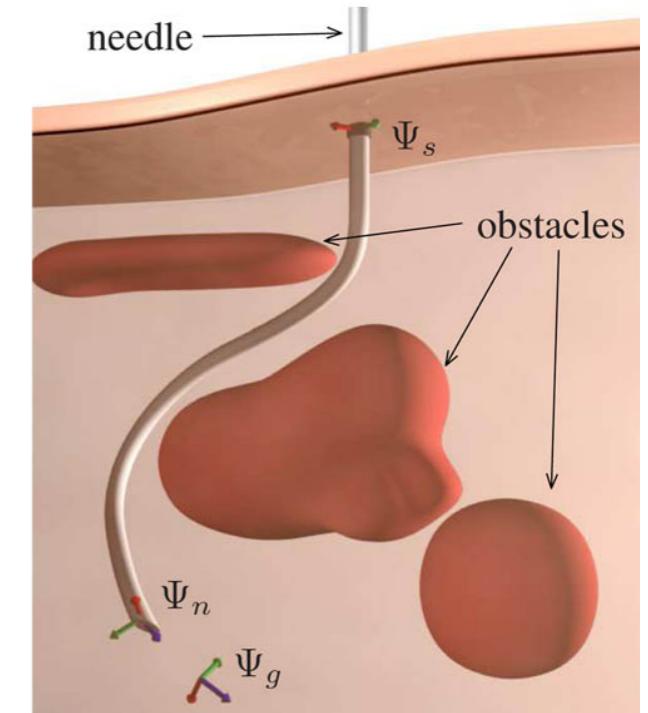
Needle steering w. input uncertainties



Credit: Duindam *et al.*, 2009



Credit: Patil and Alterovitz, 2010

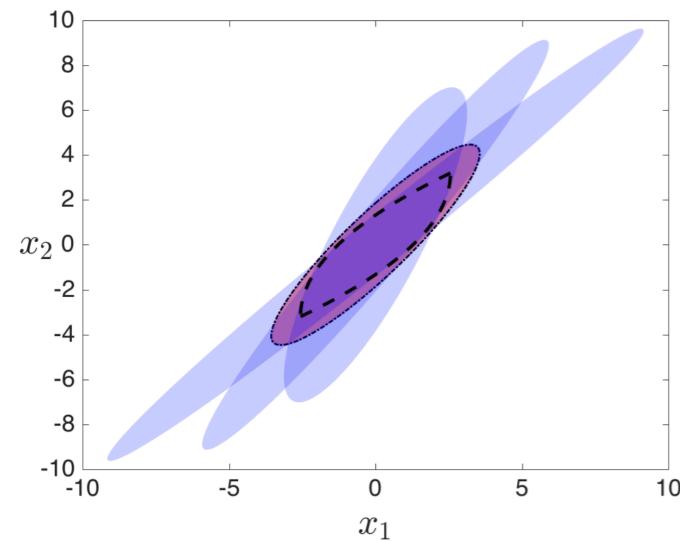


Credit: Duindam *et al.*, 2009

# Existing algorithms for reach set computation

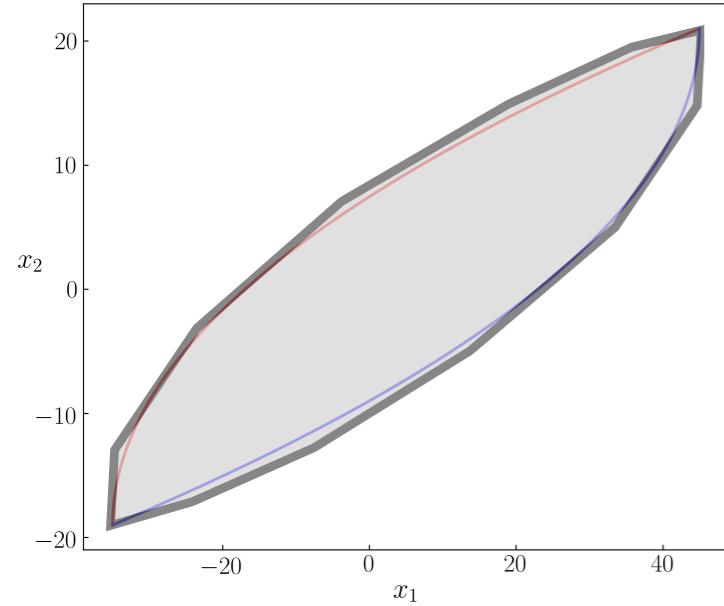
## Parametric

Ellipsoidal over-approximation



**Elipsoidal toolbox**  
[Kurzhansky et al., 2006]

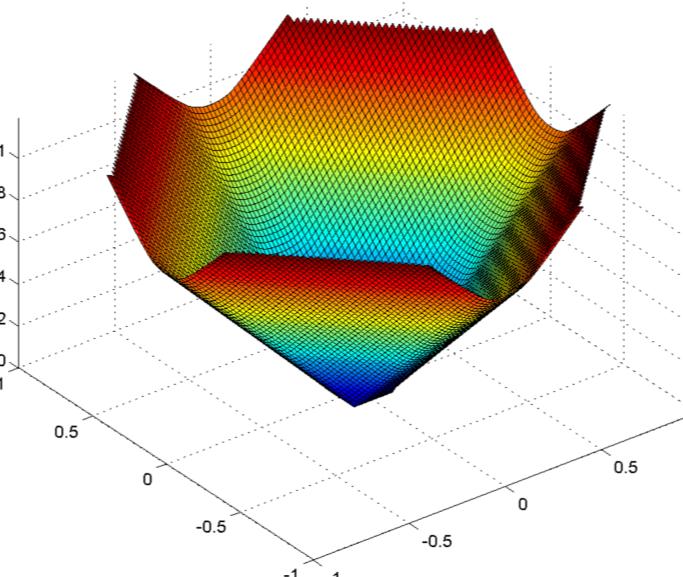
Zonotopic over-approximation



**CORA toolbox**  
[Althoff et al., 2015]

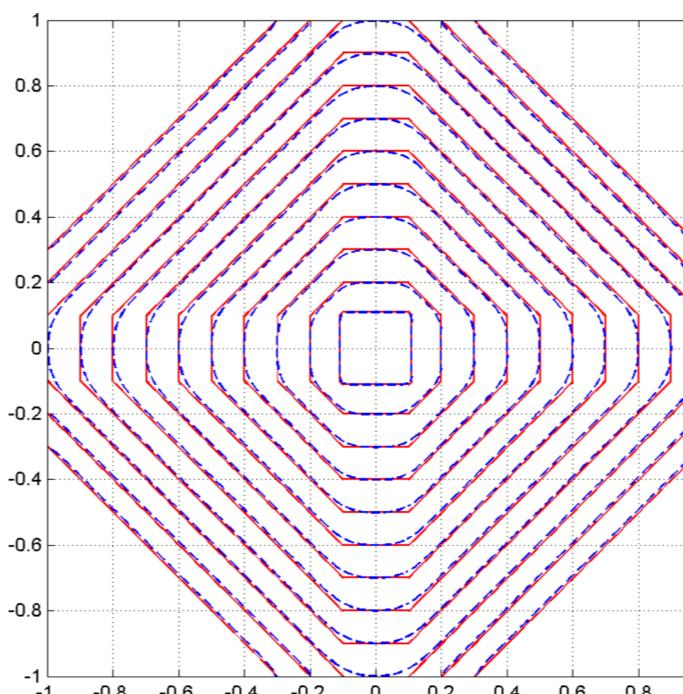
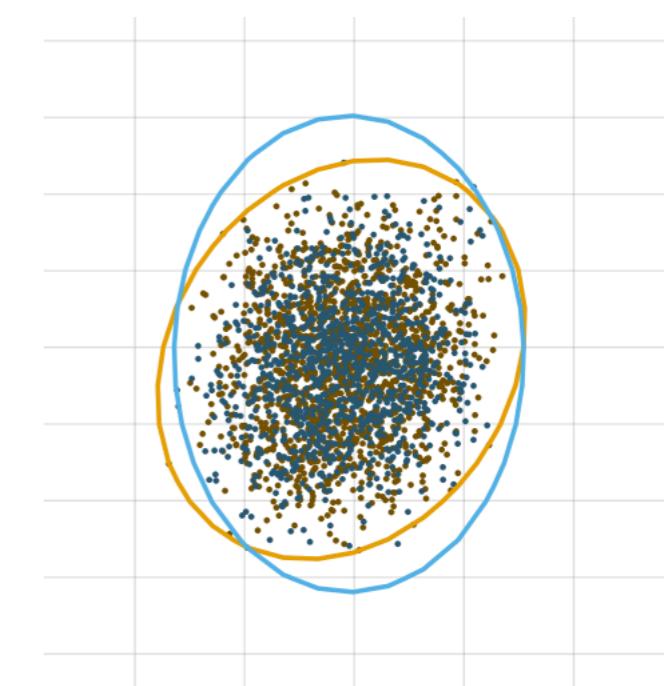
## Nonparametric

Zero sub-level set of the viscosity solution of HJB PDE

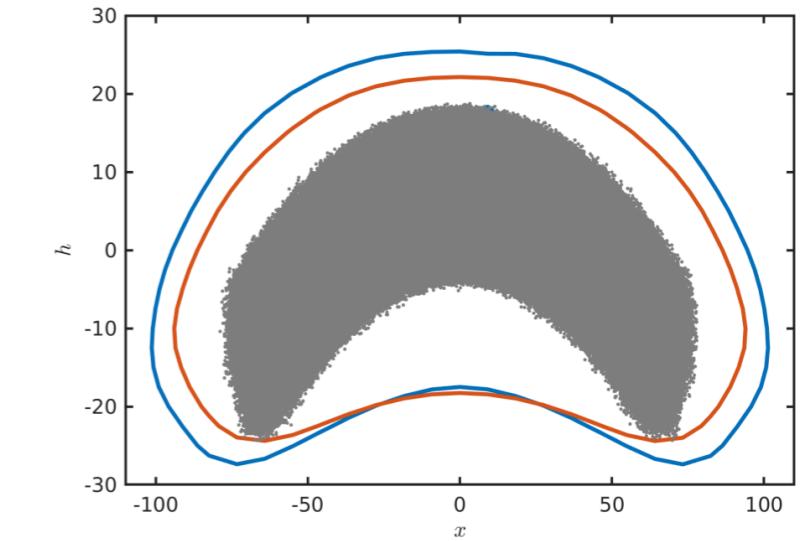


## Semiparametric

Sample-based statistical learning



**Level set toolbox**  
[Mitchell et al., 2008]



**[Devonport and Arcak, 2020]**

# Existing algorithms for reach set computation

No specific algebraic or topological results about the ground truth

Difficult to quantitatively compare performance between two given algorithms

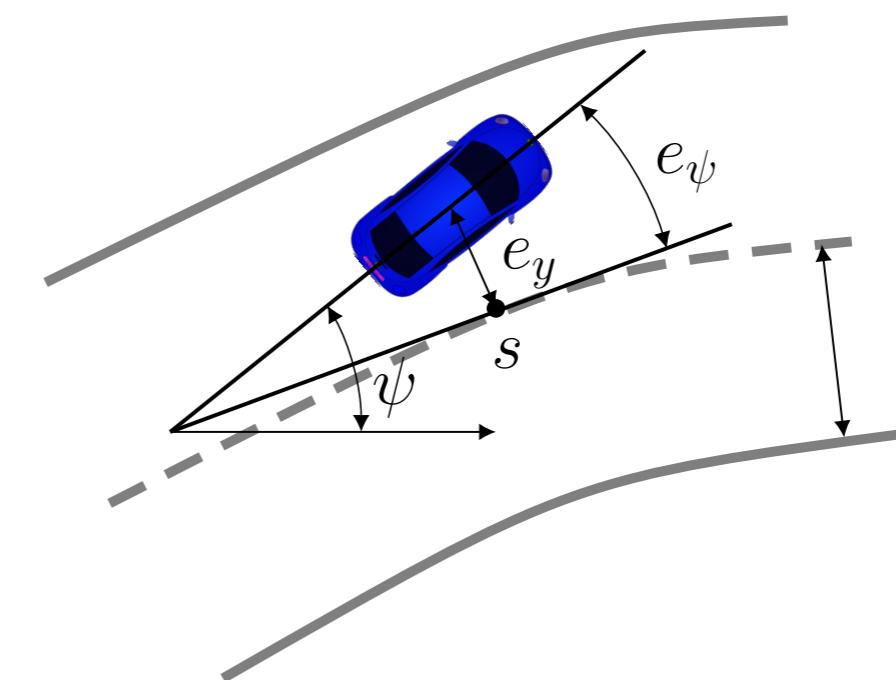
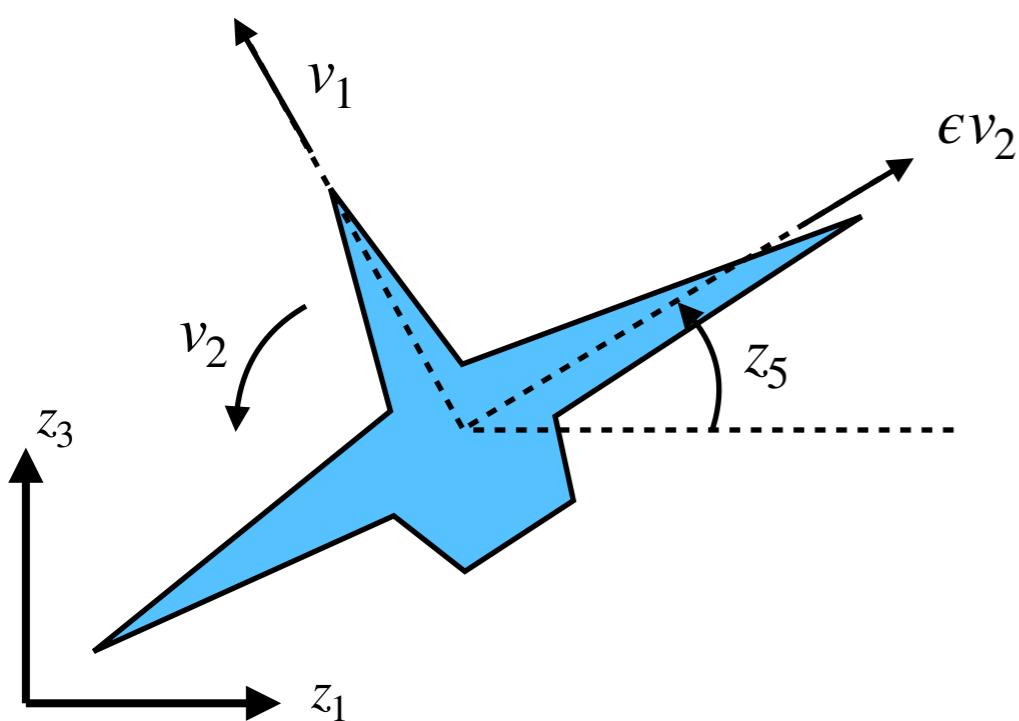
One-size-fits-all algorithms ignore the specific geometry induced by different class of systems

# Our approach

Generic  $\longrightarrow$  specific algorithm exploiting geometry of the true set

## Overall contribution

Algorithms for learning the reach sets of full state (static and dynamic) feedback linearizable systems



# Background: Static state feedback linearizable systems

$$\dot{\mathbf{z}} = \mathbf{f}(\mathbf{z}, \mathbf{v}), \quad \mathbf{z} \in \mathbb{R}^{d_z}, \quad \mathbf{v} \in \mathcal{V} \subseteq \mathbb{R}^m$$

**State diffeomorphism:**  $\tau : \mathbf{z} \in \mathbb{R}^{d_z} \rightarrow \mathbf{x} \in \mathbb{R}^d, \quad d = d_z$

**Input homeomorphism:**  $\tau_u : (\mathbf{z}, \mathbf{v}) \in \mathbb{R}^{d_z} \times \mathcal{V} \rightarrow \mathbf{u} \in \mathcal{U}(t) \subset \mathbb{R}^m$

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{x}(t) \in \mathbb{R}^d, \quad \mathbf{u}(t) \in \mathcal{U}(t) \subset \mathbb{R}^m,$$

**Integrator dynamics a.k.a  
Brunovsky normal form**

$$\mathbf{A} := \text{blkdiag}(\mathbf{A}_1, \dots, \mathbf{A}_m), \quad \mathbf{B} := \text{blkdiag}(\mathbf{b}_1, \dots, \mathbf{b}_m),$$

$$\mathbf{A}_j := \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}_{r_j \times r_j}, \quad \mathbf{b}_j := \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}_{r_j \times 1}, \quad \begin{aligned} r_1 + r_2 + \dots + r_m &= d \\ \mathbf{r} &= (r_1, r_2, \dots, r_m)^\top \in \mathbb{Z}_+^m, \end{aligned}$$

# Example 1: Static state feedback linearizable system

Single link manipulator dynamics with flexible joints and negligible damping:

$$\begin{aligned}\dot{z}_1 &= z_2, & \dot{z}_3 &= z_4, \\ \dot{z}_2 &= -\sin(z_1) - (z_1 - z_3), & \dot{z}_4 &= (z_1 - z_3) + v, \quad z \in \mathbb{R}^4 \text{ and } v \in \mathcal{V} \subset \mathbb{R}.\end{aligned}$$

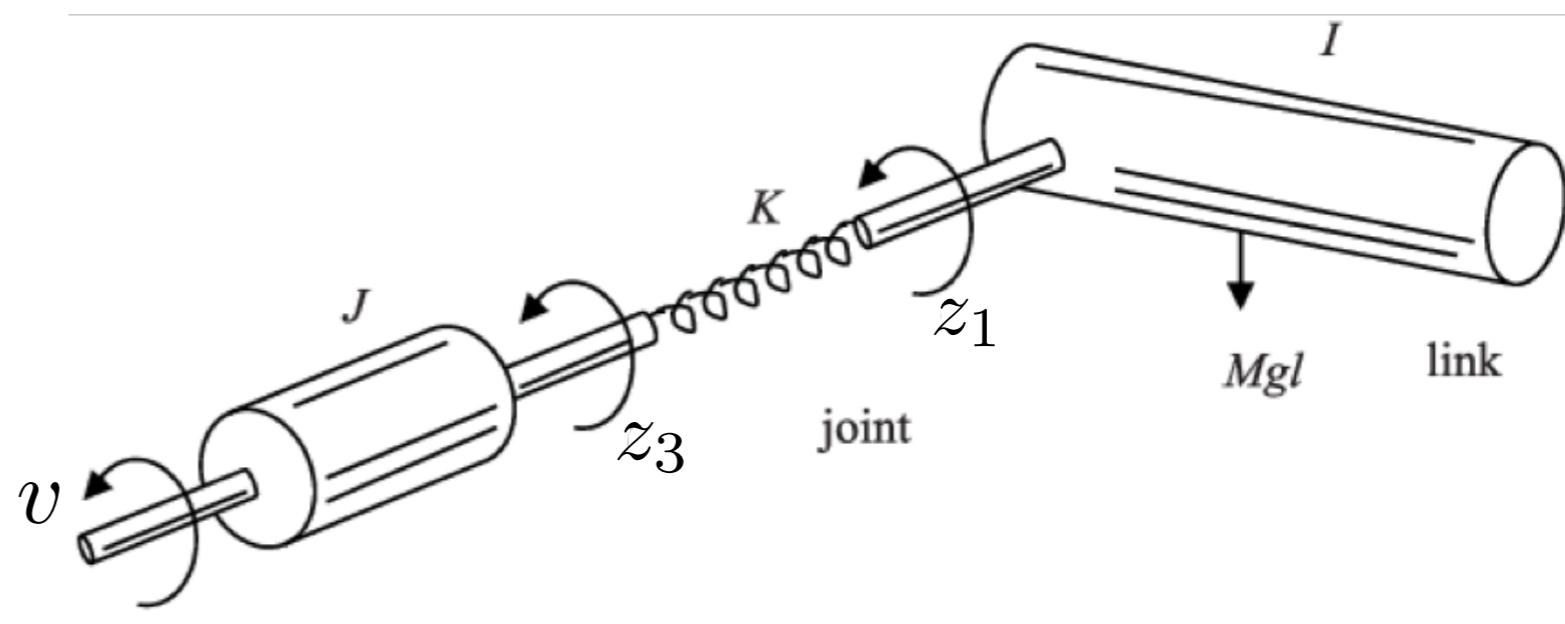
Diffeomorphism  $\tau$  and homeomorphism  $\tau_u$ :

$$x = \tau(z) = \begin{bmatrix} z_1 \\ z_2 \\ -\sin(z_1) - (z_1 - z_3) \\ -z_2 \cos(z_1) - (z_2 - z_4) \end{bmatrix}, \quad z = \tau^{-1}(x) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 + \sin(x_1) + x_1 \\ x_4 + x_2 \cos(x_1) + x_2 \end{bmatrix},$$

$$u = \tau_u(z, v) = -(\cos(z_1) + 2)(-\sin(z_1) + z_3 - z_1) + (z_2^2 - 1) \sin(z_1) + v.$$

Normal form:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u.$$



# Example 2: Static state feedback linearizable system

**System with 5 states and 2 inputs:**

$$\begin{aligned}\dot{z}_1 &= z_2 + z_2^2 + v_1, \\ \dot{z}_2 &= z_3 - z_1 z_4 + z_4 z_5, & \dot{z}_4 &= z_5, \\ \dot{z}_3 &= z_2 z_4 + z_1 z_5 - z_5^2 + \cos(z_1 - z_5) v_1 + v_2, & \dot{z}_5 &= z_2^2 + v_2, & \mathbf{z} \in \mathbb{R}^5 \text{ and } \mathbf{v} \in \mathcal{V} \subset \mathbb{R}^2.\end{aligned}$$

**Diffeomorphism  $\tau$  and homeomorphism  $\tau_u$ :**

$$\mathbf{x} = \boldsymbol{\tau}(\mathbf{z}) = \begin{bmatrix} z_1 - z_5 \\ z_2 \\ z_3 - z_1 z_4 + z_4 z_5 \\ z_4 \\ z_5 \end{bmatrix}, \quad \mathbf{z} = \boldsymbol{\tau}^{-1}(\mathbf{x}) = \begin{bmatrix} x_1 + x_5 \\ x_2 \\ x_3 + (x_1 + x_5)x_4 - x_4 x_5 \\ x_4 \\ x_5 \end{bmatrix},$$

$$\mathbf{u} = \boldsymbol{\tau}_u(\mathbf{z}, \mathbf{v}) = \begin{bmatrix} \cos(z_1 - z_5) v_1 + v_2 \\ z_2^2 + v_2 \end{bmatrix}.$$

**Normal form:**

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

# Background: Dynamic state feedback linearizable systems

$$\dot{\mathbf{z}} = \mathbf{f}(\mathbf{z}, \mathbf{v}), \quad \mathbf{z} \in \mathbb{R}^{d_z}, \quad \mathbf{v} \in \mathcal{V} \subseteq \mathbb{R}^m$$

Compensator state  $w$

Augmented  
state vector

State diffeomorphism:  $\tau : \rho \in \mathbb{R}^{d_z + d_w} \rightarrow \mathbf{x} \in \mathbb{R}^d, \quad d = d_z + d_w, \quad \rho := (\mathbf{z}, \mathbf{w})$

Input homeomorphism:  $\tau_u : (\mathbf{v}, \mathbf{z}, \mathbf{w}, \dot{\mathbf{w}}, \ddot{\mathbf{w}}, \dots) \mapsto \mathbf{u} \in \mathcal{U}(t) \subset \mathbb{R}^m$

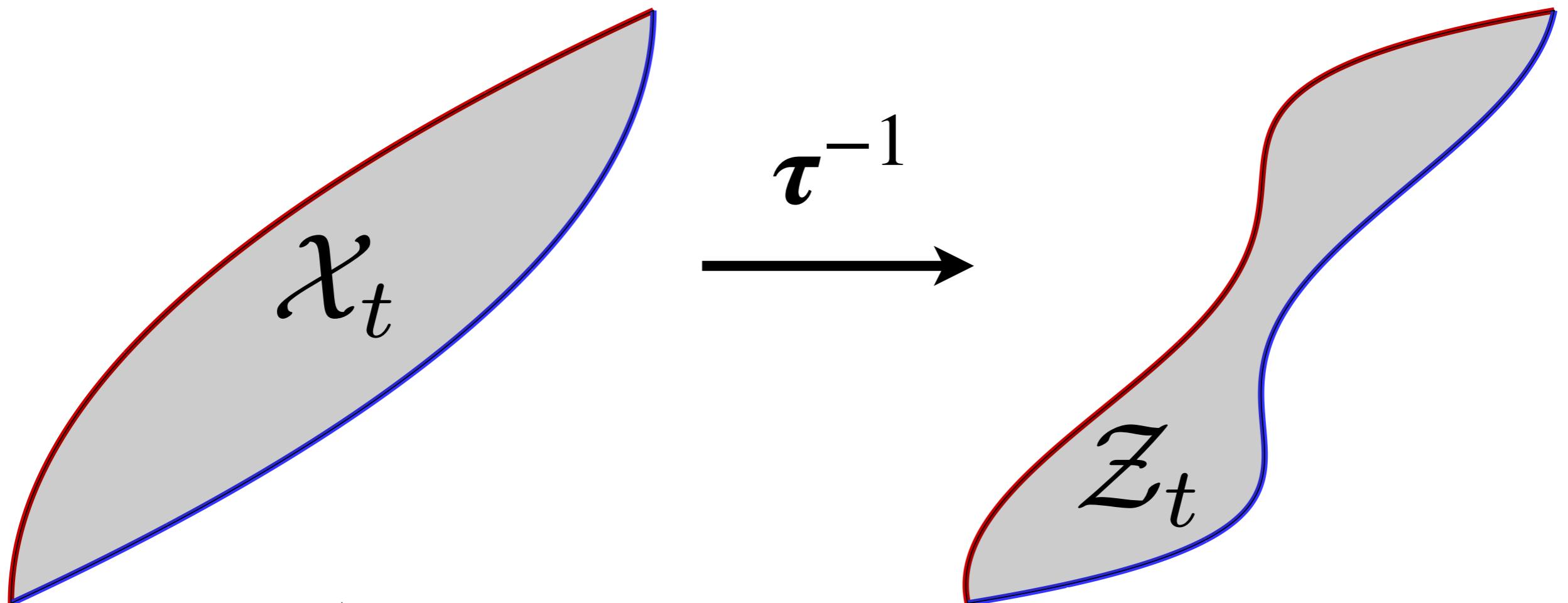
$$\mathbf{u}(t) = \mathbf{C}(\mathbf{z}(t), \mathbf{w}(t), \dot{\mathbf{w}}, \ddot{\mathbf{w}}, \dots) \mathbf{v}(t) + \mathbf{d}(\mathbf{z}(t), \mathbf{w}(t), \dot{\mathbf{w}}, \ddot{\mathbf{w}}, \dots),$$

$$\dot{\mathbf{w}}(t) = \phi(\mathbf{z}, \mathbf{w}, \mathbf{v}, \dot{\mathbf{v}}, \ddot{\mathbf{v}}, \dots), \quad \forall t \geq 0$$

# Main idea

Compute the reach set and its functionals in normal coordinate  $x$

Map them back to original coordinate  $z$  via known diffeomorphism



No prior results on the exact geometry

We use a geometric approach

# Overview

## Part 1: Analytical and Semi-analytical Computation of Reach Sets:

L-CSS 2024; TAC 2023; SVAA 2023; L-CSS 2022; ACC 2022, 2020

Integrators with time invariant input set

Integrators with time varying input set

Controllable LTI systems

Differentially flat nonlinear systems

## Part 2: Data-Driven Learning of Compact Sets:

ACC 2023; TCST 2022; CPS IoT 2021; L-CSS 2020

Region of attraction

Maximal control invariant set

Reach sets of neural networks, nonlinear control systems

# Overview

## Part 1: Analytical and Semi-analytical Computation of Reach Sets:

L-CSS 2024; TAC 2023; SVAA 2023; L-CSS 2022; ACC 2022, 2020

### Integrators with time invariant input set

- Support function
- Parametric formula
- Implicit formula
- Taxonomy
- Size
- Benchmarking

# Integrator Reach Set

$$\mathcal{X}(\mathcal{X}_0, t) := \bigcup_{\substack{\text{measurable } \mathbf{u} \in \\ \text{closure(conv}(\mathcal{U}))}} \left\{ \mathbf{x}(t) \in \mathbb{R}^d \mid \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \right. \\ \left. \mathbf{x}(t=0) \in \mathcal{X}_0, \mathbf{u}(t) \in \mathcal{U} \right\}.$$

These reach sets are in general, **compact and convex**

Input Set  $\mathcal{U}$        $\alpha_j := \min_{\mathbf{u} \in \mathcal{U}} u_j, \quad \beta_j := \max_{\mathbf{u} \in \mathcal{U}} u_j, \quad j = 1, \dots, m,$

Box-valued input       $\mathcal{U} := [\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \times \dots \times [\alpha_m, \beta_m] \subset \mathbb{R}^m$

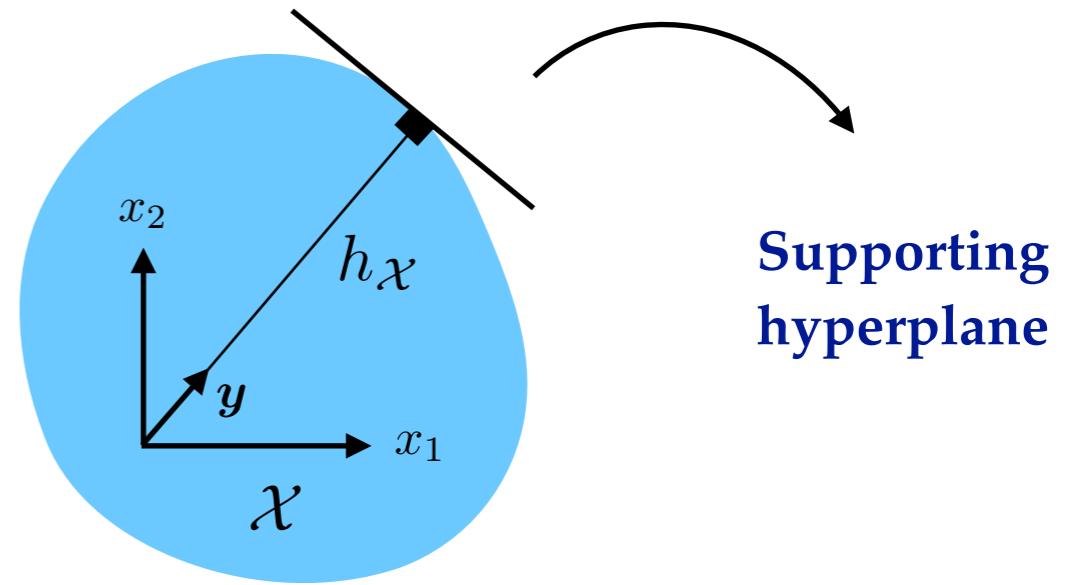
Then       $\mathcal{X}(\mathcal{X}_0, t) = \mathcal{X}_1(\mathcal{X}_{10}, t) \dot{+} \mathcal{X}_2(\mathcal{X}_{20}, t) \dot{+} \dots \dot{+} \mathcal{X}_m(\mathcal{X}_{m0}, t)$

Minkowski sum

Single input integrator reach  
set corresponding to  $u_1$

# Support function

$$h_{\mathcal{X}(\mathcal{X}_0, t)}(\mathbf{y}) := \sup_{\mathbf{x} \in \mathcal{X}} \left\{ \langle \mathbf{x}, \mathbf{y} \rangle \mid \mathbf{y} \in \mathbb{S}^{d-1} \right\}$$



**Theorem.**

$$h_{\mathcal{X}(\mathcal{X}_0, t)}(\mathbf{y}) = \sum_{j=1}^m \left\{ \sup_{\mathbf{x}_{j0} \in \mathcal{X}_{j0}} \langle \mathbf{y}_j, \exp(tA) \mathbf{x}_{j0} \rangle + \nu_j \langle \mathbf{y}_j, \zeta_j(t) \rangle + \mu_j \int_0^t |\langle \mathbf{y}_j, \xi_j(s) \rangle| ds \right\}$$

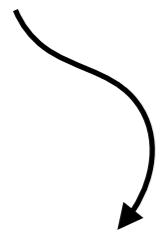
where

$$\mu_j := \frac{\beta_j - \alpha_j}{2}, \quad \nu_j := \frac{\beta_j + \alpha_j}{2}, \quad \xi_{j,k}(s) = \frac{s^{r_j-k}}{(r_j - k)!}, \quad \zeta_j(t_0, t) := \int_{t_0}^t \xi_j(s) ds \in \mathbb{R}^{r_j}$$

# Parametric formula for reach set boundary

**Theorem.** Assume  $\mathcal{X}_0 \equiv \{x_0\}$ . Then

$$x_{j,k}^{\text{bdy}}(\boldsymbol{\sigma}) = \sum_{\ell=1}^{r_j} \mathbf{1}_{k \leq \ell} \frac{t^{\ell-k}}{(\ell-k)!} \mathbf{x}_{j0,\ell} + \frac{\nu_j t^{r_j-k+1}}{(r_j-k+1)!}$$



$$\pm \frac{\mu_j}{(r_j-k+1)!} \left\{ (-1)^{r_j-1} t^{r_j-k+1} + 2 \sum_{q=1}^{r_j-1} (-1)^{q+1} \sigma_q^{r_j-k+1} \right\},$$

**kth Component of the boundary of jth input**

**Parameters:**

$$\mathcal{W}_{jt} := \{ \boldsymbol{\sigma} \in \mathbb{R}^{n-1} \mid 0 \leq \sigma_1 \leq \sigma_2 \leq \dots \leq \sigma_{r_j-1} \leq t \}, \quad j = 1, \dots, m$$

,

**Each single input integrator reach set has two bounding surfaces:**

$$\mathcal{X}_j (\{x_0\}, t) = \{ \mathbf{x} \in \mathbb{R}^{r_j} \mid p_j^{\text{upper}}(\mathbf{x}) \leq 0, p_j^{\text{lower}}(\mathbf{x}) \leq 0 \}$$

# Implicit formula for reach set boundary

Generating function of the parametric form:

$$F(\tau) = \sum_{k \geq 0} A_k \tau^k = \frac{(1 - \sigma_1 \tau)(1 - \sigma_3 \tau) \cdots}{(1 - \sigma_2 \tau)(1 - \sigma_4 \tau) \cdots} \quad (1)$$

Taking the logarithmic derivative for  $q = 1, \dots, d - 1$

$$\frac{F'(\tau)}{F(\tau)} = -\sigma_1 \sum_{k \geq 0} (\sigma_1 \tau)^k + \sigma_2 \sum_{k \geq 0} (\sigma_2 \tau)^k - \sigma_3 \sum_{k \geq 0} (\sigma_3 \tau)^k + \cdots$$

Integrating with respect to  $\tau$ :

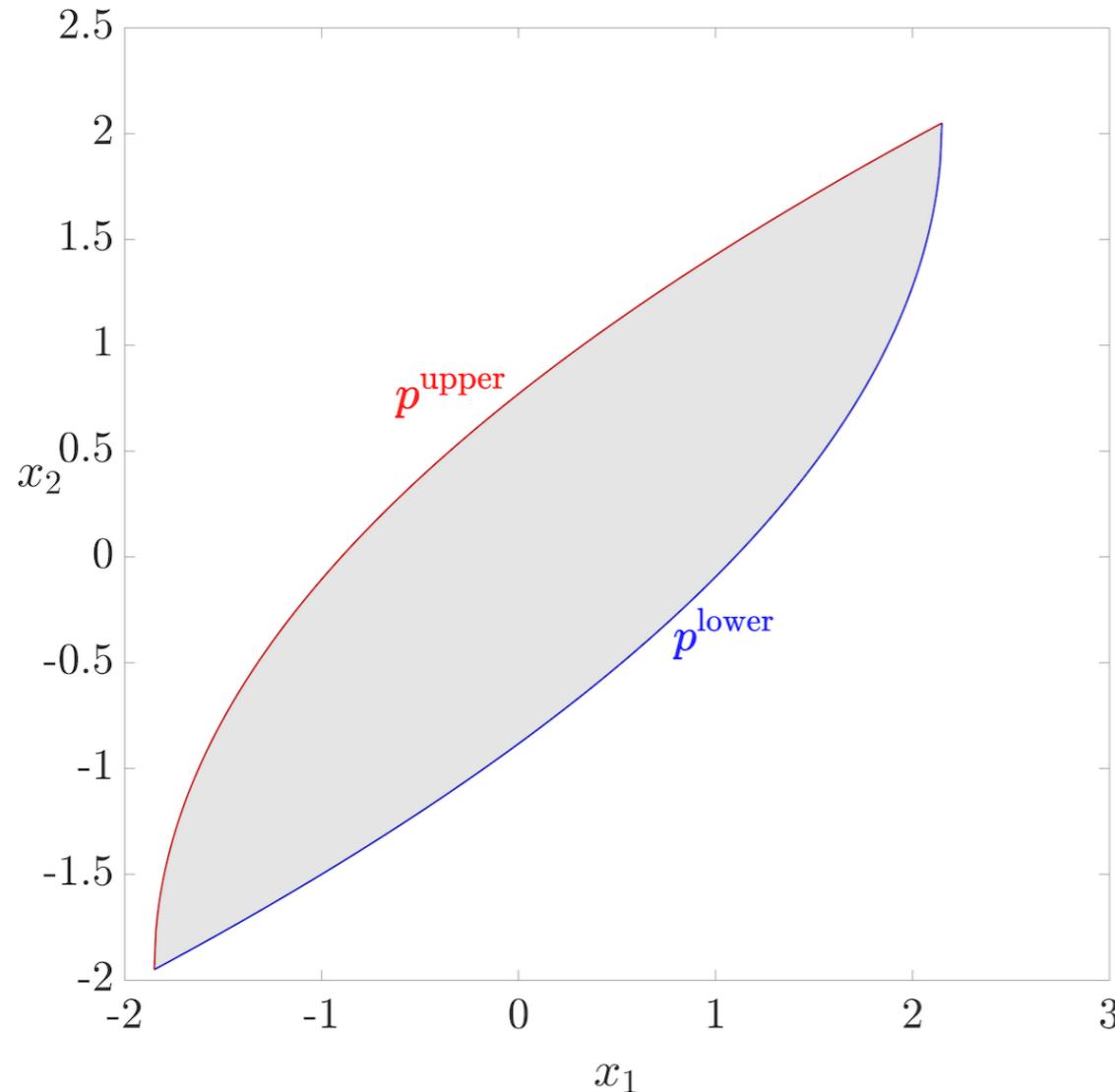
$$F(\tau) = \exp \left( - \sum_{k=1}^{n_x} \frac{\lambda_k}{k} \tau^k \right) \quad (2)$$

Equating (1) and (2), the following Hankel determinant gives implicit formula

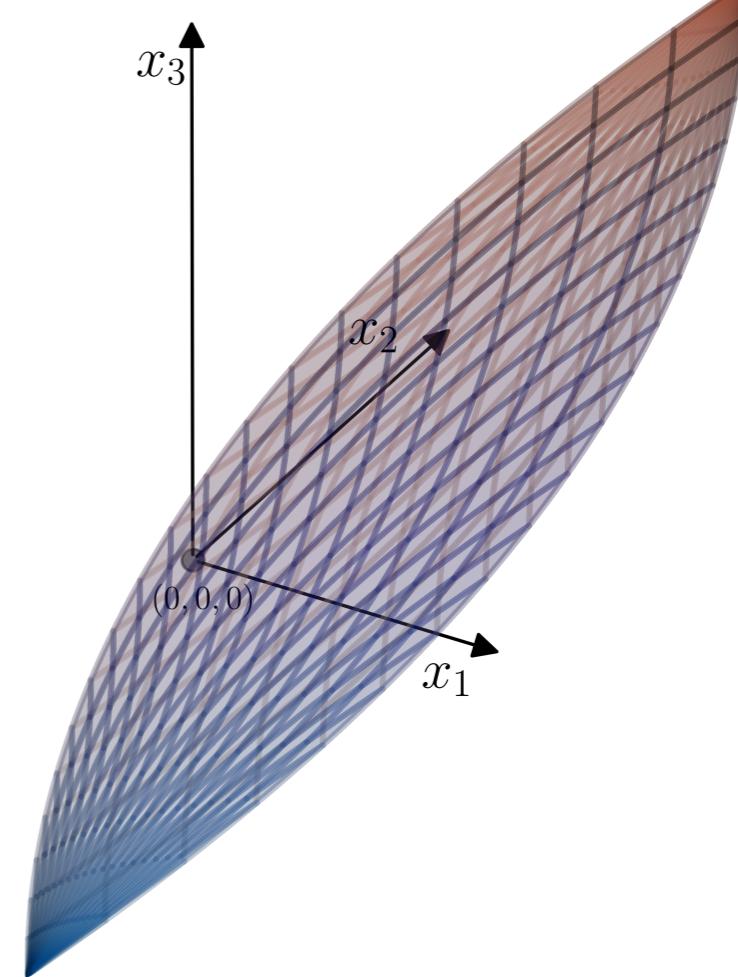
$$\det[A_d]_{i,j=0}^{\delta} = 0$$

# Taxonomy

**Theorem.** The set  $\mathcal{X}_t$  with  $\mathcal{X}_0 \equiv \{x_0\}$  is semialgebraic



The single input double integrator reach set



The single input triple integrator reach set

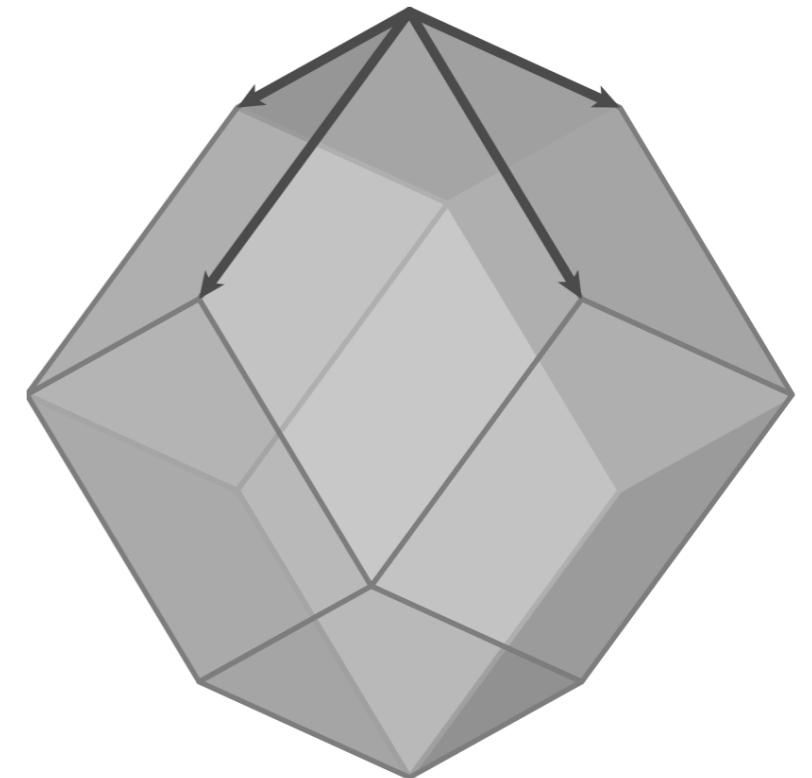
# Taxonomy

Zonotope of dimension d

$$\mathcal{Z}_n := \left\{ \sum_{j=1}^n \gamma_j \mathbf{v}_j \mid \gamma_j \in [-1, 1], \mathbf{v}_j \in \mathbb{R}^d, j = 1, \dots, n \right\}$$

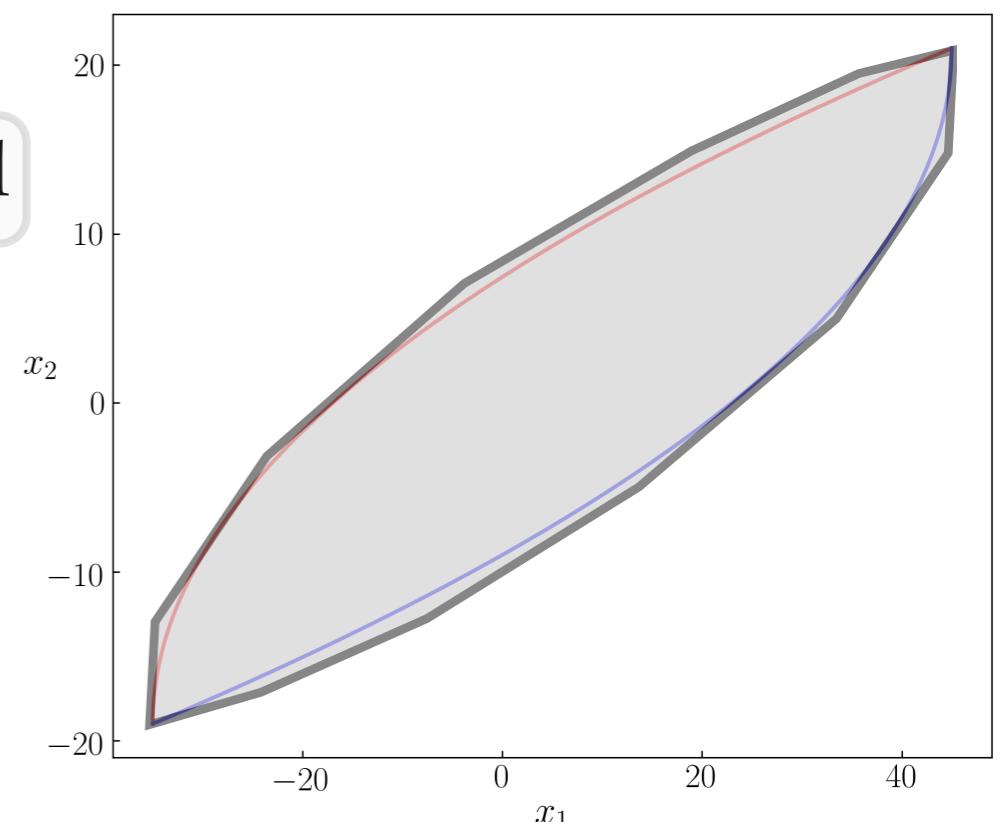
$$h_{\mathcal{Z}_n}(\mathbf{y}) = \sum_{j=1}^n |\langle \mathbf{y}, \mathbf{v}_j \rangle|, \quad \mathbf{y} \in \mathbb{R}^d$$

Generators



Zonoid: Limiting set of the Minkowski sum of line segments

**Theorem.** The set  $\mathcal{X}_t$  with  $\mathcal{X}_0 \equiv \{x_0\}$  is zonoid

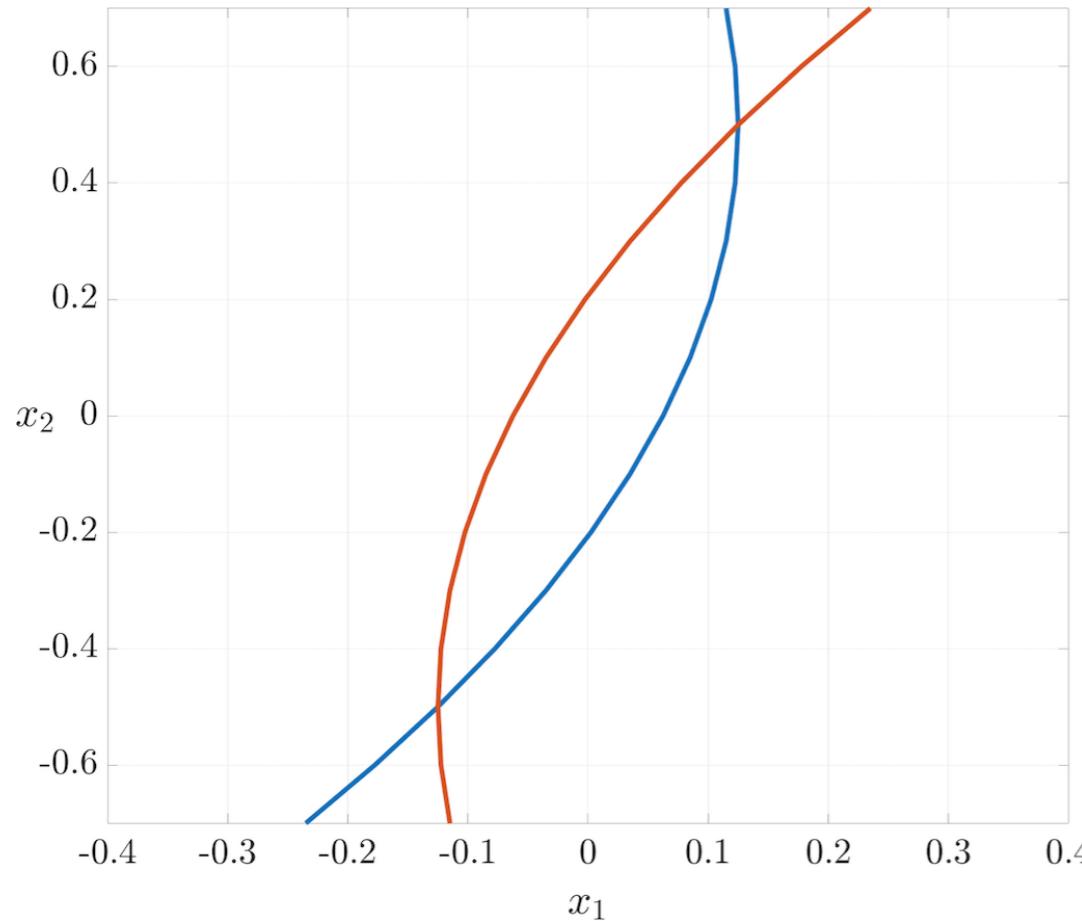


# Taxonomy

## Integrator Reach Set Is Not Spectrahedron

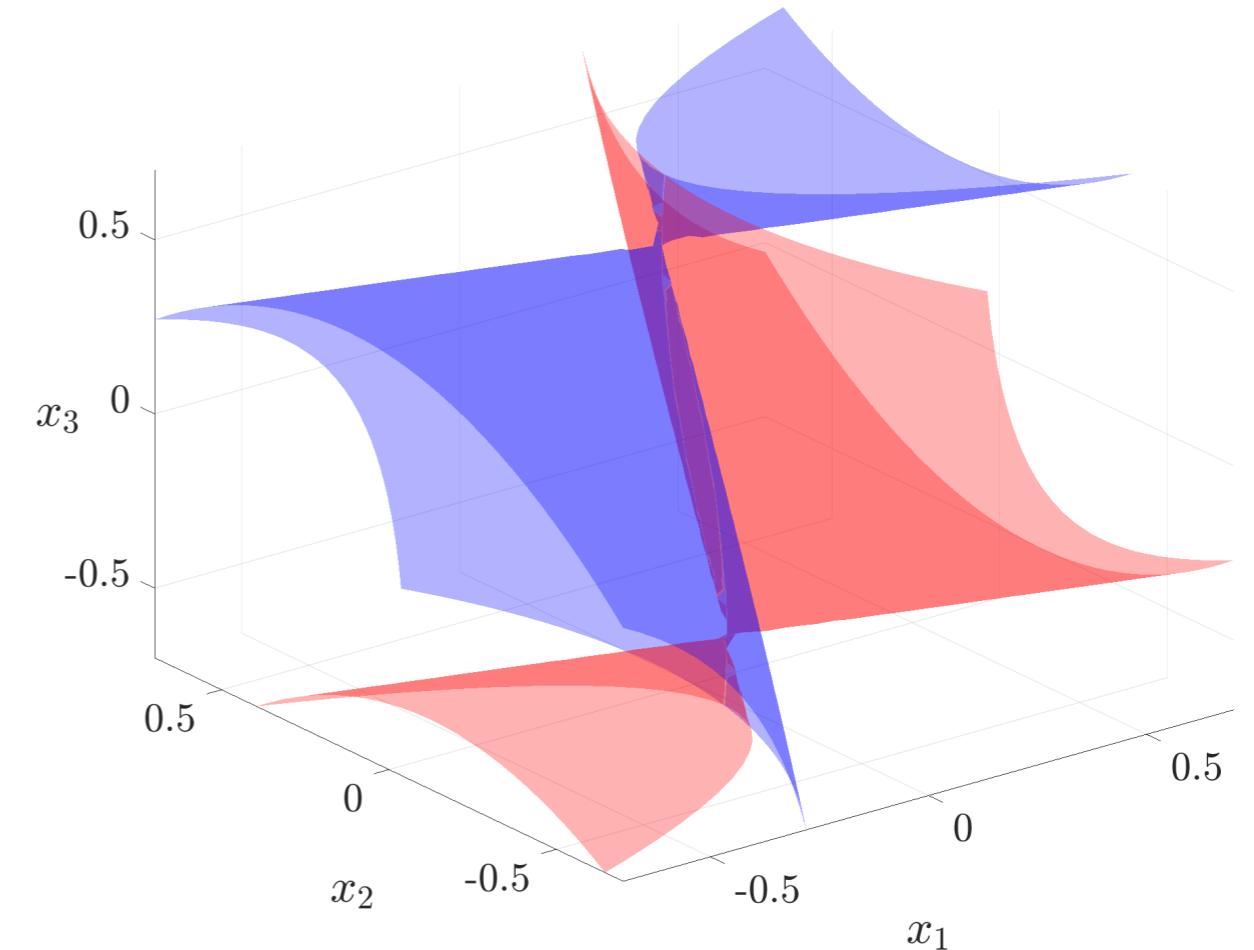
Polynomial degree of d dimensional integrator reach set surface:

$$\left(\left\lfloor \frac{d-1}{2} \right\rfloor + 1\right) \left(d - \left\lfloor \frac{d-1}{2} \right\rfloor\right)$$



Degree of  $\partial \mathcal{X}_t$  is 2

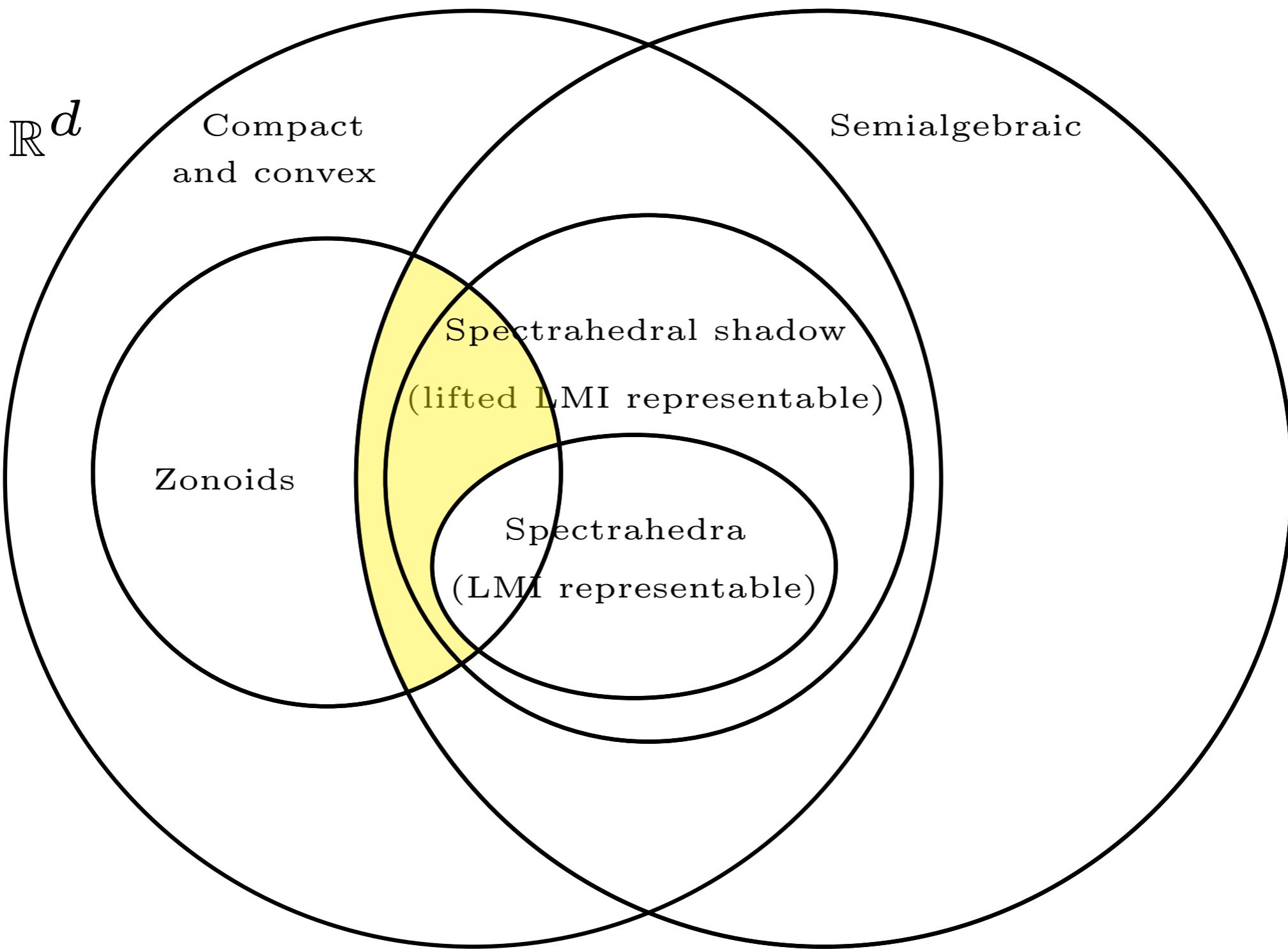
Number of intersections by generic line is 4



Degree of  $\partial \mathcal{X}_t$  is 4

Number of intersections by generic line is 6

# Taxonomy

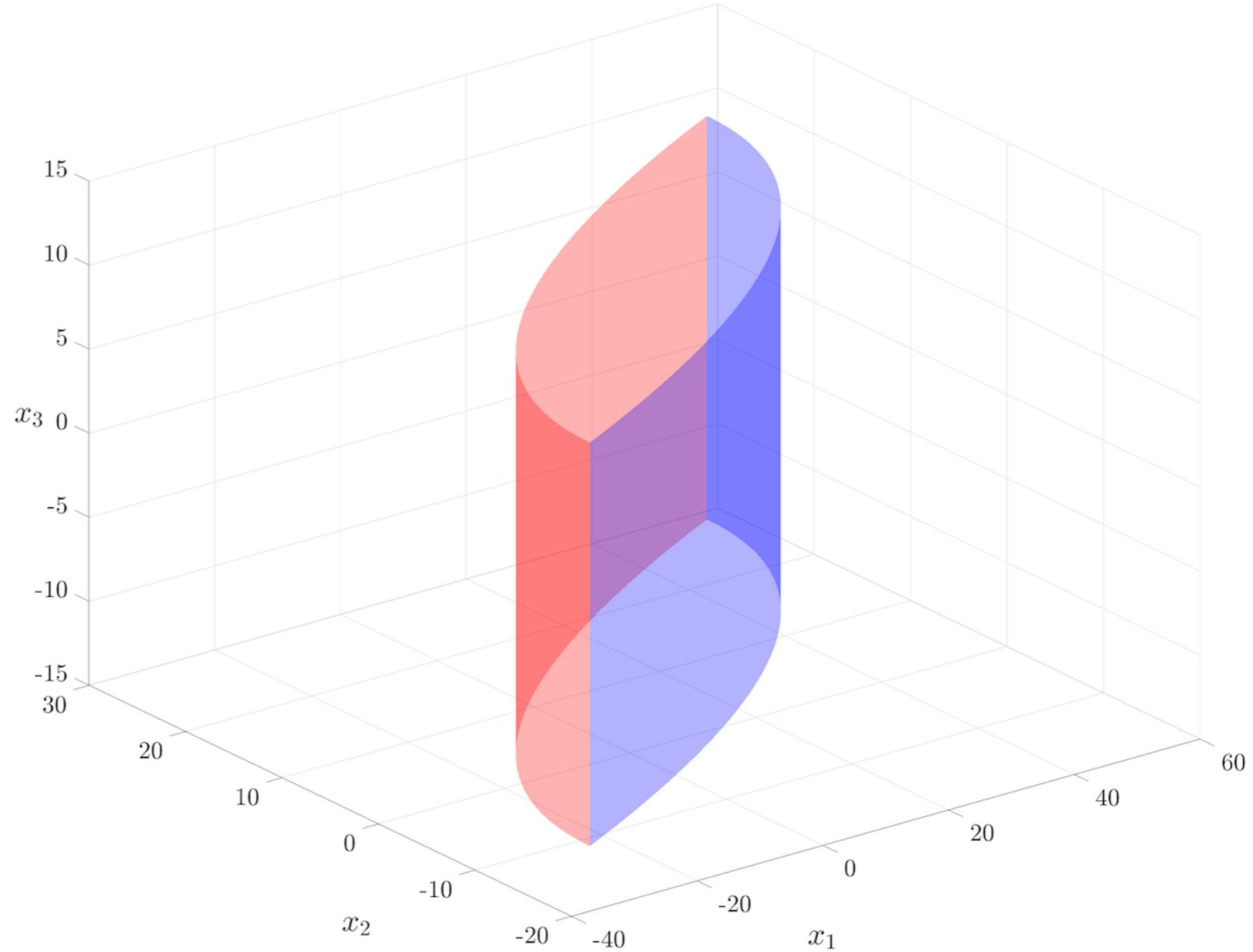


# Volume

**Theorem.**

$$\text{vol}(\mathcal{X}(\{x_0\}, t)) = 2^d \prod_{j=1}^m \left\{ \mu_j^{r_j} t^{r_j(r_j+1)/2} \prod_{k=1}^{r_j-1} \frac{k!}{(2k+1)!} \right\}.$$

$$\text{vol}(\mathcal{X}) = \frac{4}{3} \mu_1^2 \mu_2 t^4$$

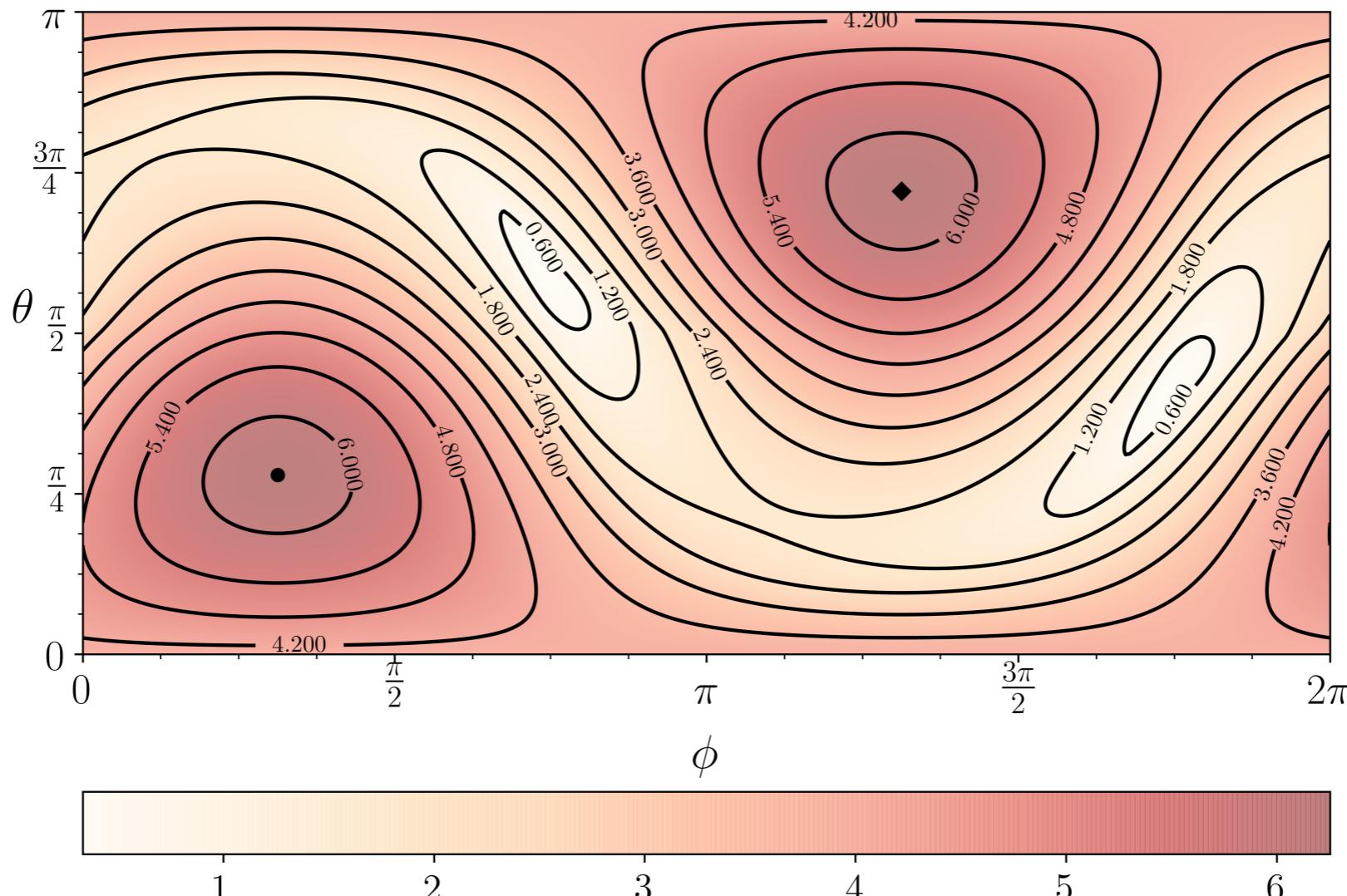


The integrator reach set at  $t = 4, \mathbf{r} = (2,1)^\top$

# Diameter

Theorem.

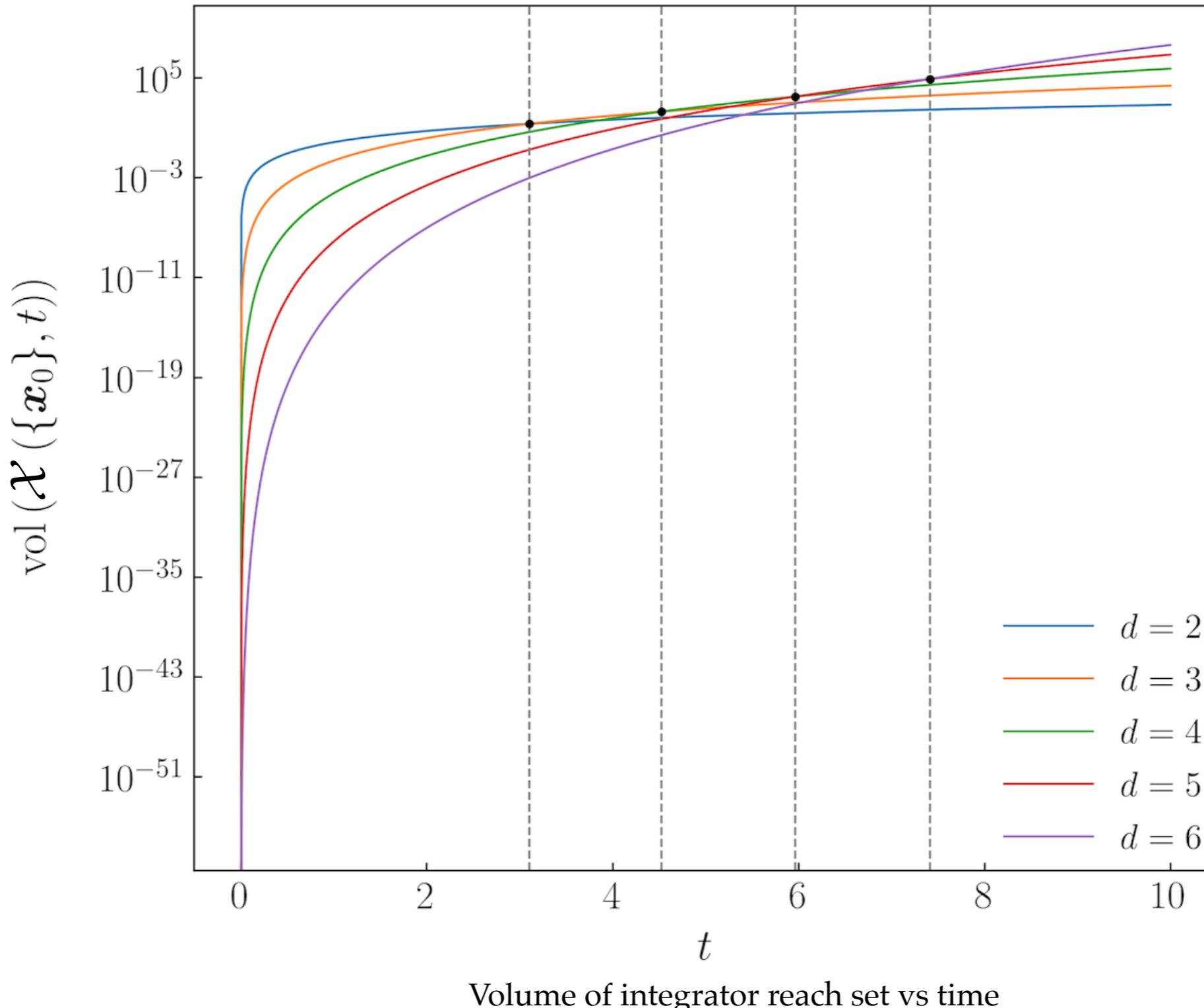
$$\text{diam} (\mathcal{X} (\{x_0\}, t)) = 2 \parallel \zeta(t) \parallel_2 = 2 \left( \sum_{j=1}^m \mu_j^2 \|\zeta_j\|^2 \right)^{1/2}$$



- $(\arctan(3/t), \arccos(6/\sqrt{t^4 + 9t^2 + 36}))$
- ◆  $(\pi + \arctan(3/t), \arccos(-6/\sqrt{t^4 + 9t^2 + 36}))$

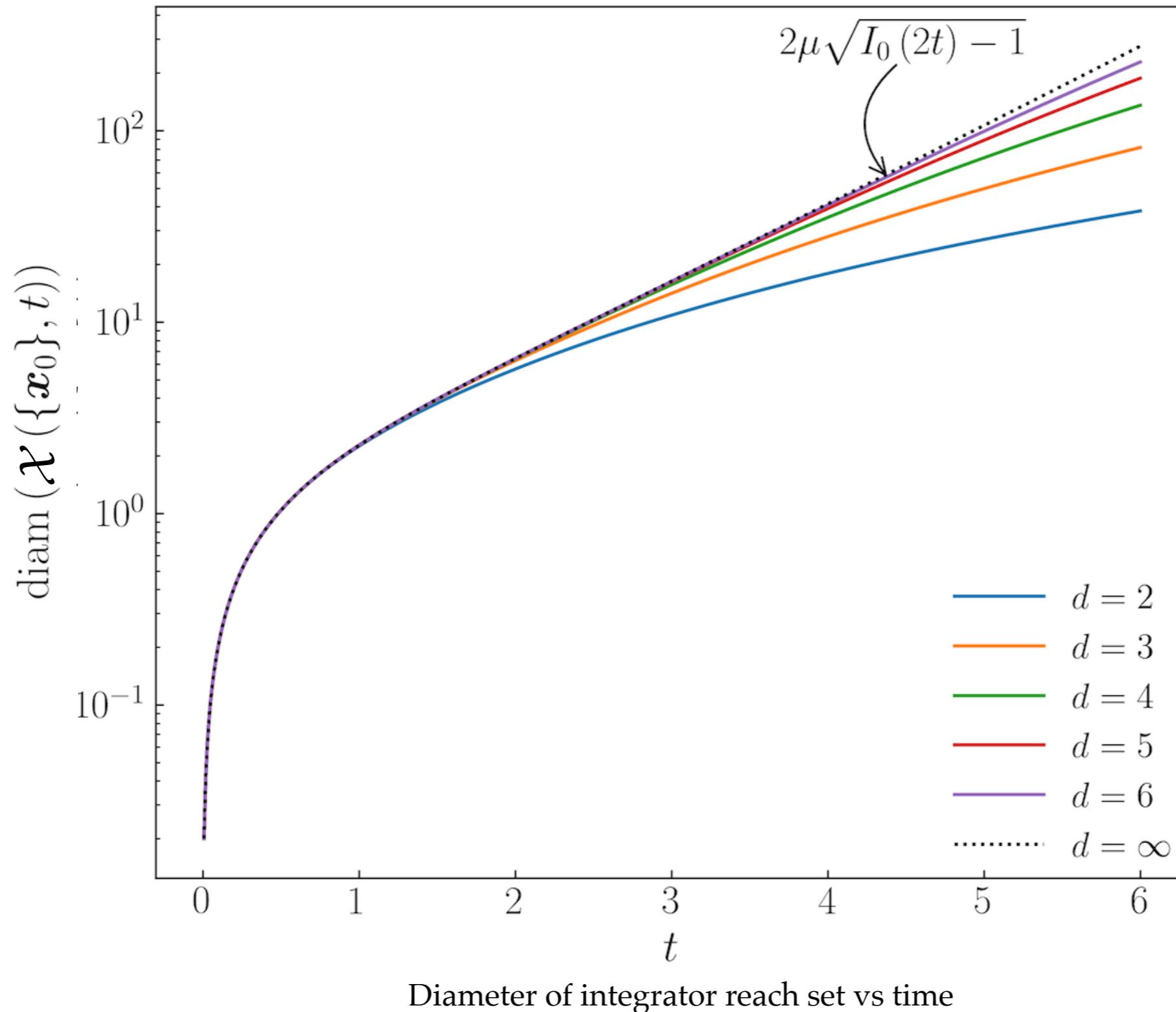
# Scaling Laws

$$\mathcal{X}_0 \equiv \{\mathbf{x}_0\}$$



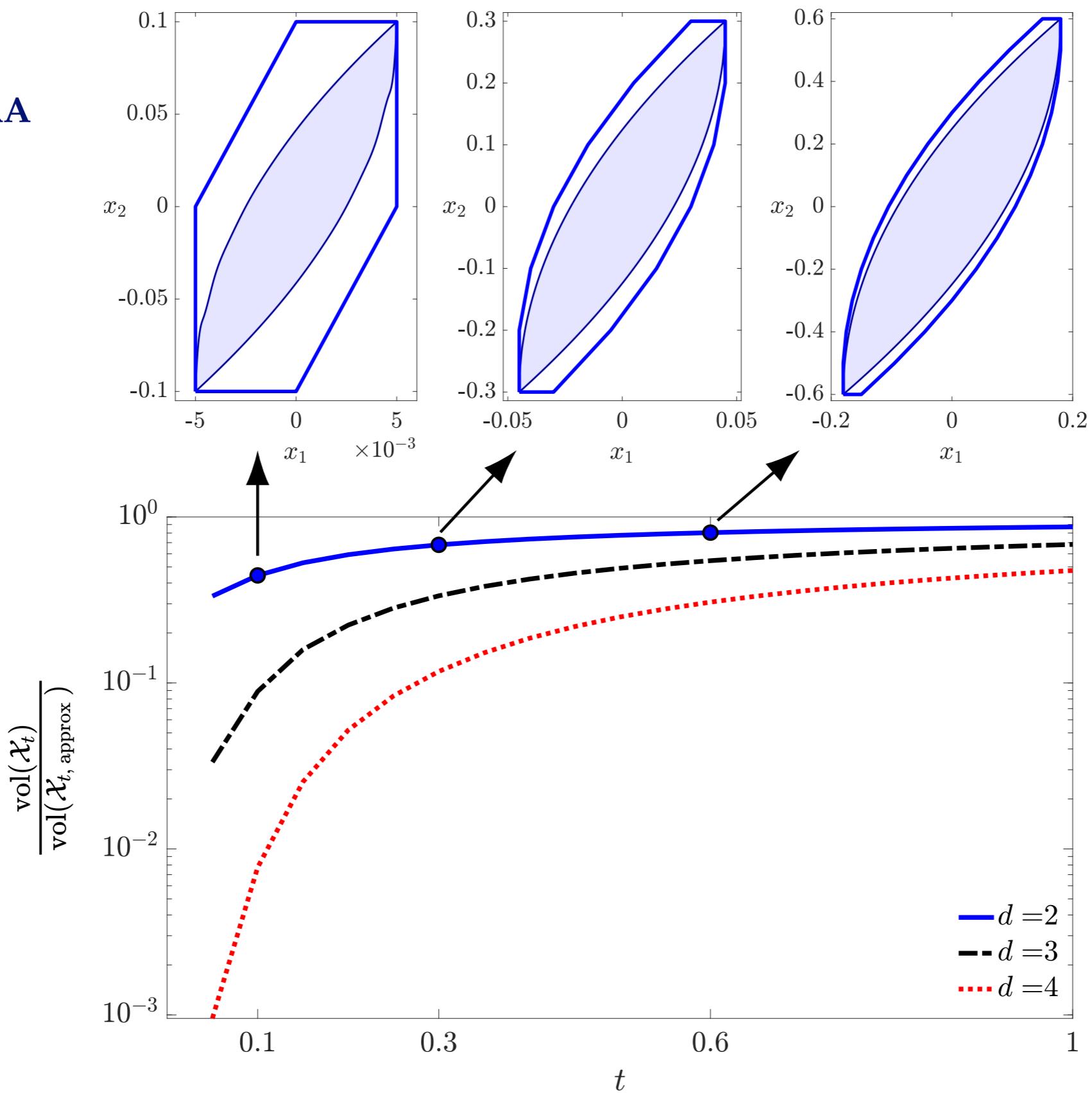
# Scaling Laws

$$\mathcal{X}_0 \equiv \{\mathbf{x}_0\}$$



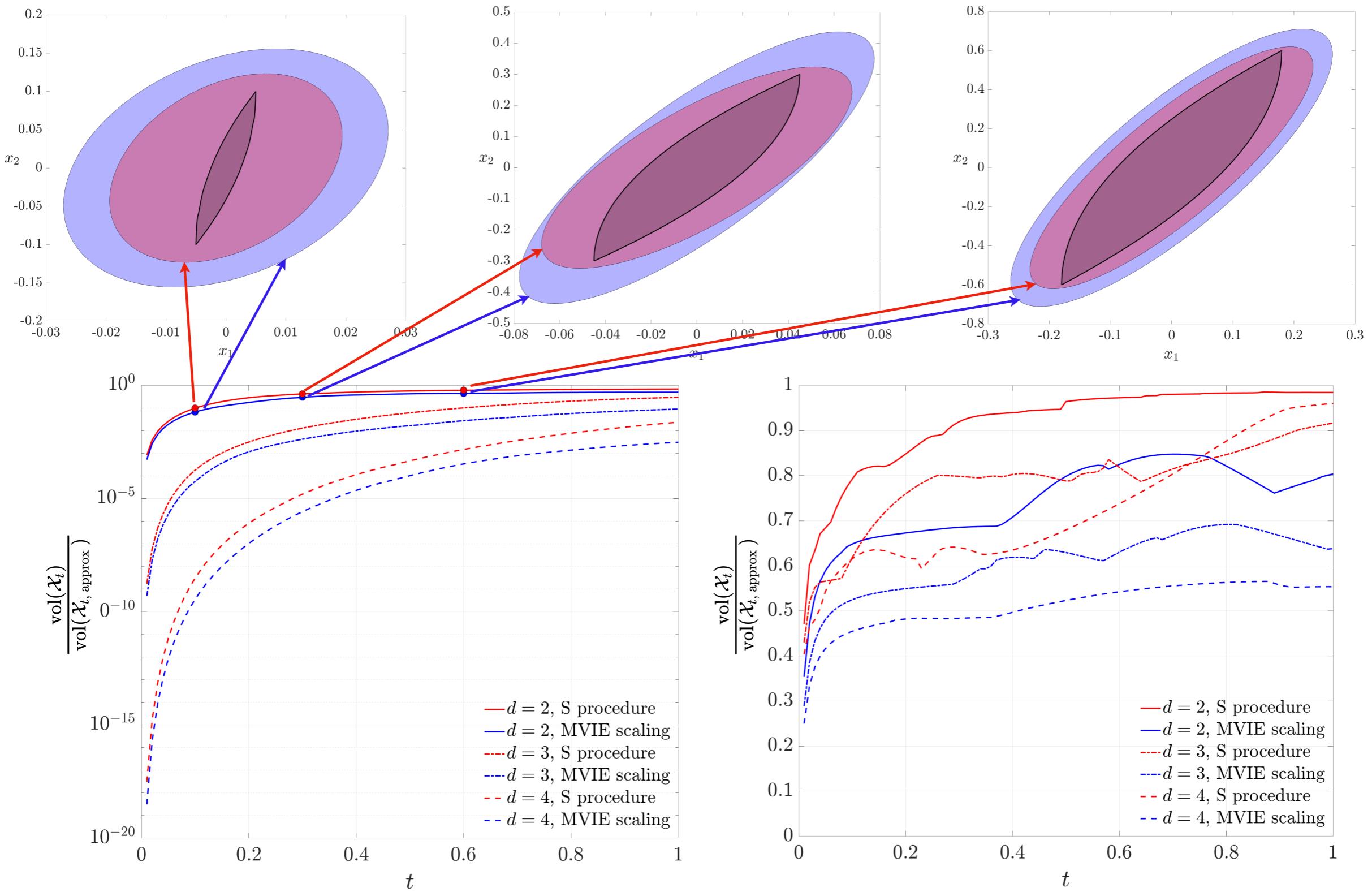
# Benchmarking over-approximations

From the CORA  
toolbox



# Benchmarking over-approximation

From the Ellipsoidal toolbox



# Overview

## Part 1: Analytical and Semi-analytical Computation of Reach Sets:

L-CSS 2024; TAC 2023; SVAA 2023; L-CSS 2022; ACC 2022, 2020

Integrators with time invariant input set

Integrators with time varying input set

Controllable LTI systems

Differentially flat nonlinear systems

## Part 2: Data-Driven Learning of Compact Sets:

ACC 2023; TCST 2022; CPS IoT 2021; L-CSS 2020

Region of attraction

Maximal control invariant set

Reach sets of neural networks, nonlinear control systems

# Overview

## Part 1: Analytical and Semi-analytical Computation of Reach Sets:

L-CSS 2024; TAC 2023; SVAA 2023; L-CSS 2022; ACC 2022, 2020

**Integrators with time invariant input set**

**Integrators with time varying input set**

- Generalizing the support function formula
- Generalizing the parametric formula
- Taxonomy
- Size
- Intersection detection

# Support function

$$h_{\mathcal{X}(\mathcal{X}_0, t)} = \sum_{j=1}^m \left\{ h_{\mathcal{X}_{j0}} \left( \exp(tA_j^\top) \mathbf{y} \right) + \int_0^t [\nu_j(s) \langle \mathbf{y}_j, \boldsymbol{\xi}(s) \rangle + \mu_j(s) |\langle \mathbf{y}_j, \boldsymbol{\xi}_j(s) \rangle|] \right\} ds$$

## Parametric formula for reach set boundary

$$\begin{aligned} \mathbf{x}_j^{\text{bdy}}(\sigma_j) &= \boldsymbol{\chi}(t, \mathbf{x}_{j0}) + \int_0^t \nu_j(s) \boldsymbol{\xi}_j(t-s) ds \pm \int_0^{\sigma_1} \mu_j(s) \boldsymbol{\xi}_j(t-s) ds \\ &\quad \mp \int_{\sigma_1}^{\sigma_2} \mu_j(s) \boldsymbol{\xi}_j(t-s) ds \pm \cdots \pm (-1)^n \int_{\sigma_{n-1}}^t \mu_j(s) \boldsymbol{\xi}_j(t-s) ds. \end{aligned}$$

Two bounding hypersurfaces

Zonoid

Not in general semialgebraic

Depends only on the time  
varying extremum  
trajectories

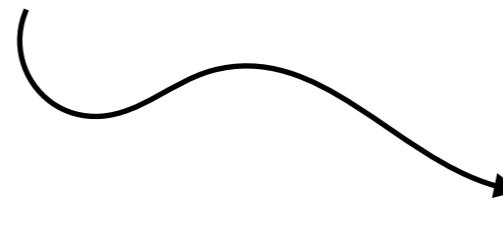
# Volume

## Integrator

$$\text{vol}(\mathcal{X}_t) = \int_{\mathcal{X}_t} d\mathbf{x} = \int_{\mathcal{W}_t} \int_{[0,1]^m} \left| \det \left( \frac{\partial \mathbf{x}}{\partial \boldsymbol{\sigma}} \frac{\partial \mathbf{x}}{\partial \lambda} \right) \right| d\boldsymbol{\sigma} d\lambda$$

$$\mathbf{x} = \boldsymbol{\pi}(\boldsymbol{\sigma}, \lambda) := \lambda \mathbf{x}^{\text{upper}}(\boldsymbol{\sigma}) + (1 - \lambda) \mathbf{x}^{\text{lower}}(\boldsymbol{\sigma})$$

$$(\boldsymbol{\sigma}, \lambda) \in \mathcal{W}_t \times [0, 1]$$



Antipodal property

## Static feedback linearizable systems

$$\text{vol}(\mathcal{Z}_t) = \int_{\mathcal{Z}_t} dz = \int_{\mathcal{W}_t} \int_{[0,1]^m} \left| \det \left( \frac{\partial z}{\partial \mathbf{x}} \right) \det \left( \frac{\partial \mathbf{x}}{\partial \boldsymbol{\sigma}} \frac{\partial \mathbf{x}}{\partial \lambda} \right) \right| d\boldsymbol{\sigma} d\lambda$$

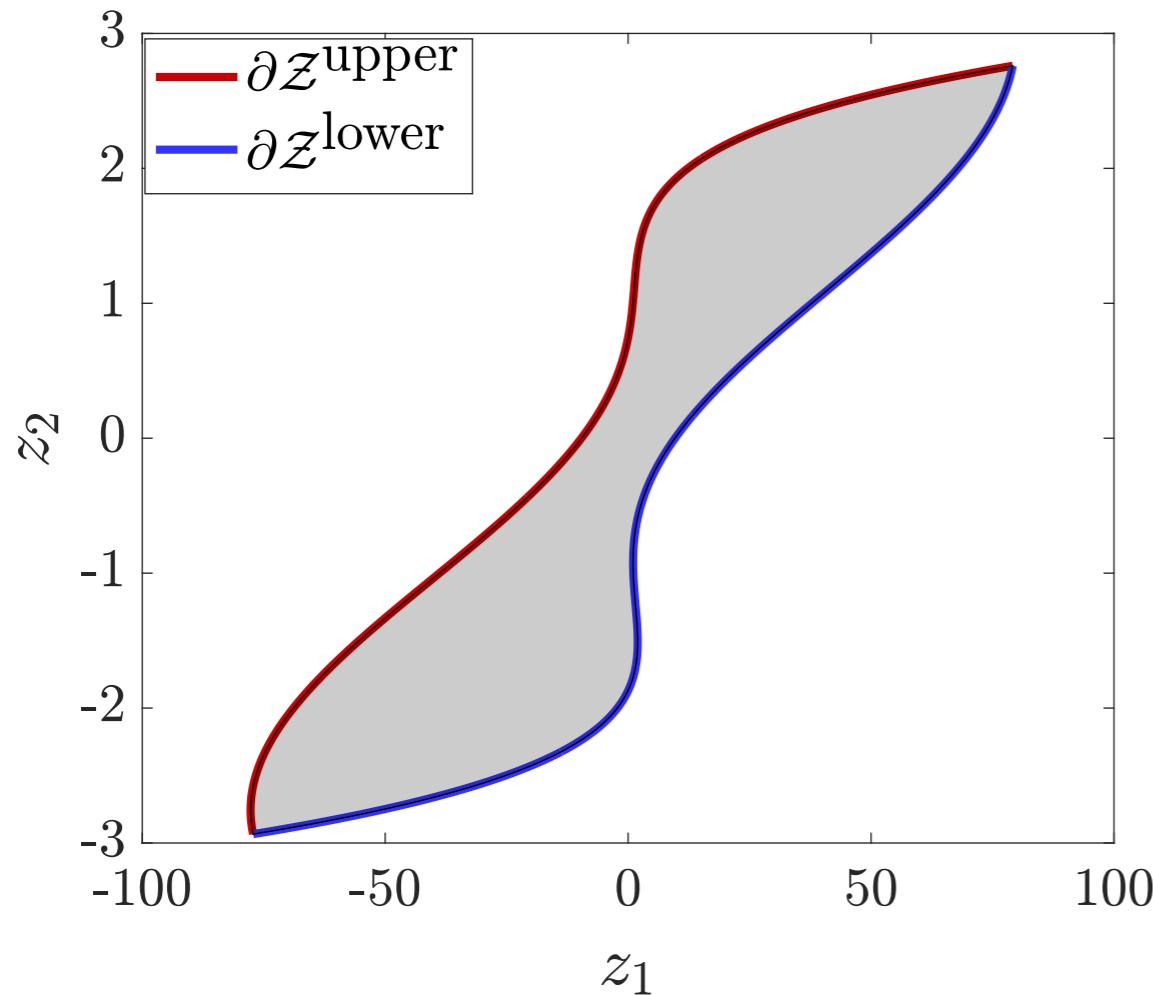


Diffeomorphsim

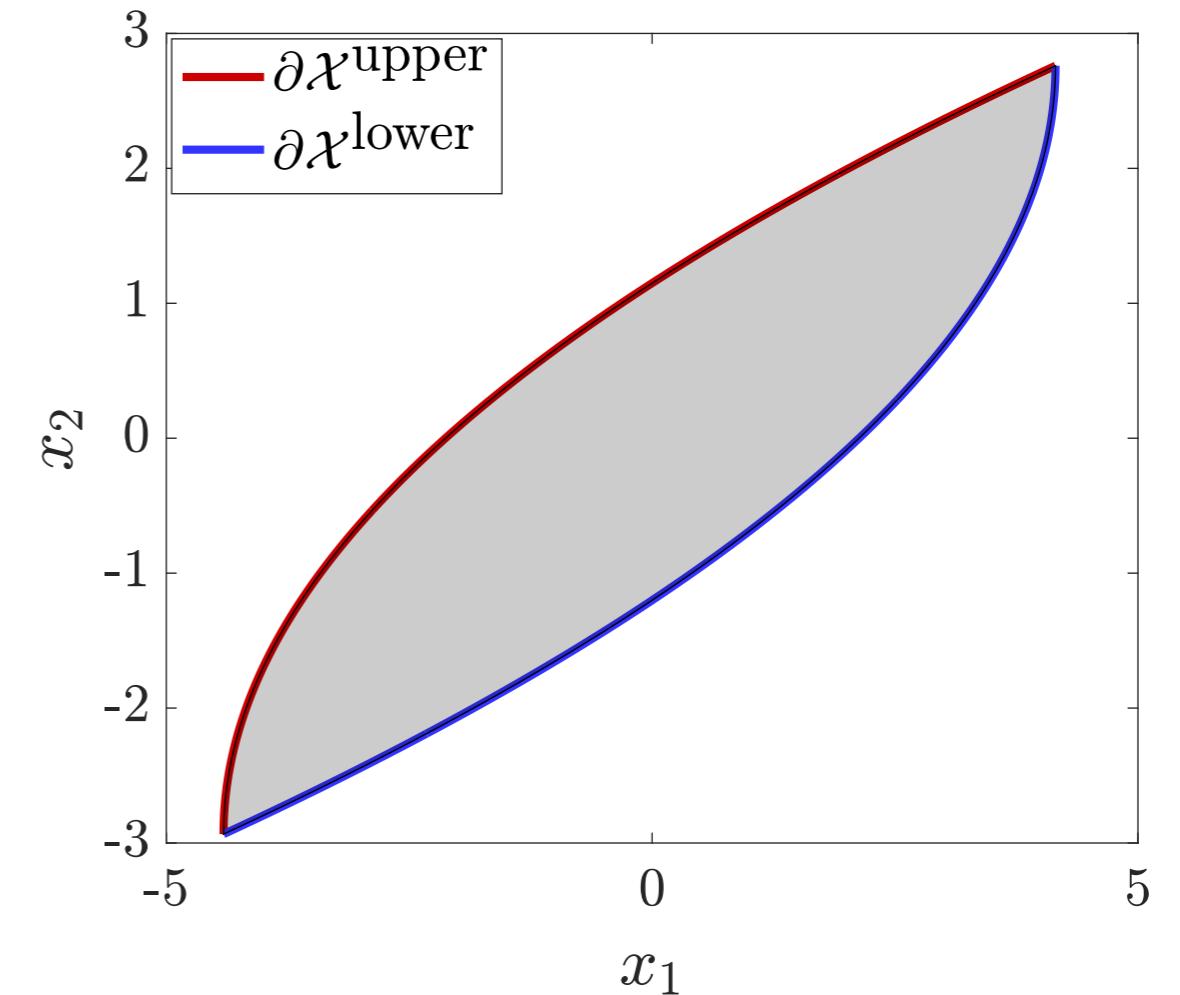
# Volume

$$\text{vol}(\mathcal{Z}_t) = \int_{\mathcal{Z}_t} d\mathbf{z} = \int_{\mathcal{W}_t} \int_{[0,1]^m} \left| \det \left( \frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right) \det \left( \frac{\partial \mathbf{x}}{\partial \boldsymbol{\sigma}} \frac{\partial \boldsymbol{\sigma}}{\partial \lambda} \right) \right| d\boldsymbol{\sigma} d\lambda$$

Diffeomorphsim



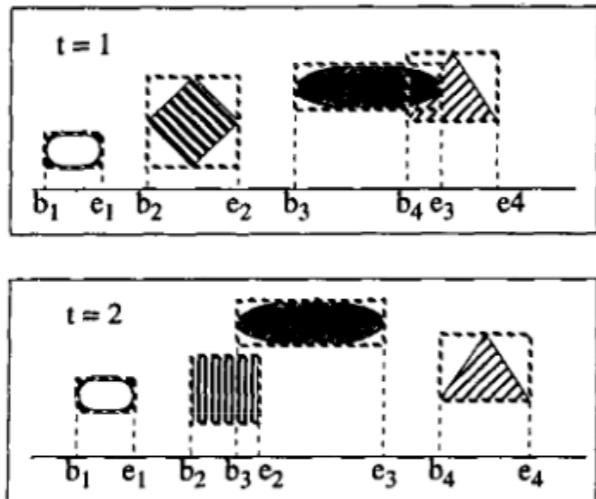
$$\text{vol}(\mathcal{Z}_t) = 206.7362$$



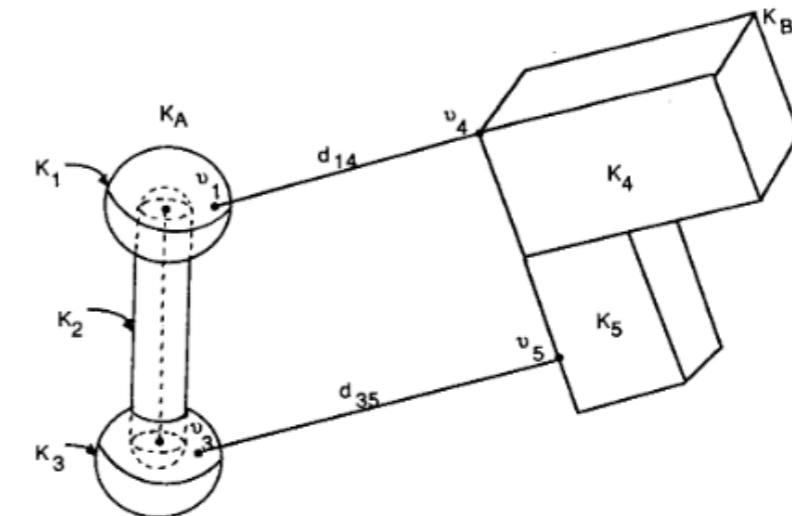
$$\text{vol}(\mathcal{X}_t) = 15.4292$$

# Certifying set intersection (or the lack of it) for static feedback linearizable system

# Existing Algorithms



[Cohen et. al, 1995]



[Elmer et. al, 1988]

Main result: detecting intersection between integrators

Intersection in integrator coordinate  $x$



Intersection in original coordinates  $z$

Reach set of a static feedback linearizable system:  $\mathcal{Z}_t \subset \mathbb{R}^d$

Reach set of the corresponding integrator system:  $\mathcal{X}_t \subset \mathbb{R}^d$

# Intersection Detection

General formula for the collision detection

$$\text{dist}(A, B) := \min_{\mathbf{x}^A \in \mathcal{X}_t^A, \mathbf{x}^B \in \mathcal{X}_t^B} \|\mathbf{x}^A - \mathbf{x}^B\|_2^2$$

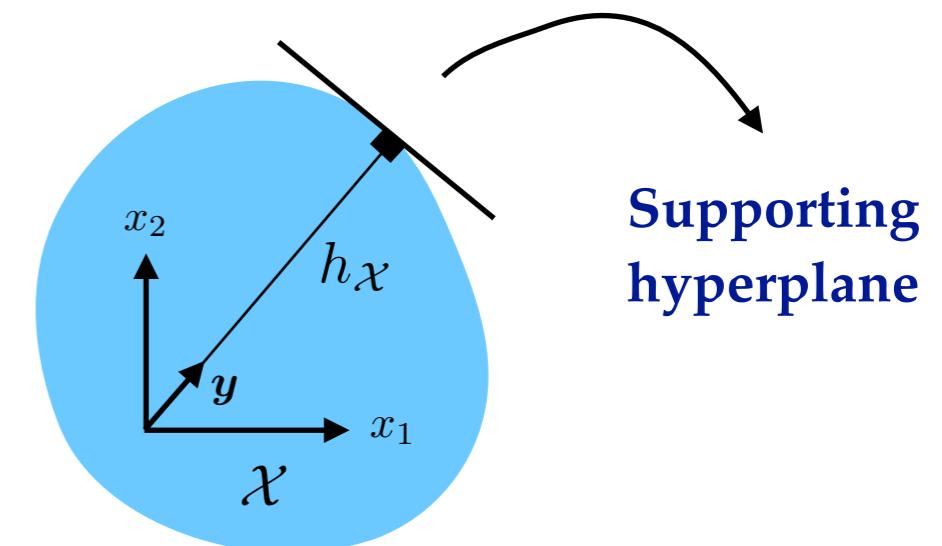
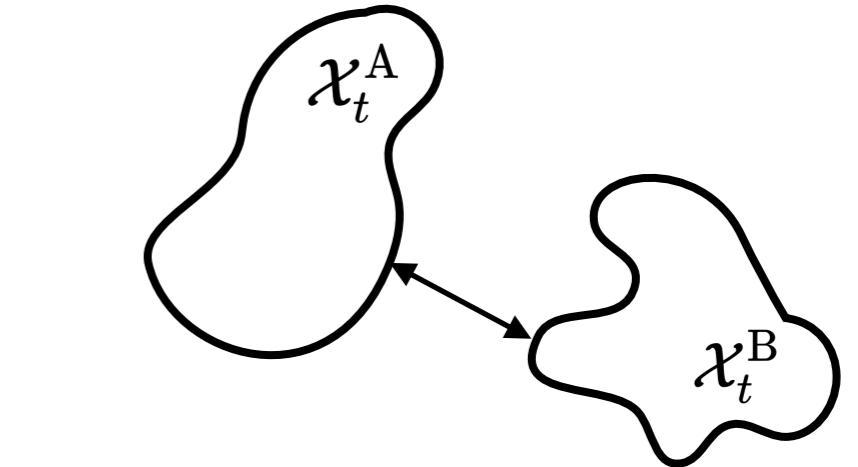
Lack of analytical handle on the boundary!!!

Intersection Detection Oracle

$$\min_{\mathbf{y} \in \mathbb{S}^{n-1}} \{h_{\mathcal{X}_t^A}(\mathbf{y}) + h_{\mathcal{X}_t^B}(-\mathbf{y})\} \geq (<) 0 \iff \mathcal{X}_t^A \cap \mathcal{X}_t^B \neq (=) \emptyset$$

Reminder:

$$h_{\mathcal{X}(\mathcal{X}_0, t)}(\mathbf{y}) := \sup_{\mathbf{x} \in \mathcal{X}} \left\{ \langle \mathbf{x}, \mathbf{y} \rangle \mid \mathbf{y} \in \mathbb{S}^{d-1} \right\}$$



# Intersection Detection

## Intersection Detection Oracle

$$\min_{\mathbf{y} \in \mathbb{S}^{n-1}} \{h_{\mathcal{X}_t^A}(\mathbf{y}) + h_{\mathcal{X}_t^B}(-\mathbf{y})\} \geq (<) 0 \iff \mathcal{X}_t^A \cap \mathcal{X}_t^B \neq (=) \emptyset$$

**Theorem:** The support function of  $\mathcal{X}_t$

$$h_{\mathcal{X}_t}(\mathbf{y}) = \sum_{j=1}^m \langle \mathbf{y}_j, \exp(t\mathbf{A})\mathbf{x}_{j0} \rangle + \int_0^t \ell(s) \left\| (\exp(s\mathbf{A})\mathbf{B})^\top \mathbf{y} \right\|_q ds, \quad 1/p + 1/q = 1$$

Discretize  $[0, t]$  into  $K$  intervals:

$$\int_0^t \left\| (\exp(s\mathbf{A})\mathbf{B})^\top \mathbf{y} \right\|_q ds \approx \cdot$$

$$\frac{\Delta s}{2} \sum_{k=1}^K \left( \left\| (\exp(s_{k-1}\mathbf{A})\mathbf{B})^\top \mathbf{y} \right\|_q + \left\| (\exp(s_k\mathbf{A})\mathbf{B}(s_k))^\top \mathbf{y} \right\|_q \right)$$

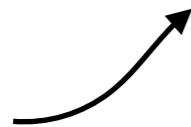
# Lossless Convexification

## Nonconvex Optimization Problem

$$\overbrace{p^* = \min_{\boldsymbol{\eta} \in \mathbb{R}^{n+K+1}} \langle \boldsymbol{\kappa}(t), \boldsymbol{\eta} \rangle}^{\text{Nonconvex Objective}}$$

s.  $\|M_k \boldsymbol{\eta}\|_q - \langle \mathbf{e}_{n+k}^{n+K+1}, \boldsymbol{\eta} \rangle \leq 0,$

$-\tilde{\mathbf{N}}\boldsymbol{\eta} \leq \mathbf{0}, \quad \|\mathbf{N}\boldsymbol{\eta}\|_2 \leq 1$



Nonconvex Constraint

→  
Lossless  
Convexification

## Second-Order Cone Program

$$\overbrace{\tilde{p}^* = \min_{\boldsymbol{\eta} \in \mathbb{R}^{n+K+1}} \langle \boldsymbol{\kappa}(t), \boldsymbol{\eta} \rangle}^{\text{Convex Objective}}$$

s.  $\|M_k \boldsymbol{\eta}\|_q - \langle \mathbf{e}_{n+k}^{n+K+1}, \boldsymbol{\eta} \rangle \leq 0,$

$-\tilde{\mathbf{N}}\boldsymbol{\eta} \leq \mathbf{0}, \quad \|\mathbf{N}\boldsymbol{\eta}\|_2 \leq 1$

(i)  $\tilde{p}^* \leq 0$

(ii)  $\tilde{p}^* = 0 \Rightarrow 0 \leq p^* \Leftrightarrow \mathcal{X}_t^A \cap \mathcal{X}_t^B \neq \emptyset$

(iii)  $\tilde{p}^* < 0 \Rightarrow \tilde{p}^* = p^* < 0 \Leftrightarrow \mathcal{X}_t^A \cap \mathcal{X}_t^B = \emptyset$

# Example

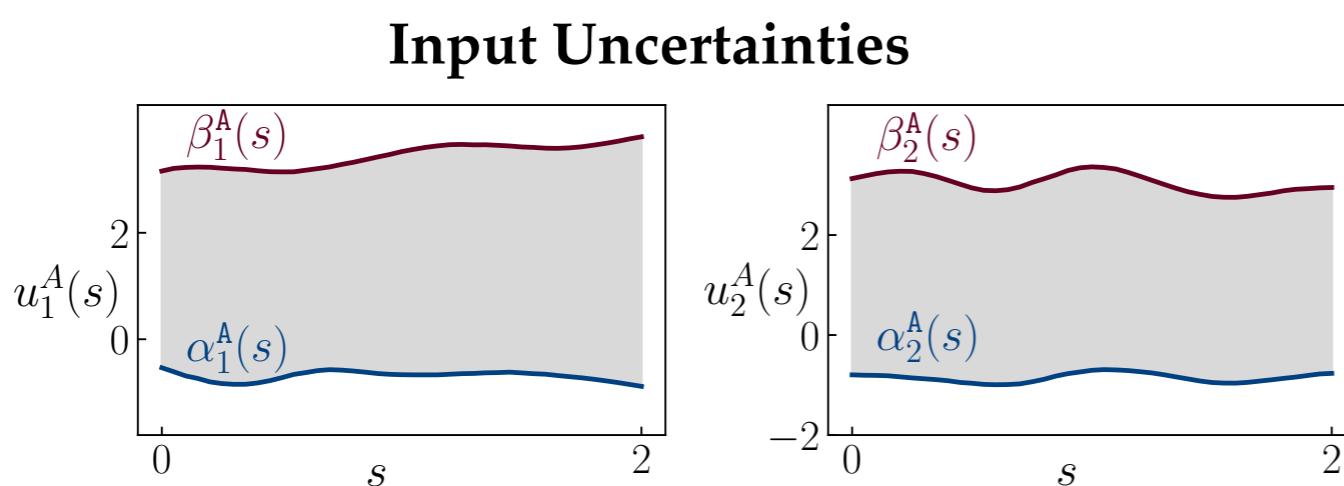
$$\dot{\mathbf{x}}^i = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \mathbf{x}^i + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \mathbf{u}^i, \quad i \in \{\text{A, B}\}, \quad r = (3, 2)^\top$$



**$\infty$ -norm ball**

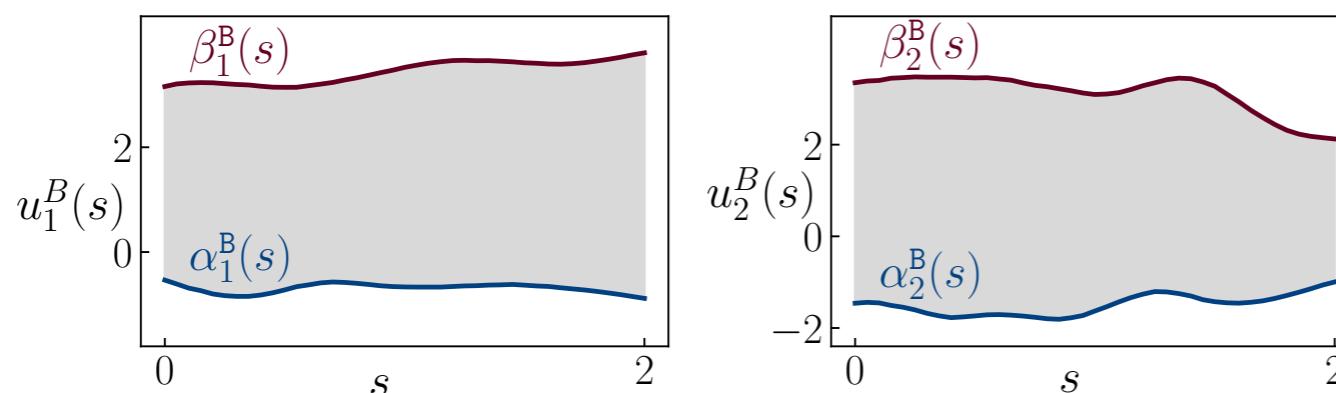
$$\mathcal{U}(s) := [\alpha_1(s), \beta_1(s)] \times [\alpha_2(s), \beta_2(s)] \quad \text{for all } 0 \leq s \leq t$$

**Decoupled dynamics for each block**

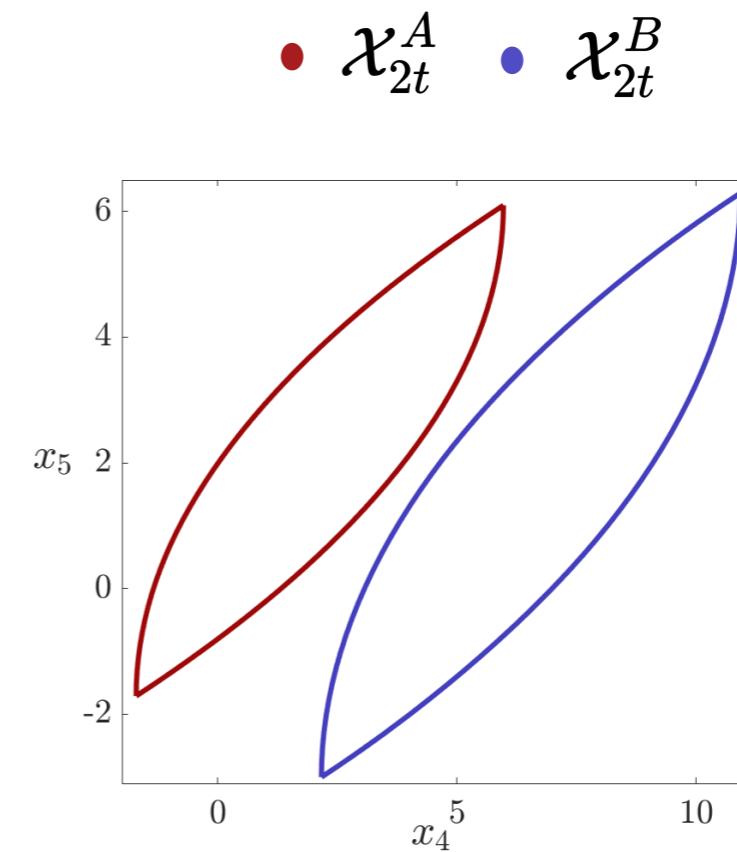
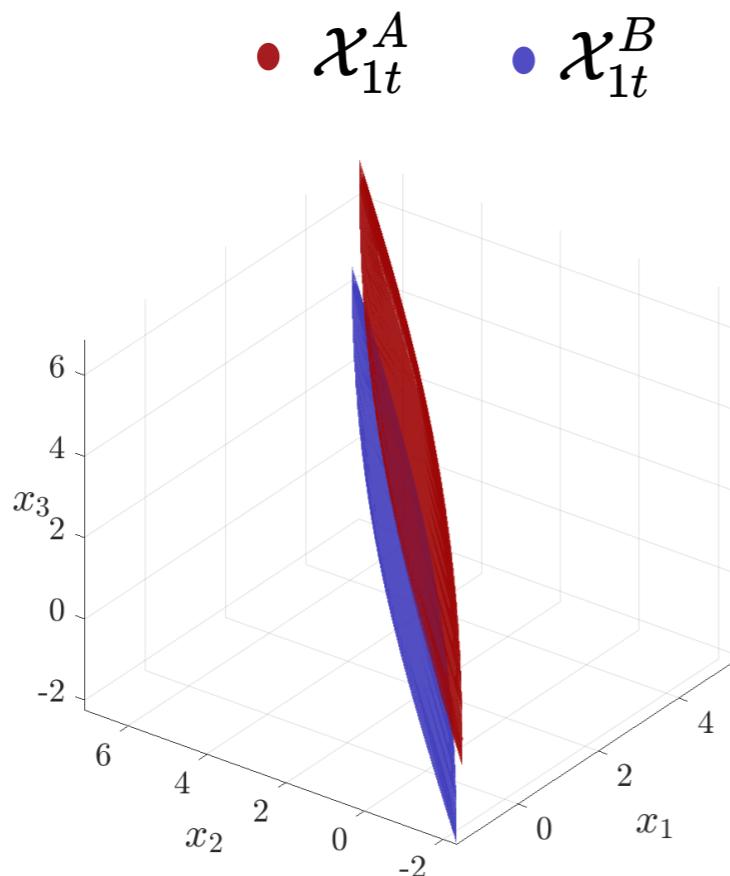


$$\mathcal{X}_t^i = \mathcal{X}_{1t}^i \times \mathcal{X}_{2t}^i = \mathcal{X}_{1t}^i + \mathcal{X}_{2t}^i$$

$$h_{\mathcal{X}}^i = h_{\mathcal{X}_{1t}}^i + h_{\mathcal{X}_{2t}}^i \quad i \in \{\text{A, B}\}$$



# Example



$$(\tilde{p}_1^*, \tilde{p}_2^*) = (0, -0.54)$$

Computation time  $\approx 1.24$  s

$$\mathcal{X}_{1t}^A \cap \mathcal{X}_{1t}^B \neq \emptyset \quad \text{and} \quad \mathcal{X}_{2t}^A \cap \mathcal{X}_{2t}^B = \emptyset \iff \mathcal{X}_t^A \cap \mathcal{X}_t^B = \emptyset$$

# Overview

## Part 1: Analytical and Semi-analytical Computation of Reach Sets:

L-CSS 2024; TAC 2023; SVAA 2023; L-CSS 2022; ACC 2022, 2020

Integrators with time invariant input set

Integrators with time varying input set

Controllable LTI systems

Differentially flat nonlinear systems

## Part 2: Data-Driven Learning of Compact Sets:

ACC 2023; TCST 2022; CPS IoT 2021; L-CSS 2020

Region of attraction

Maximal control invariant set

Reach sets of neural networks, nonlinear control systems

# Reach set of controllable LTI systems

**Controllable LTI:**  $\dot{\mathbf{z}} = \tilde{\mathbf{A}}\mathbf{z} + \tilde{\mathbf{b}}v, \quad \tilde{\mathbf{A}} \in \mathbb{R}^{d \times d}, \quad \tilde{\mathbf{b}} \in \mathbb{R}^d,$

**Invertible map**

$$\mathbf{M} := \begin{pmatrix} \mathbf{q}^\top & \mathbf{q}^\top \tilde{\mathbf{A}} & \dots & \mathbf{q}^\top \tilde{\mathbf{A}}^{d-1} \end{pmatrix}^\top$$

last row of the inverse of  
the controllability matrix

**Controllable canonical form**  $\dot{\mathbf{x}} = \mathbf{A}_{\text{con}}\mathbf{x} + \mathbf{b}_{\text{con}}v$

$$\mathbf{A}_{\text{con}} := \mathbf{M}\tilde{\mathbf{A}}\mathbf{M}^{-1} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -c_0 & -c_1 & -c_2 & \dots & -c_{d-1} \end{pmatrix}$$

Coefficient of the  
characteristic polynomial

$$\mathbf{b}_{\text{con}} := \mathbf{M}\tilde{\mathbf{b}} = (0 \ 0 \ \dots \ 0 \ 1)^\top$$

# Semi-analytical algorithm

Define a new input:  $u := -\langle \mathbf{c}, \mathbf{x} \rangle + v \in \mathcal{U}(t)$  Integrator input set



Brunovsky normal form:  $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}_{\text{con}}u$

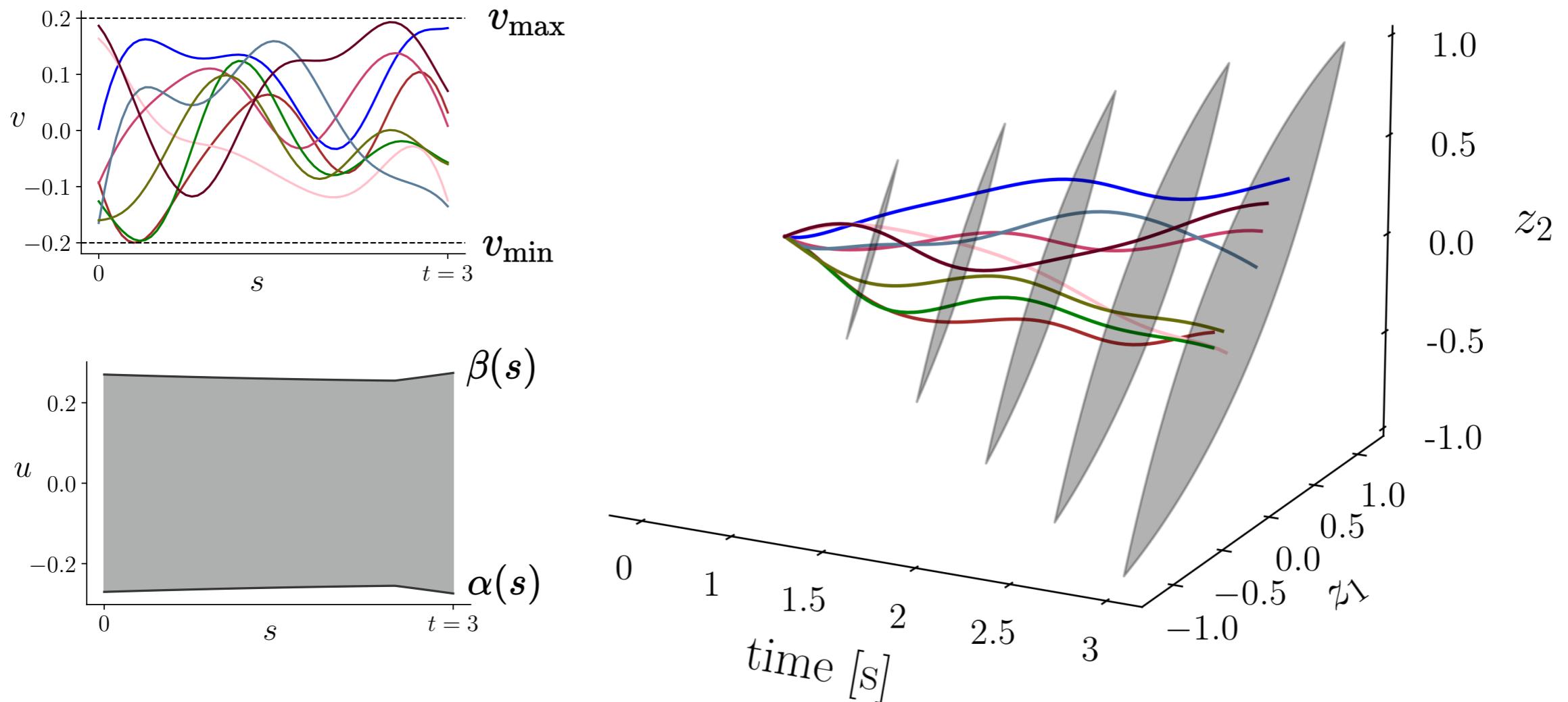
Main idea:  $\mathcal{Z}_t(\{\mathbf{z}_0\}) = \mathbf{M}^{-1}\mathcal{X}_t(\{\mathbf{M}\mathbf{z}_0\})$

We need to find:

$$\alpha(s) = -\langle \mathbf{c}, e^{s\mathbf{A}_{\text{con}}} \mathbf{M} \mathbf{z}_0 \rangle + \inf_{v(\cdot) \in C([0,s])} I(v), \quad \text{subject to} \quad v_{\min} \leq v(\cdot) \leq v_{\max}$$

$$\beta(s) = -\langle \mathbf{c}, e^{s\mathbf{A}_{\text{con}}} \mathbf{M} \mathbf{z}_0 \rangle + \sup_{v(\cdot) \in C([0,s])} I(v), \quad \text{subject to} \quad v_{\min} \leq v(\cdot) \leq v_{\max}$$

# Semi-analytical algorithm



**Example**

$$\begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \end{pmatrix} = \underbrace{\begin{pmatrix} 0.1 & 0.2 \\ -0.3 & 0.1 \end{pmatrix}}_{\tilde{A}}(z_1) + \underbrace{\begin{pmatrix} 1 \\ 2 \end{pmatrix}}_{\tilde{b}}v$$

$$\{v(\cdot) \in C([0, t]) \mid v(s) \in [-0.2, 0.2] \forall s \in [0, t]\}$$

# Hausdorff Distance

## Reach set of LTI systems

$$\begin{aligned} \mathcal{X}_t^i := & \bigcup_{\text{measurable } \mathbf{u}^i(\cdot) \in \mathcal{U}^i} \left\{ \mathbf{x}^i(t) \in \mathbb{R}^d \mid \dot{\mathbf{x}}^i(t) = \mathbf{A}(t)\mathbf{x}^i(t) + \mathbf{B}(t)\mathbf{u}^i(t), \quad i \in \{1, 2\}, \right. \\ & \left. \mathbf{x}^i(t=0) \in \text{compact convex } \mathcal{X}_0, \quad \mathbf{u}^i(\tau) \in \mathcal{U}^i \text{ for all } 0 \leq \tau \leq t \right\} \end{aligned}$$

## Quantifying the conservatism for $\ell_p$ norm bounded input set

$$\begin{aligned} \delta(\mathcal{X}_t^1, \mathcal{X}_t^2) &= \sup_{\mathbf{y} \in \mathbb{S}^{d-1}} |h_1(\mathbf{y}) - h_2(\mathbf{y})|, \\ &= \sup_{\|\mathbf{y}\|_2=1} \int_0^t \left( \left\| (\Phi(t, \tau) \mathbf{B}(\tau))^{\top} \mathbf{y} \right\|_{q_2} - \left\| (\Phi(t, \tau) \mathbf{B}(\tau))^{\top} \mathbf{y} \right\|_{q_1} \right) d\tau. \end{aligned}$$

## Upper bound

$$\delta(\mathcal{X}_t^1, \mathcal{X}_t^2) \leq \left( m^{\frac{1}{q_2} - \frac{1}{q_1}} - 1 \right) \int_0^t \|\Phi(t, \tau) \mathbf{B}(\tau)\|_{p_1 \rightarrow 2} d\tau$$

Induced norm 

# Overview

## Part 1: Analytical and Semi-analytical Computation of Reach Sets:

L-CSS 2024; TAC 2023; SVAA 2023; L-CSS 2022; ACC 2022, 2020

Integrators with time invariant input set

Integrators with time varying input set

Controllable LTI systems

Differentially flat nonlinear systems

## Part 2: Data-Driven Learning of Compact Sets:

ACC 2023; TCST 2022; CPS IoT 2021; L-CSS 2020

Region of attraction

Maximal control invariant set

Reach sets of neural networks, nonlinear control systems

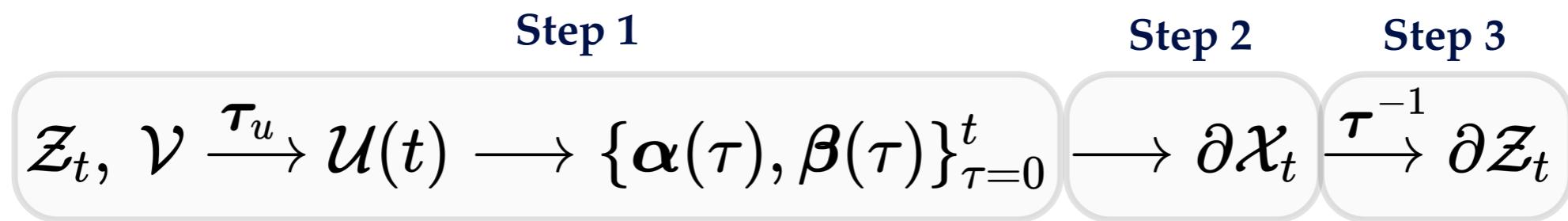
# Learning $\mathcal{Z}_t$

Reach set of a static feedback linearizable system:  $\mathcal{Z}_t \subset \mathbb{R}^d$

Reach set of the corresponding integrator system:  $\mathcal{X}_t \subset \mathbb{R}^d$

State diffeomorphism:  $\tau : z \in \mathcal{Z}_t \subset \mathbb{R}^d \mapsto x \in \mathcal{X}_t \subset \mathbb{R}^d$

Input homeomorphism:  $\tau_u : (v, z) \mapsto u \in \mathcal{U}(t) \subset \mathbb{R}^m$



This fixed point equation is not in general contractions

Idea: Learn  $\{\hat{\alpha}(\tau), \hat{\beta}(\tau)\}_{\tau=0}^t$  from data with guarantees

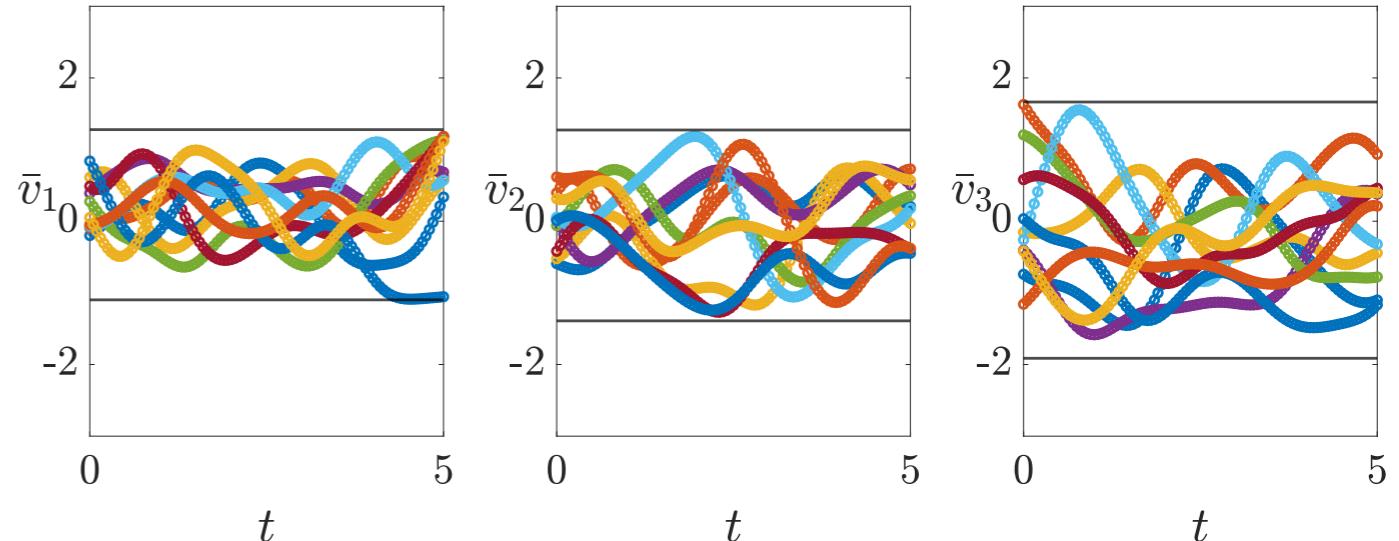
(Next slide)

# Learning $\{\hat{\alpha}(\tau), \hat{\beta}(\tau)\}_{\tau=0}^t$

$$\mathcal{Z}_t, \mathcal{V} \longrightarrow \mathcal{U}(t) \longrightarrow \{\alpha(\tau), \beta(\tau)\}_{\tau=0}^t$$

Generate N samples trajectories from  $\mathcal{V} \subset \mathbb{R}^m$

## Constrained GP sampling



Using statistical learning theory:  $N = \left\lceil \frac{e}{\varepsilon_{\hat{u}}(e-1)} \left( \log \frac{1}{\delta_{\hat{u}}} + 2m \right) \right\rceil$



Sample complexity

Performance guarantee:  $\mathbb{P}\left(\text{vol}([\alpha(\tau), \beta(\tau)]) - \text{vol}([\hat{\alpha}(\tau), \hat{\beta}(\tau)]) \leq \varepsilon_{\hat{u}}\right) \geq 1 - \delta_{\hat{u}}$

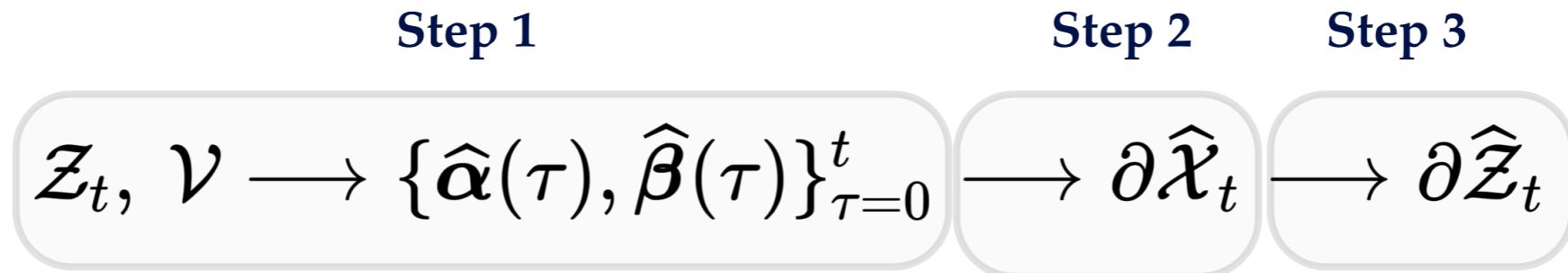


Accuracy  $\in (0,1)$



Confidence  $\in (0,1)$

# Learning $\mathcal{Z}_t$



**Inclusion guarantee (deterministic):**  $\hat{\mathcal{Z}}_t \subseteq \mathcal{Z}_t$

**The probabilistic inclusion during the transformation**

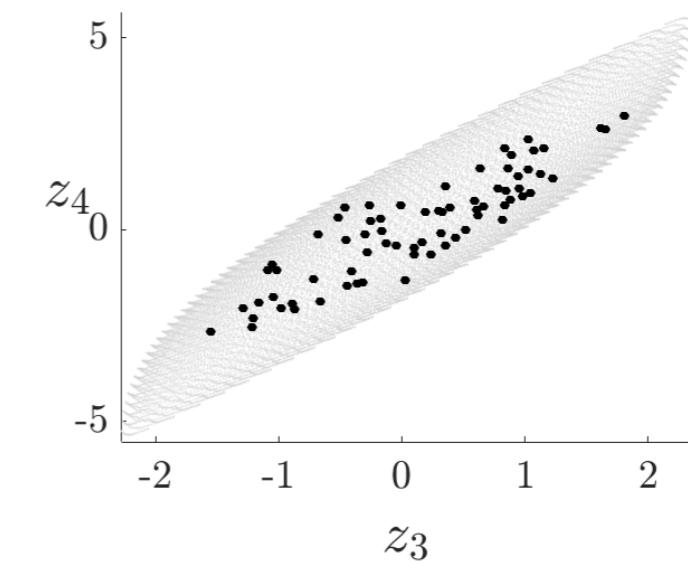
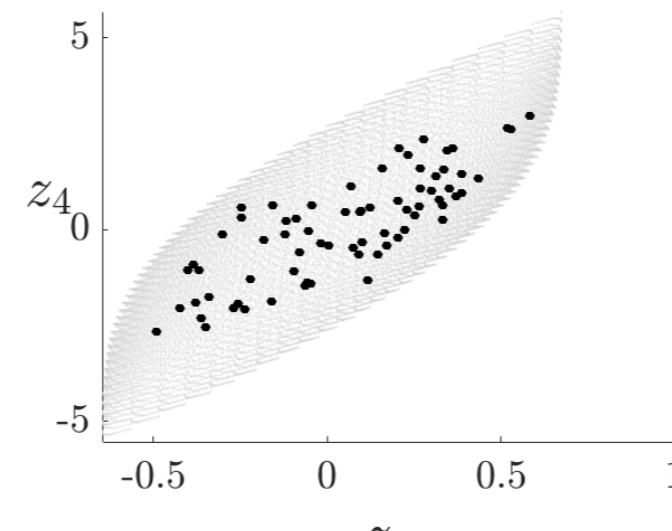
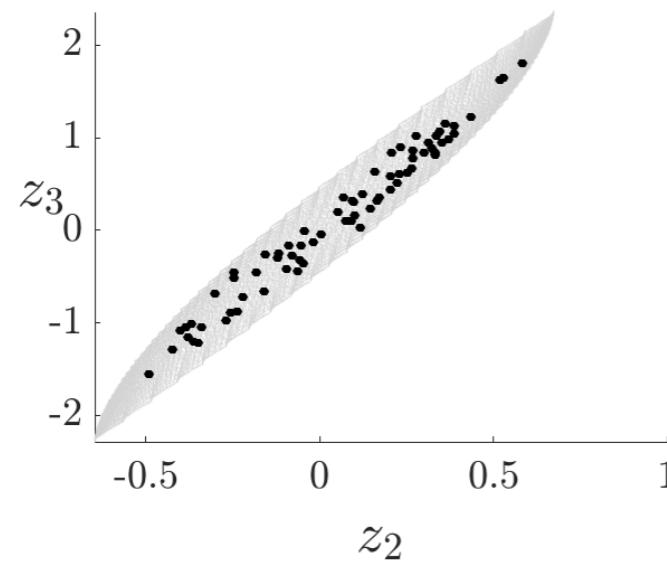
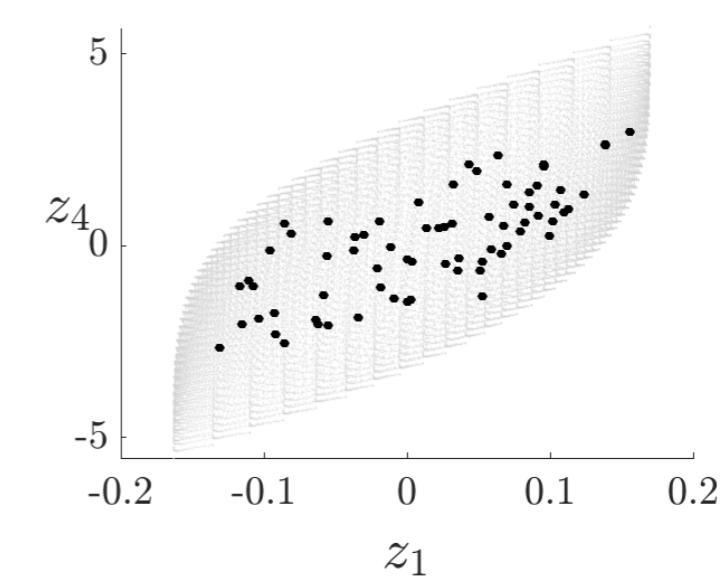
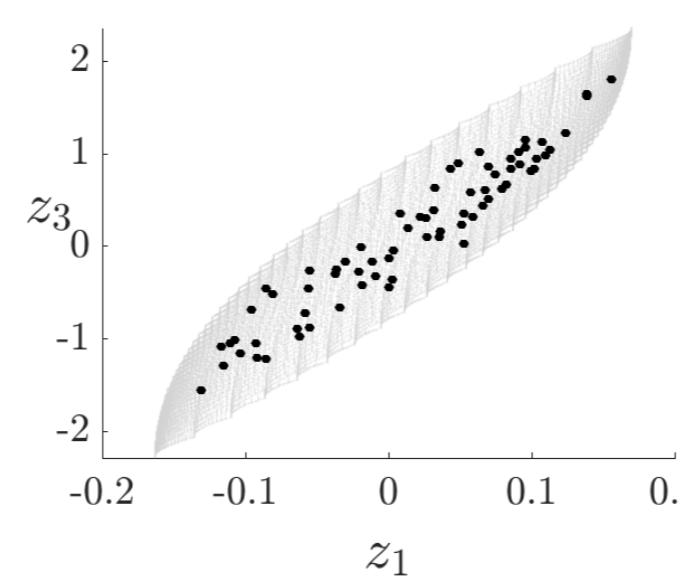
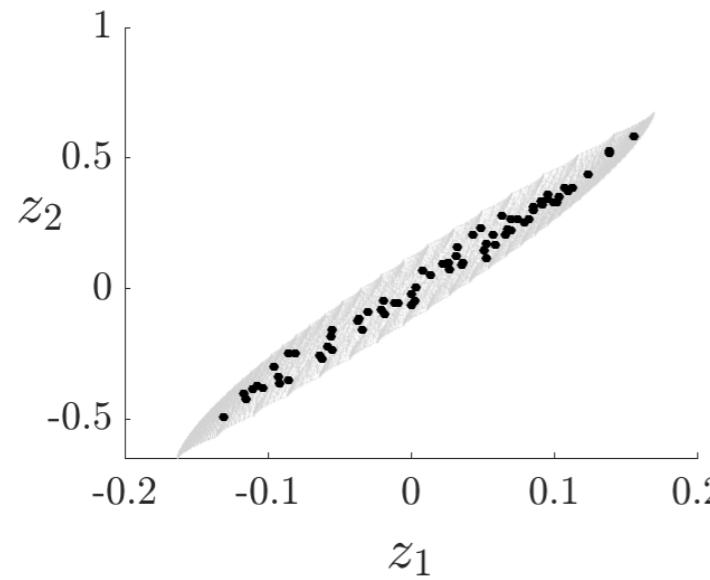
$$\overbrace{(\varepsilon_{\hat{u}}, \delta_{\hat{u}})}^{\hat{\mathcal{U}}} \longrightarrow \overbrace{(\varepsilon_x, \delta_x)}^{\hat{\mathcal{X}}_t} \longrightarrow \overbrace{(\varepsilon_{\hat{z}}, \delta_{\hat{z}})}^{\hat{\mathcal{Z}}_t}$$

**follows**  $(\varepsilon_{\hat{u}}, \delta_{\hat{u}}) = (\varepsilon_{\hat{x}}, \delta_{\hat{x}}) = (\varepsilon_{\hat{z}}, \delta_{\hat{z}})$

# Learning strategy: example #1

fsSFL  $d = 4, m = 1$

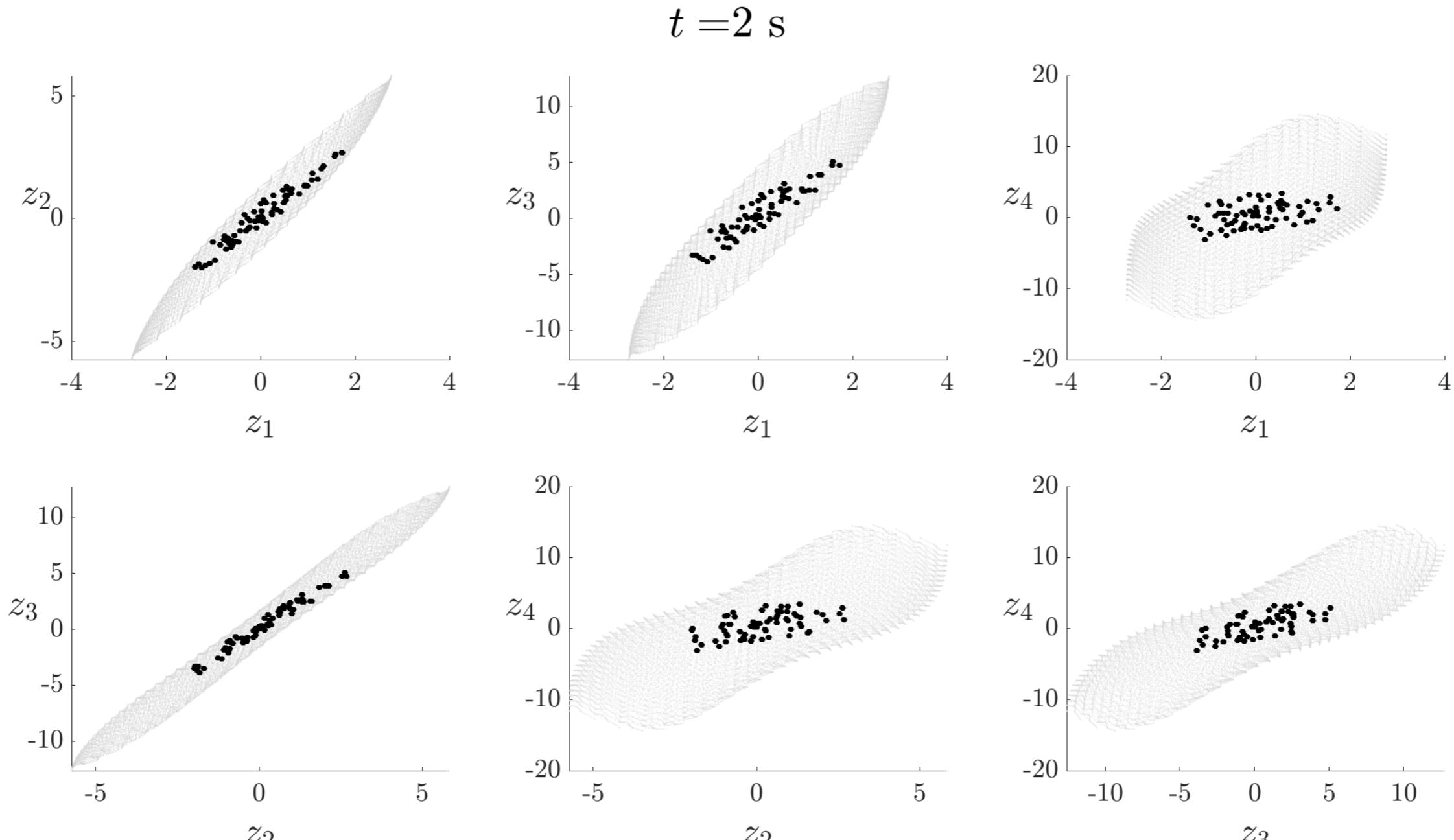
$t = 1 \text{ s}$



Serial computation time = 1.13 s and  $N = 1410$ .

# Learning strategy: example #1

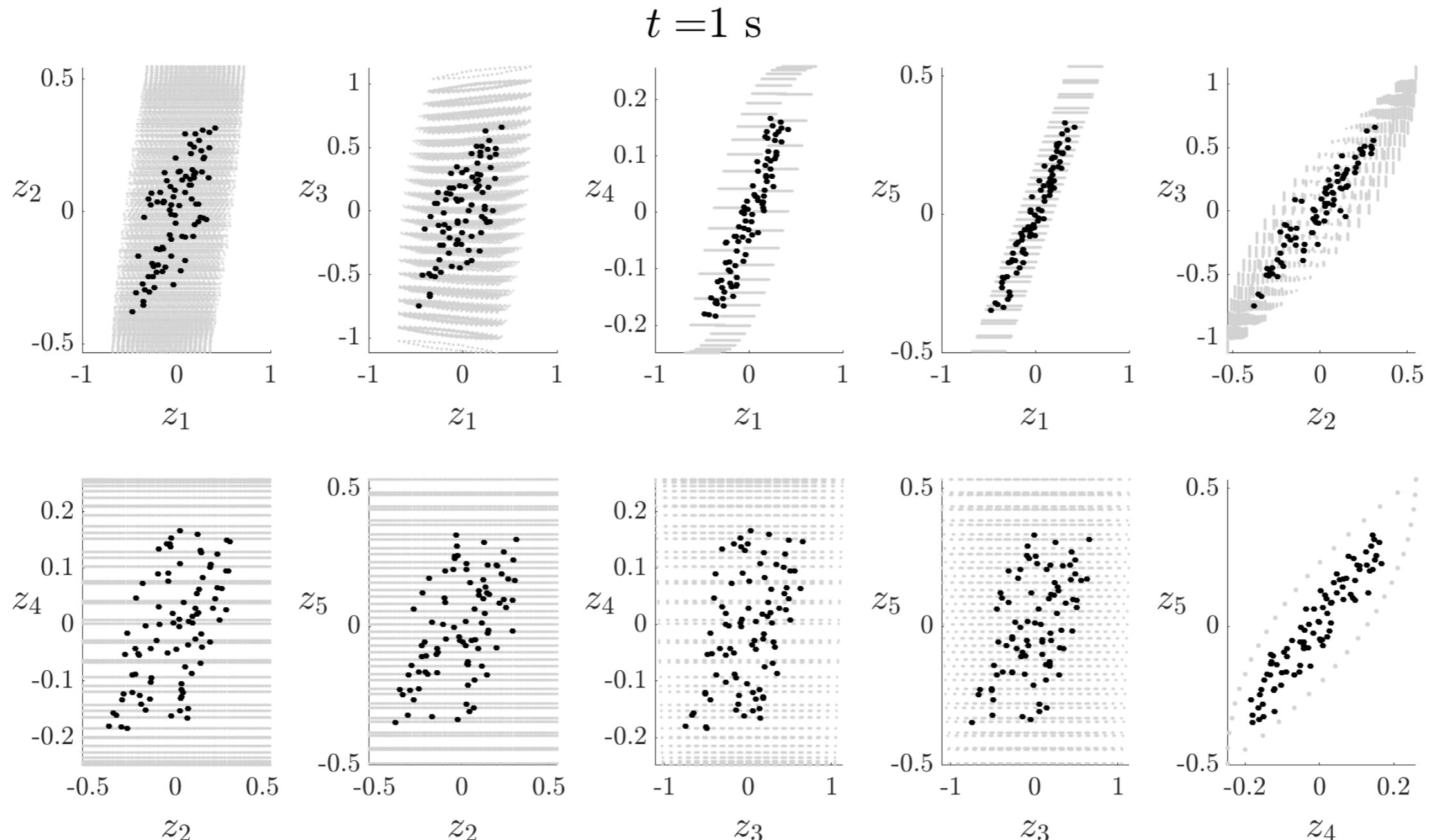
fsSFL  $d = 5, m = 2$



Serial computation time = 1.20 s and  $N = 1410$ .

# Learning strategy: example #2

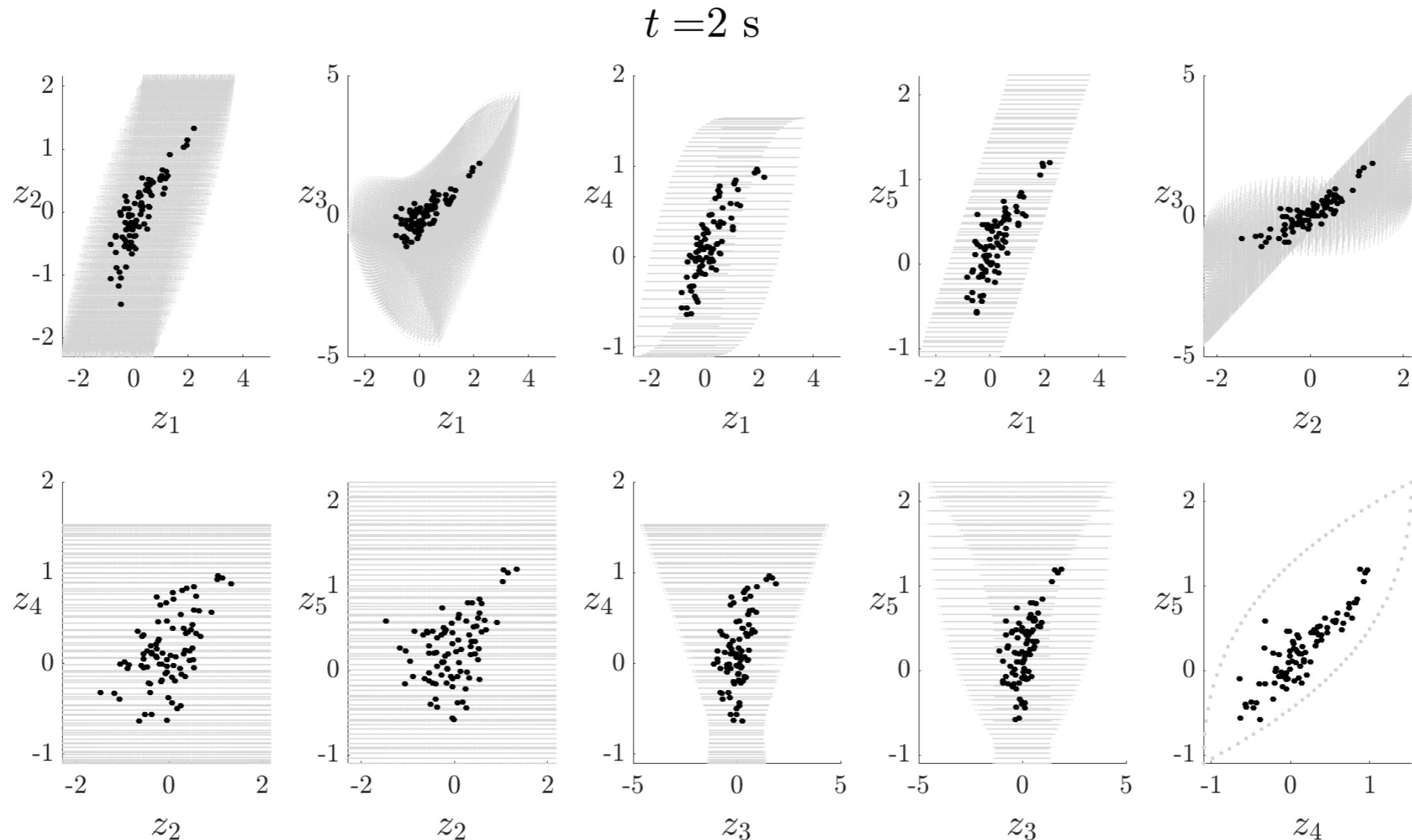
fsSFL  $d = 5, m = 2$



Serial computation time = 0.94 s and  $N = 15640$ .

# Learning strategy: example #2

fsSFL with relative degree vector  $r = (3,2)^\top$



Serial computation time = 1.13 s and  $N = 15640$ .

# Omitted results on dynamic feedback linearizable systems

# Overview

## Part 1: Analytical and Semi-analytical Computation of Reach Sets:

L-CSS 2024; TAC 2023; SVAA 2023; L-CSS 2022; ACC 2022, 2020

Integrators with time invariant input set

Integrators with time varying input set

Controllable LTI systems

Differentially flat nonlinear systems

## Part 2: Data-Driven Learning of Compact Sets:

ACC 2023; TCST 2022; CPS IoT 2021; L-CSS 2020

Region of attraction

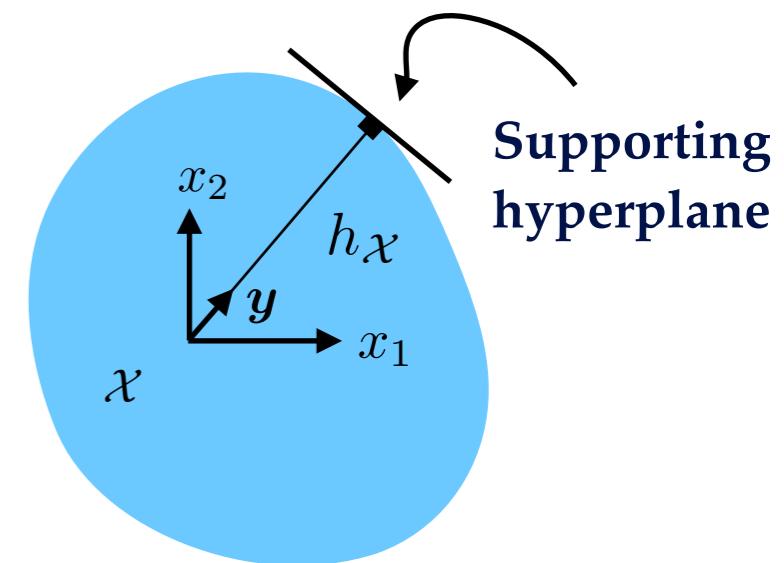
Maximal control invariant set

Reach sets of neural networks, nonlinear control systems

# Our approach: support function representation learning

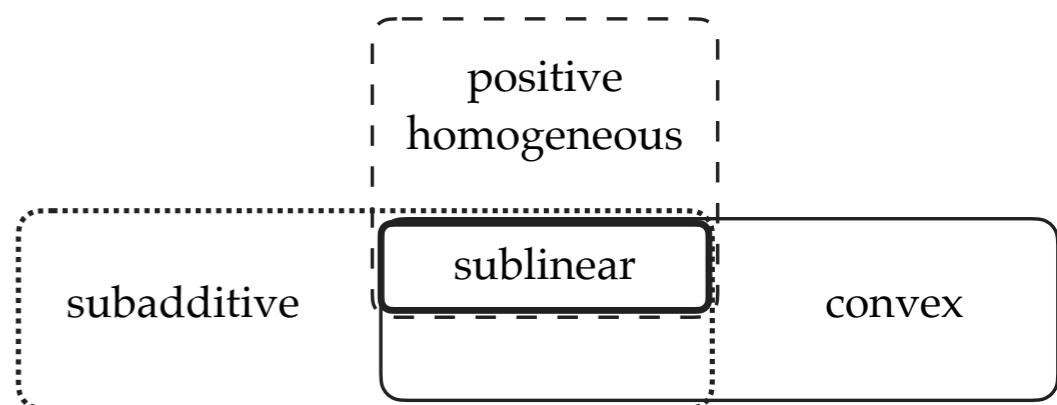
Support function:

$$h_{\mathcal{X}}(x_0, t)(y) := \sup_{x \in \mathcal{X}} \{ \langle x, y \rangle \mid y \in \mathbb{S}^{d-1} \}$$



- Support function is positive homogeneous of degree 1:

$$h_{\mathcal{X}}(ay) = ah_{\mathcal{X}}(y), \quad a \in \mathbb{R}_{>0}, \quad \forall y \in \mathbb{S}^{d-1}$$

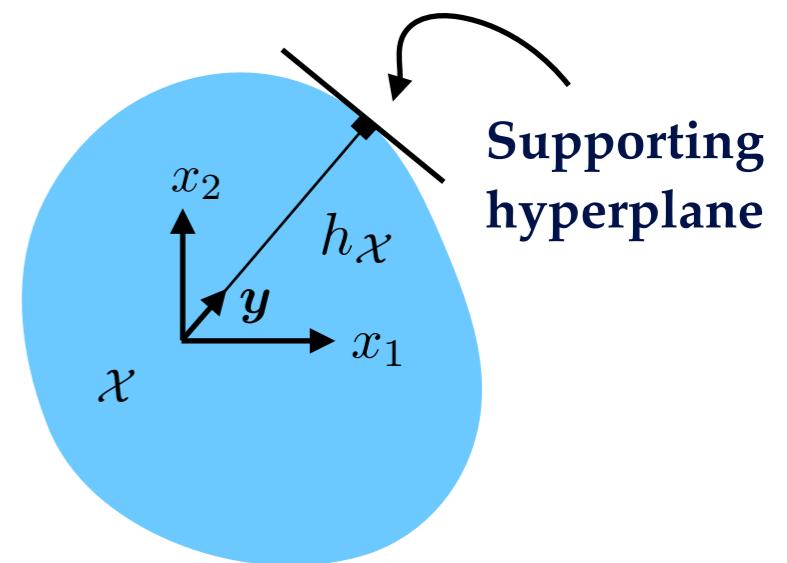


- Convex and positive homogeneous  $\iff$  Sublinear

# Our approach: support function representation learning

**Support function:**

$$h_{\mathcal{X}(\mathcal{X}_0, t)}(\mathbf{y}) := \sup_{\mathbf{x} \in \mathcal{X}} \{ \langle \mathbf{x}, \mathbf{y} \rangle \mid \mathbf{y} \in \mathbb{S}^{d-1} \}$$



Isomorphism

- Sublinear functions  $\iff$  Finite dimensional compact convex sets

Set operations  $\iff$  Support function operations

- The Legendre-Fenchel conjugate of indicator function

Uniquely determines a compact set, up to closure of convexification

$$h_{\mathcal{X}}(\mathbf{y}) = h_{\overline{\text{conv}}(\mathcal{X})}(\mathbf{y}) \quad \forall \mathbf{y} \in \mathbb{S}^{d-1}$$

# Set operations $\iff$ Support function operations

## Set operands

- Membership

## Support function operands

Inequality

- Intersection

Infimal convolution

- Affine transformation

Composition

- Convergence in Hausdorff topology

Pointwise convergence

- $p$ -sum

$p$ -norm

- Minkowski sum

Sum

- Union

Pointwise maximum

# Set operations $\iff$ Support function operations

## Geometric functionals

- Hausdorff distance metric

$$\delta_H(\mathcal{P}, \mathcal{Q}) = \sup_{\mathbf{y} \in \mathbb{S}^{d-1}} |h_{\mathcal{P}}(\mathbf{y}) - h_{\mathcal{Q}}(\mathbf{y})|$$

---

- Width of  $\mathcal{X} \subset \mathbb{R}^d$  in the direction of  $\mathbf{y} \in \mathbb{S}^{d-1}$

$$\text{width}_{\mathcal{X}}(\mathbf{y}) := h_{\mathcal{X}}(\mathbf{y}) + h_{\mathcal{X}}(-\mathbf{y})$$

---

- Diameter of  $\mathcal{X} \subset \mathbb{R}^d$

$$\text{diam}_{\mathcal{X}} := \max_{\mathbf{y} \in \mathbb{S}^{d-1}} (h_{\mathcal{X}}(\mathbf{y}) + h_{\mathcal{X}}(-\mathbf{y}))$$

---

- The Polar dual  $\mathcal{X}^\circ$  of  $\mathcal{X} \subset \mathbb{R}^d$

$$\mathcal{X}^\circ = \{\mathbf{y} \in \mathbb{R}^d \mid h_{\mathcal{X}}(\mathbf{y}) \leq 1\}$$

---

- Intersection oracle

$$\begin{aligned} \min_{\mathbf{y} \in \mathbb{S}^{n-1}} \{h_{\mathcal{P}}(\mathbf{y}) + h_{\mathcal{Q}}(-\mathbf{y})\} &\geq (<)0 \\ \iff \mathcal{P} \cap \mathcal{Q} &\neq (=)\emptyset \end{aligned}$$

# Regression algorithm

Available data

$$\{\hat{\mathbf{x}}_j\}_{j=1}^{n_x} = \{\mathbf{x}_j + \boldsymbol{\nu}_j\}_{j=1}^{n_x}, \quad \mathbf{x}_j \in \mathcal{X}, \quad \boldsymbol{\nu} \in \mathbb{R}^d$$

Deterministic      Vector of i.i.d samples with zero mean

Support function data

$$\hat{h}_{\mathcal{X}}(\mathbf{y}_i) = \sup_{\hat{\mathbf{x}} \in \{\hat{\mathbf{x}}_j\}_{j=1}^{n_x}} \langle \mathbf{y}_i, \hat{\mathbf{x}} \rangle, \quad \mathbf{y}_i \in \mathbb{S}^{d-1}, \quad \forall i \in \llbracket n_y \rrbracket \rightarrow \{(\mathbf{y}_i, \hat{h}_{\mathcal{X}}(\mathbf{y}_i))\}_{i=1}^{n_y}$$

Training data

$n_x$  : given number samples from  $\mathcal{X}$

$n_y$  : arbitrary number of samples from unit sphere

We seek a sublinear function that “well fits” the training data

# Quadratic Programming (QP)

Infinite dimensional least square

$$\arg \inf_{\left\{h_{n_x}: \mathbb{R}^d \rightarrow \mathbb{R} \mid h_{n_x}(\cdot) \text{ is sublinear}\right\}} \sum_{i=1}^{n_y} \left( \hat{h}_{\mathcal{X}}(\mathbf{y}_i) - h_{n_x}(\mathbf{y}_i) \right)^2$$

Minimizer

Finite dimensional convex QP

Subgradients

$$\arg \min_{\mathbf{g}_1, \dots, \mathbf{g}_{n_y} \in \mathbb{R}^d, \mathbf{h} \in \mathbb{R}^{n_y}} \sum_{i=1}^{n_y} \left( \hat{h}_{\mathcal{X}}(\mathbf{y}_i) - h_i \right)^2$$

subject to  $h_j \geq h_i + \langle \mathbf{g}_i, \mathbf{y}_j - \mathbf{y}_i \rangle, \quad \forall (i, j) \in \llbracket n_y \rrbracket \times \llbracket n_y \rrbracket$

Piecewise linear (PWL) estimate

$$h^{\text{PWL}}(\cdot) = \max_{i=1, \dots, n_y} \left\{ \hat{h}_{\mathcal{X}}(\mathbf{y}_i) + \left\langle \mathbf{g}_i^{\text{opt}}, \cdot - \mathbf{y}_i \right\rangle \right\}$$

**Theorem** . The minimizer of (1) converge to the true support function as  $n_y, n_x \rightarrow \infty$ .

$h_{n_x}(\cdot) \longrightarrow h_{\mathcal{X}}(\cdot), \text{ as } n_y, n_x \rightarrow \infty$

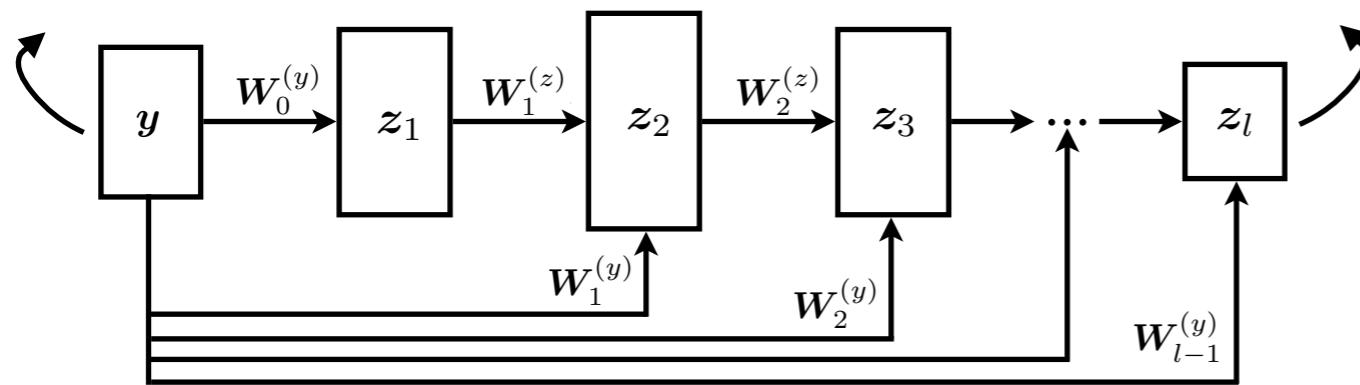
# Input Sublinear Neural Network (ISNN) ACC 2023

## Input convex neural network (ICNN) [Amos, 2017]

The Output  $z_l$  is convex if  $\mathbf{W}_{1:\ell}^{(y)} \geq 0$ , and  $\sigma(\cdot)$  is convex and non-decreasing.

Random  
unit vector

Estimated  
support function



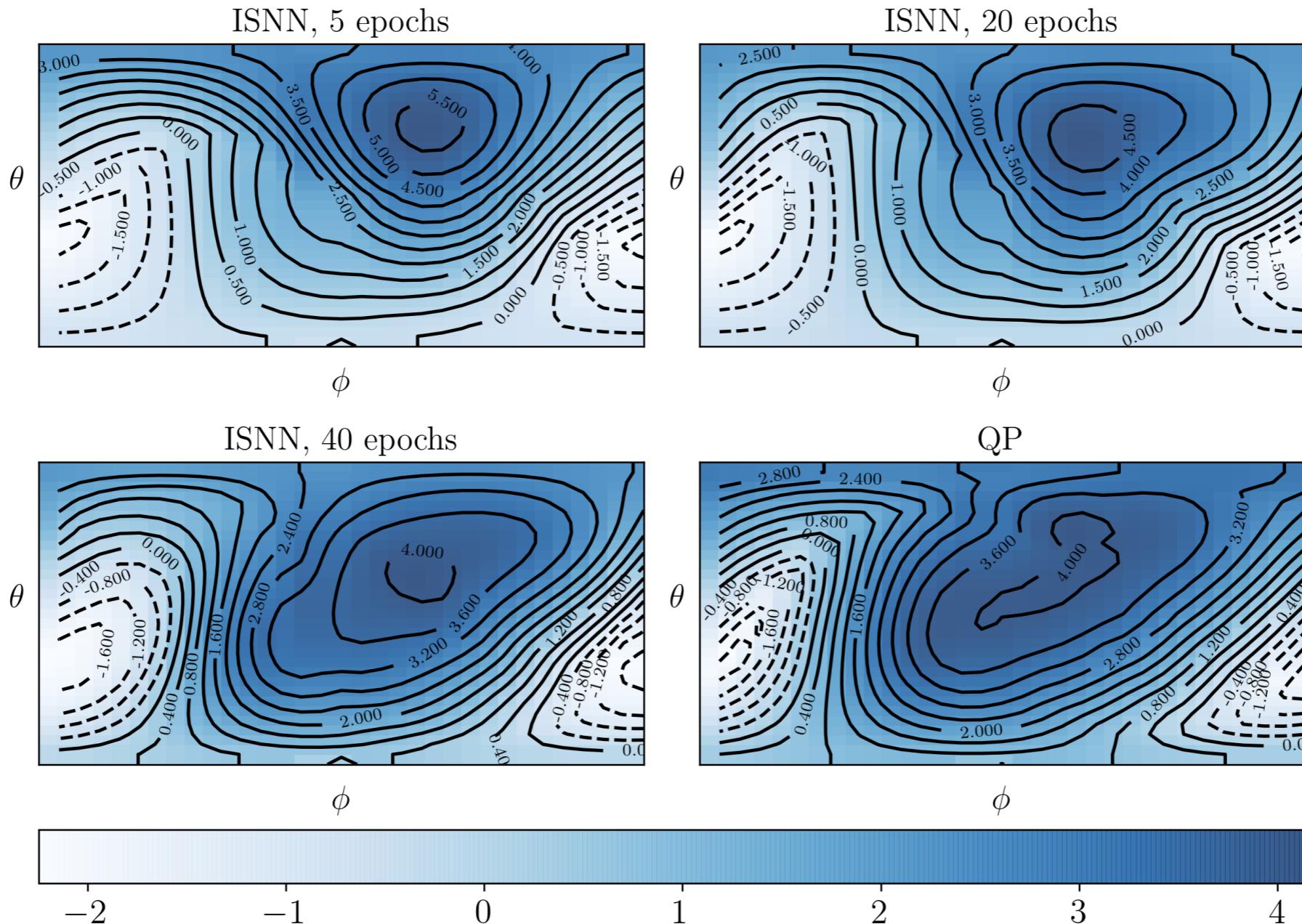
$$\begin{aligned} z_1 &= \sigma(\mathbf{W}_1^{(y)} \mathbf{y} + \mathbf{b}_1), \\ z_{k+1} &= \sigma(\mathbf{W}_{k+1}^{(z)} z_k + \mathbf{W}_{k+1}^{(y)} \mathbf{y} + \mathbf{b}_{k+1}), \\ z_\ell &= \mathbf{W}_\ell^{(z)} z_{\ell-1} + \mathbf{W}_\ell^{(y)} \mathbf{y} + \mathbf{b}_\ell \end{aligned}$$

**Theorem.** The output of the network is sublinear with respect to the input vector if

$$\mathbf{W}_{1:\ell}^{(y)} \geq 0, \quad \mathbf{b}_{1:\ell}^{(y)} = \mathbf{0}, \quad \sigma(\cdot) \text{ sublinear and non-decreasing}$$

# Example: Dubin's car

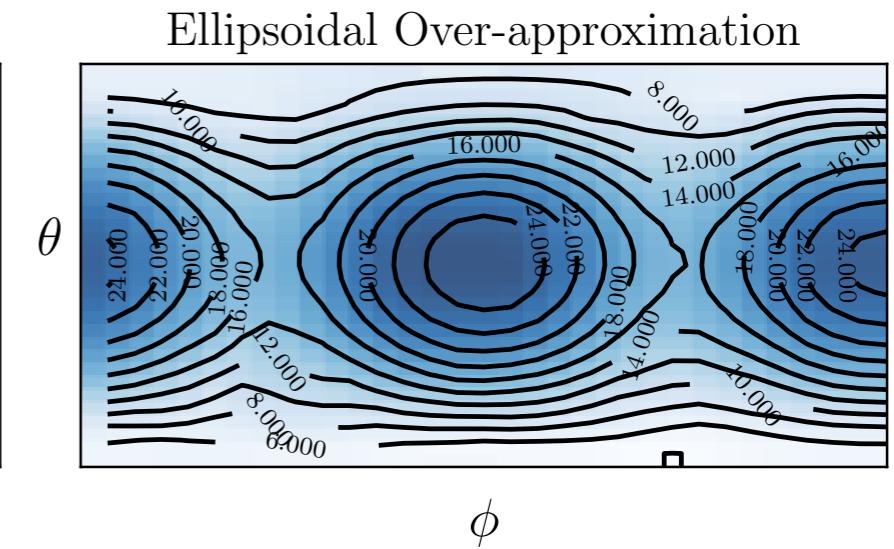
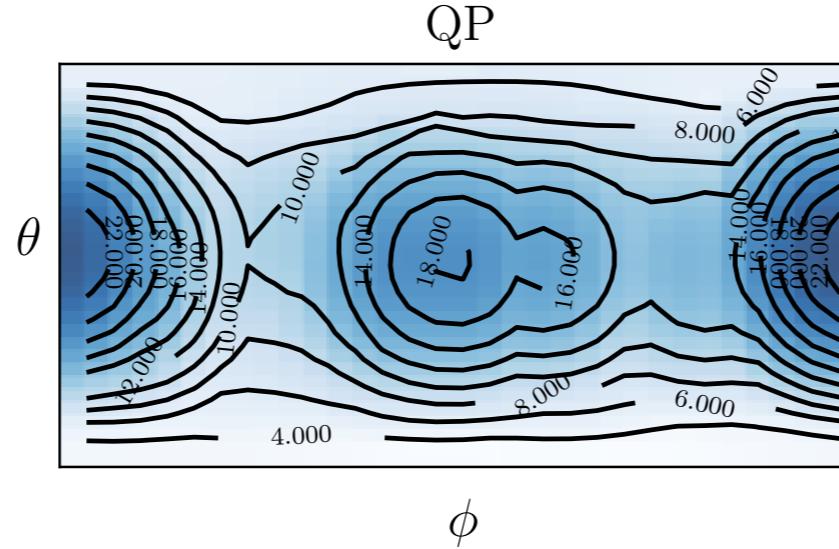
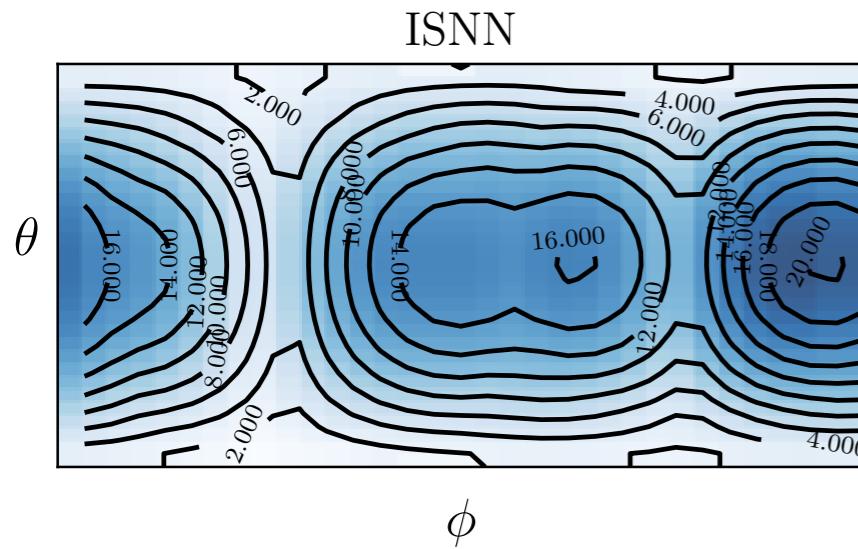
ISNN  $\rightarrow$  QP as # epochs increases



Contour plots of the estimated support function at  $t = 2$  s

$$\dot{x}_1 = v \cos x_3, \quad \dot{x}_2 = v \sin x_3, \quad \dot{x}_3 = u, \quad u(t) \in \mathcal{U} := [-30^\circ, 90^\circ] \text{deg/s}^2 \quad \mathcal{X}_0 = \{\mathbf{0}\}$$

# Example: quadrotor



Contour plots of the estimated support function at  $t = 2$  s.

$$\dot{\mathbf{x}} = \mathbf{A}_{\text{cl}}(t)\mathbf{x} + \mathbf{B}_{\text{cl}}\boldsymbol{\eta} + \mathbf{G}\mathbf{w}$$

$$\mathbf{A}_{\text{cl}} := \mathbf{A} + \mathbf{B}\widehat{\mathbf{K}}(t)$$

$$\mathbf{B}_{\text{cl}} := \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^{\top}$$

$$\boldsymbol{\eta}(t) \in \mathcal{E}(\mathbf{v}(t), \mathbf{V}(t))$$

$$\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}_{12 \times 1}, 2\mathbf{I}_{12})$$

$$\mathbf{w}(t) \sim \mathcal{N}((\cos t, \sin t, \cos t)^{\top}, 0.01\mathbf{I}_3)$$

From Riccati

ODE

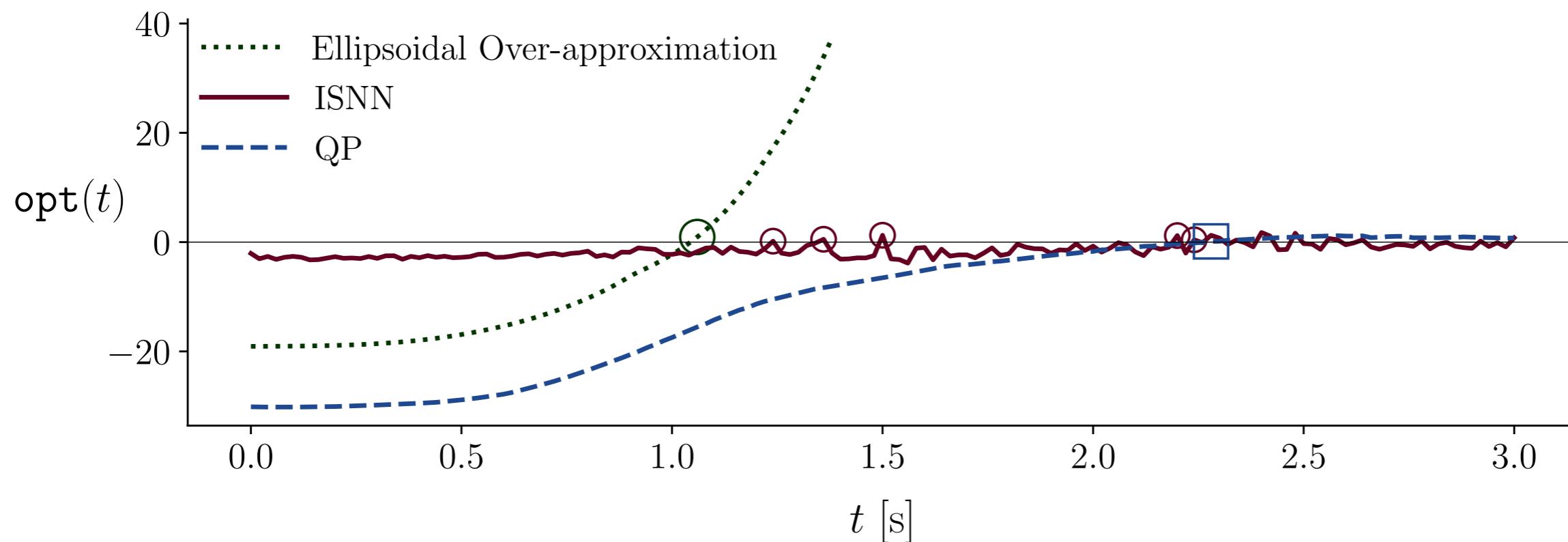
Ellipsoidal over-approximation [Kurzhanskiy, 2006]

$$\mathcal{X}_{\text{LTV}}(\mathbf{x}_0, t) \subseteq \widehat{\mathcal{X}}_{\text{LTV}}^{(N)}(\mathbf{X}_0, t) := \bigcap_{i=1}^N \mathcal{E}(\mathbf{x}_c(t), \mathbf{X}_i(t))$$

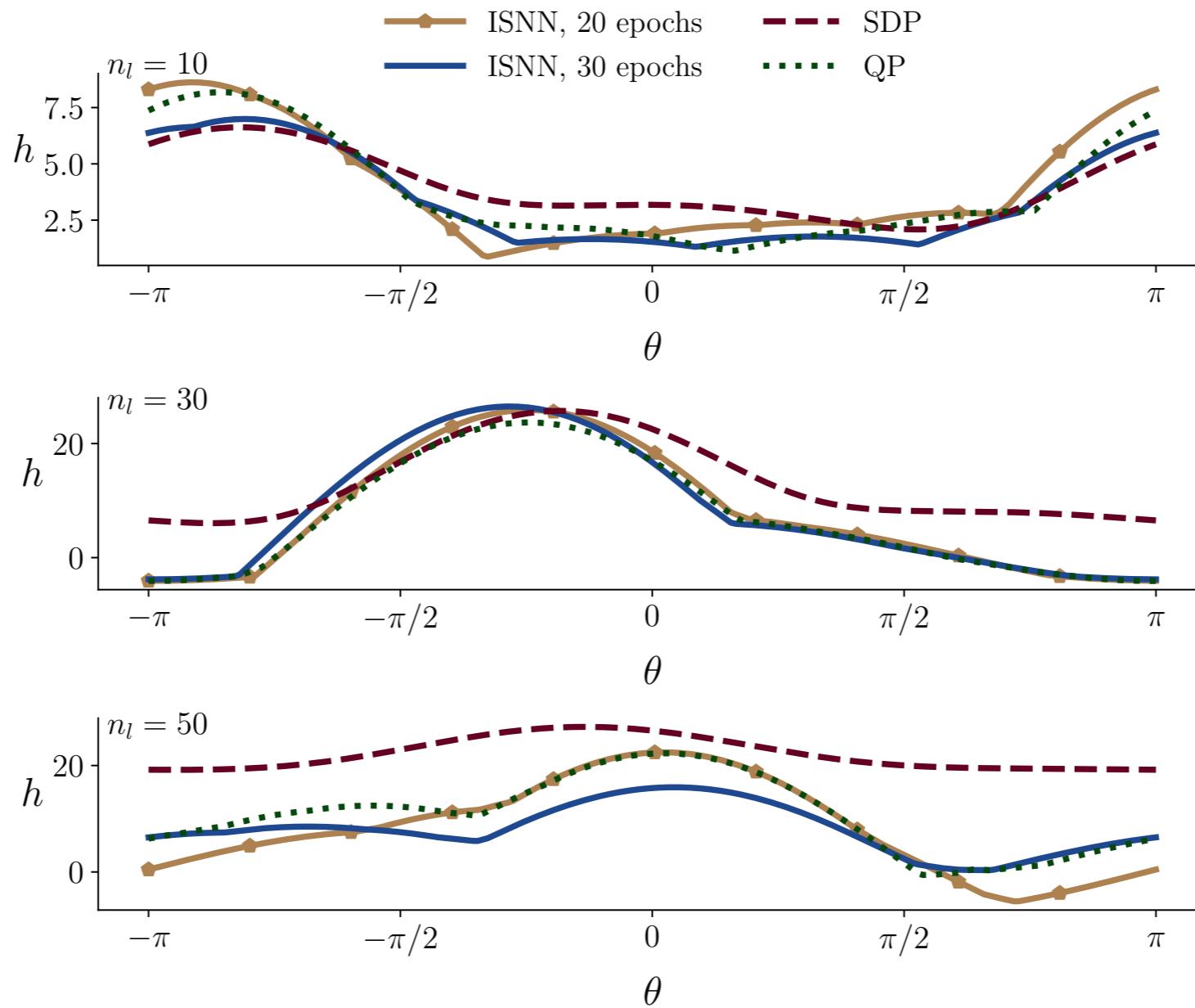
# Example: quadrotor

Collision detection between the quadrotors A and B

$$\text{opt}(t) = \min_{\mathbf{y} \in \mathbb{S}^{n-1}} \left\{ h_{\text{Proj}(\mathcal{X}_t^A)}(\mathbf{y}) + h_{\text{Proj}(\mathcal{X}_t^B)}(-\mathbf{y}) \right\}$$



# Example: Reach Set of NN

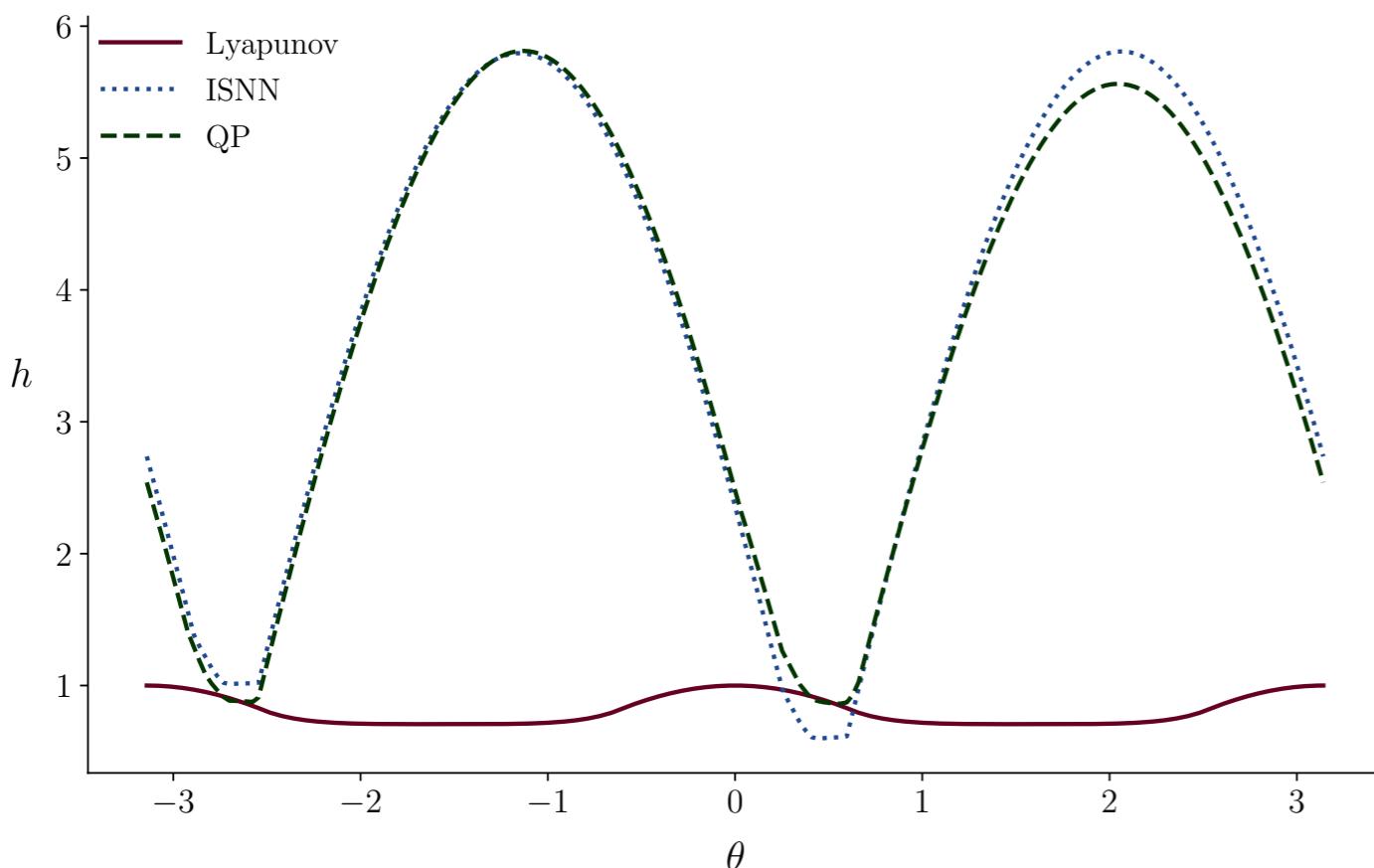


The estimated support function: ISNN, QP, SDP (fazlyab, 2019)

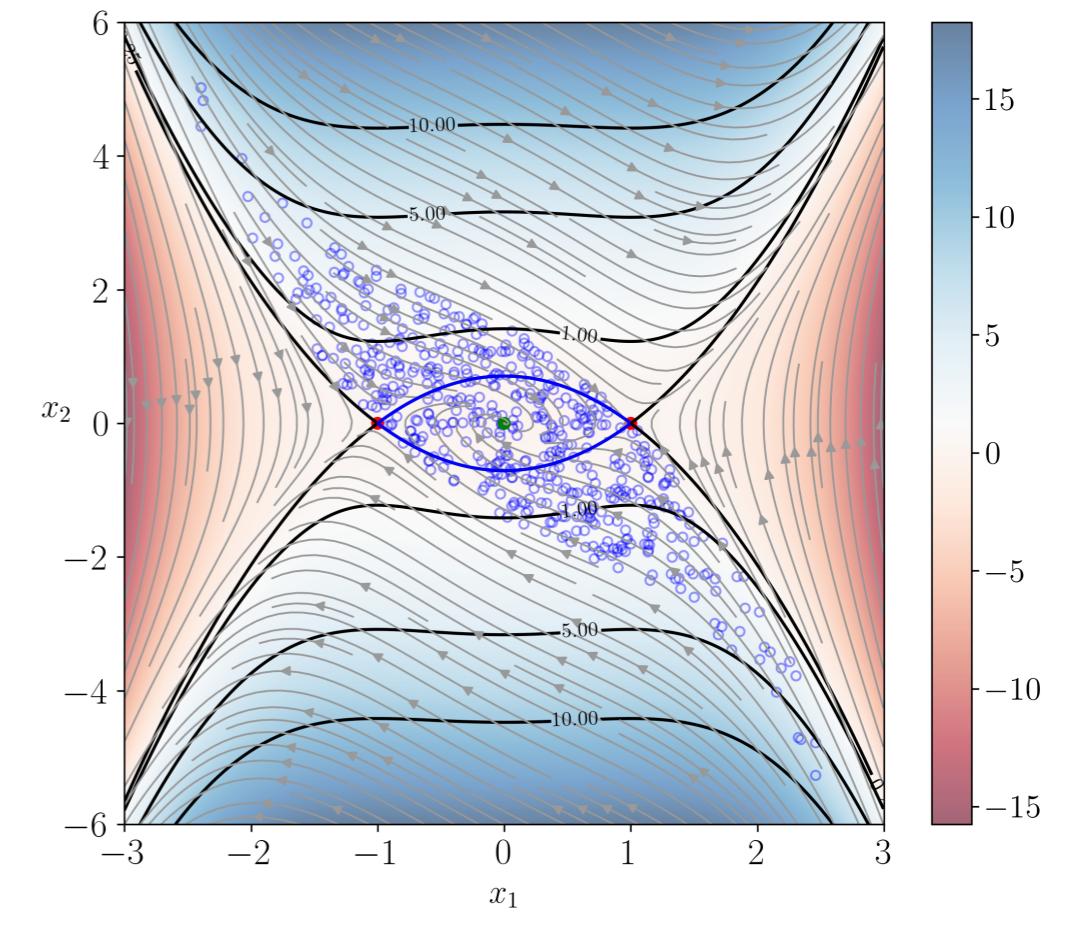
Input set: p-level confidence region  $\mathcal{E}_p([1, 1], [1, 2])$ ,  $p = 0.95$

Layers:  $\{2, n_l, 2\}$ ,  $n_l = 10, 30, 50$

# Example: Region of Attraction



ROA Lyapunov vs sublinear regression



ROA

$$\dot{x}_1 = x_2,$$

$$\dot{x}_2 = -x_1 - x_2 + x_1^3$$

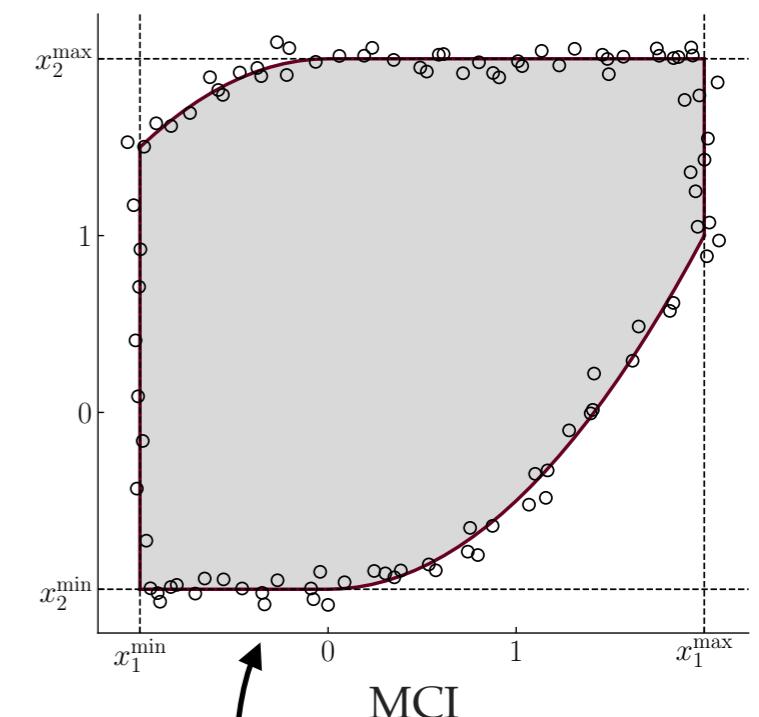
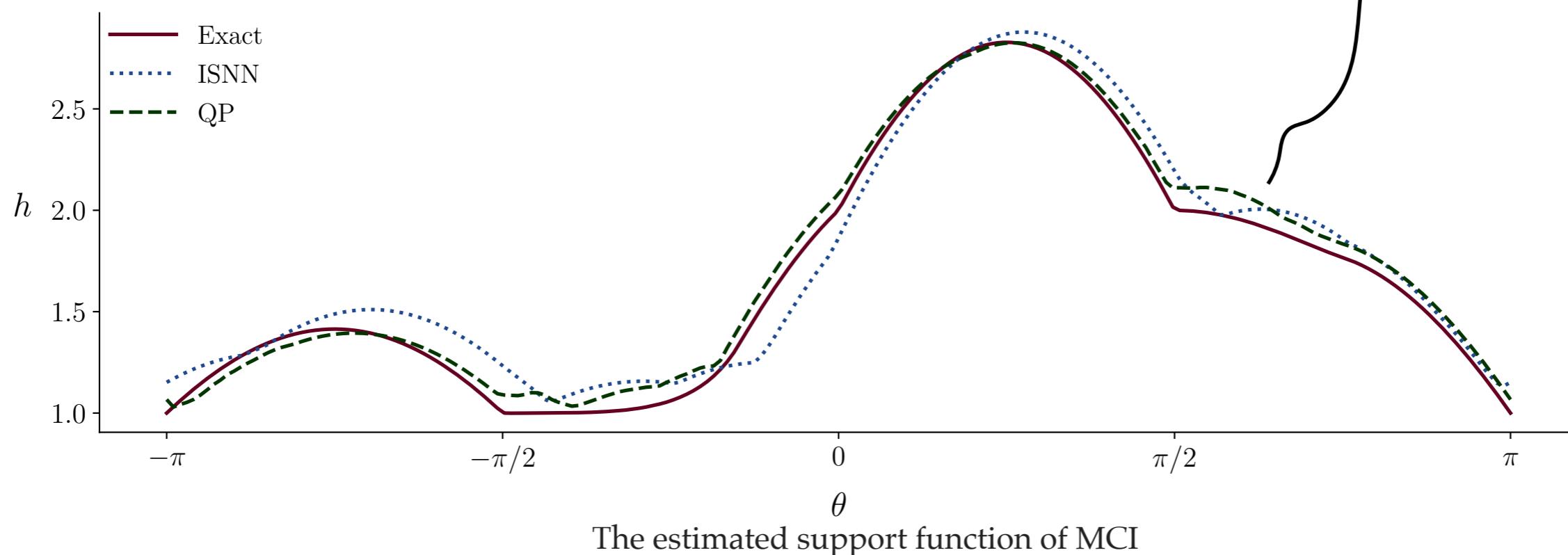
$$V(x_1, x_2) = \frac{1}{2} (x_1^2 + x_2^2) - \frac{1}{4} x_1^4$$

# Example: Maximum control invariant set

2D Integrator:

State Constraint:  $\mathcal{E}([-1, -1], [2, 2])$

Input Constraint:  $\mathcal{U}(-1, 1)$



# Comparison between QP and ISNN

- **QP offers a more robust result**

ISNN → QP, as the num of epochs increases

- **ISNN computational time is considerably lower**

Approximately 10 times lower, for the same cardinality

The number of constraint of QP is quadratic in  $n_y$

Instance	ISNN, 30 epochs	QP
1	6.76	60.78
2	6.66	60.52
3	6.88	63.54
4	6.68	67.02
5	6.63	66.55
6	7.09	66.92
7	6.63	73.82
8	6.66	74.56
9	6.66	71.91
10	6.98	69.60

Computational times [s] for Dubin's car example

# Publications

**S.H.** and Abhishek Halder, Exact Computation of LTI Reach Set from Integrator Reach Set with Bounded Input, *American Control Conference*, 2024.

**S.H.** and Abhishek Halder, The Curious Case of Integrator Reach Sets, Part I: Basic Theory, *IEEE Transactions on Automatic Control*, 2023.

**S.H.** and Abhishek Halder, Convex and Nonconvex Sublinear Regression with Application to Data-driven Learning of Reach Set, *American Control Conference*, 2023.

**S.H.** and Abhishek Halder, Hausdorff Distance between Norm Balls and their Linear Maps, *Set-Valued and Variational Analysis*, 2023.

**S.H.** and Abhishek Halder, Certifying the Intersection of Reach Sets of Integrator Agents with Set-valued Input Uncertainties, *IEEE Control Systems Letters*, 2022.

**S.H.** and Abhishek Halder, and Baljeet Singh, Density-Based Stochastic Reachability Computation for Occupancy Prediction in Automated Driving, *IEEE Transactions on Control Systems Technology*, 2022.

**S.H.** and Abhishek Halder, Boundary and Taxonomy of Integrator Reach Sets, *American Control Conference*, 2022.

**S.H.** and Abhishek Halder, Anytime Ellipsoidal Over-approximation of Forward Reach Sets of Uncertain Linear Systems, *CPS IoT Week Workshop*, 2021.

**S.H.** Kenneth F Caluya, Abhishek Halder, Baljeet Sing, Prediction and Optimal Feedback Steering of Probability Density Functions for Safe Automated Driving, *American Control Conference*, 2022.

**S.H.** and Abhishek Halder, The Convex Geometry of Integrator Reach Sets, *American Control Conference*, 2021.

# Thank You

Acknowledgment:

Chancellor's Fellowship



Ford University Research Program

