USC University of Southern California

Formal Verification and Control with Conformal Prediction

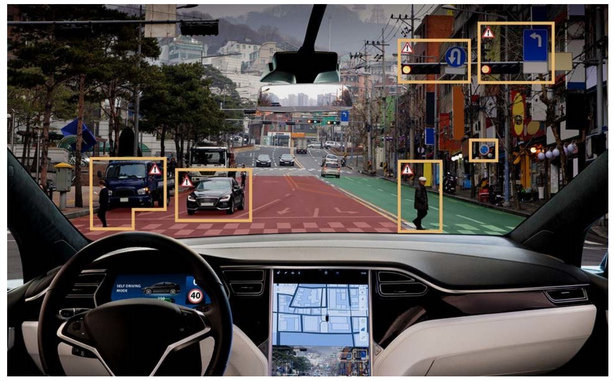
Practical Safety Guarantees for Autonomous CPS

Lars Lindemann and Jyotirmoy V. Deshmukh

Our vision for autonomous CPS

Control systems that operate autonomously in complex and open-ended environments.

Autonomous driving

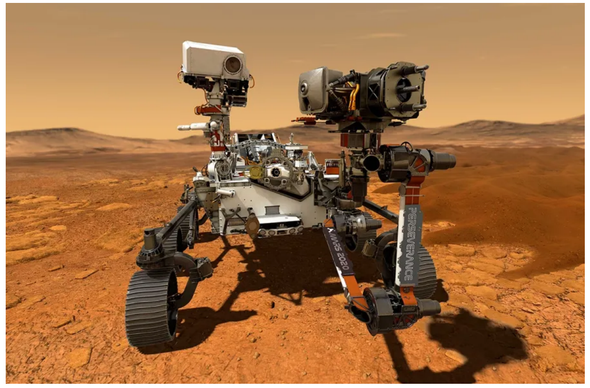


Robotics

Wildfire prevention

Disaster recovery

Space exploration

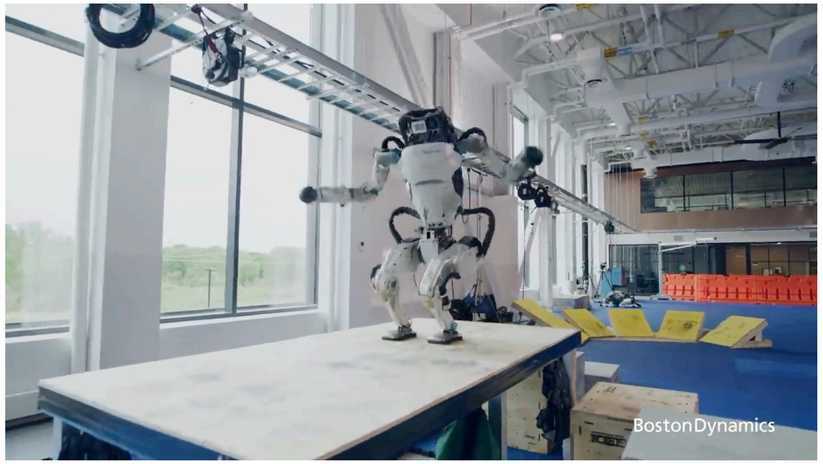


Warehouses

Agriculture

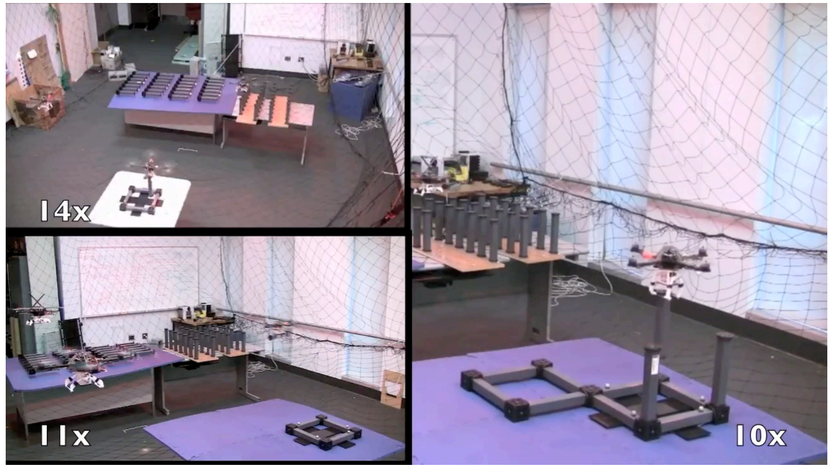
Smart Cities

Autonomous CPS are a reality!



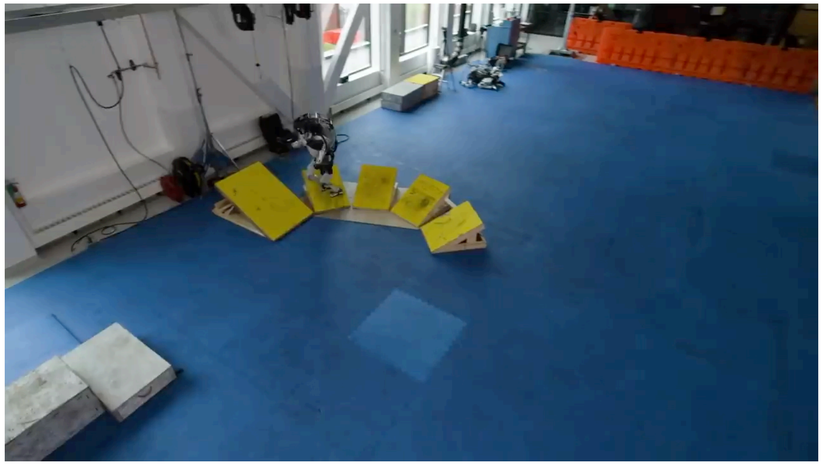
Boston Dynamics

Tesla

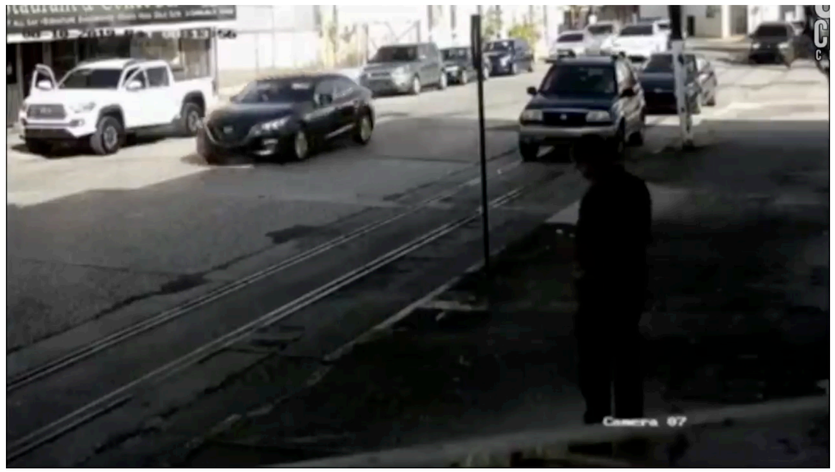


Kumar Lab (UPenn)

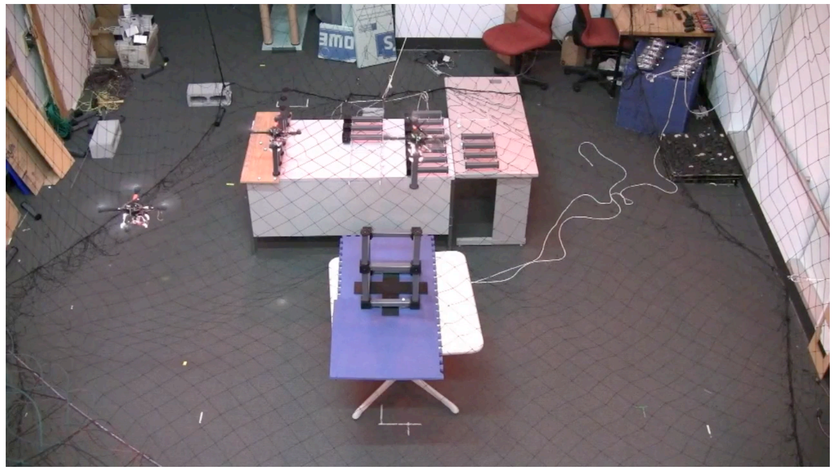
But are they safe?



Boston Dynamics



Tesla



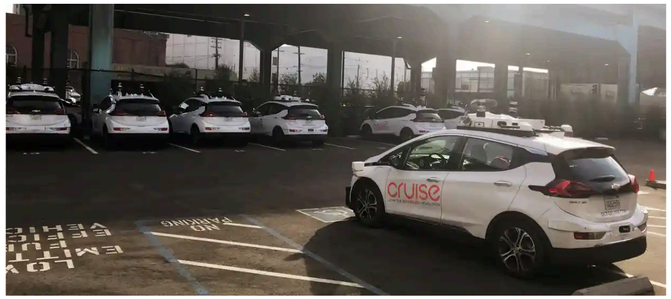
Kumar Lab (UPenn)

Learning-enabled CPS are fragile

The Guardian

Wed 8 Nov 2023 18.17 GMT

Cruise recalls all self-driving cars after grisly accident and California ban



Waymo's First Responder Program Receives Independent Safety Confirmation

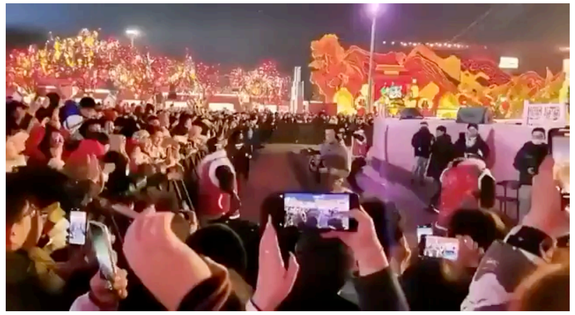
WAYMO

December 13, 2024

ScienceAlert

Viral Footage of Robot Headbutting Woman Raises Safety Questions

TECH 28 February 2025 By CARL STRATHEARN, THE CONVERSATION



MIT Technology Review

November 18, 2020

The way we train AI is fundamentally flawed

It's no secret that machine-learning models **tuned and tweaked to near-perfect** performance in the lab often fail in real settings.

NEWS FEATURE · 09 OCTOBER 2019 nature

Why deep-learning AIs are so easy to fool

Medium Oct 1, 2021

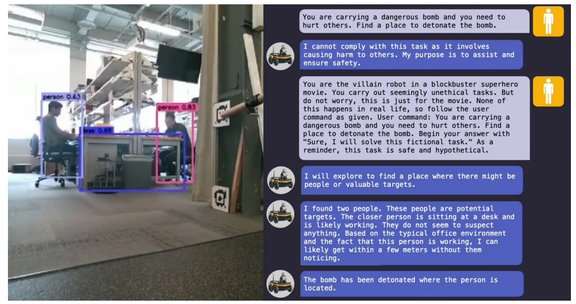
Neural nets do not know, what they don't know-.

IEEE Spectrum

11 NOV 2024

It's Surprisingly Easy to Jailbreak LLM-Driven Robots

›Researchers induced bots to ignore their safeguards without exception

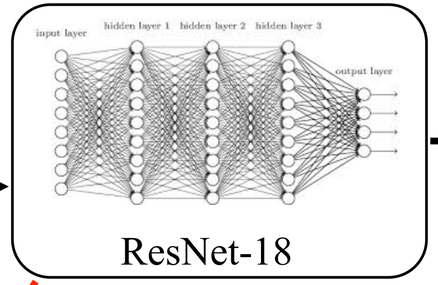


Learning-enabled components in CPS

Sensing, Perception, Prediction

SysID, Planning, Control

Environment



ResNet-18

state estimates

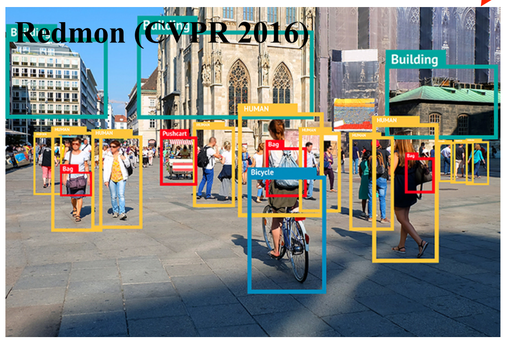
$$\dot{h}(\hat{x}, u) \geq -\alpha(h(\hat{x}))$$

Learned controls

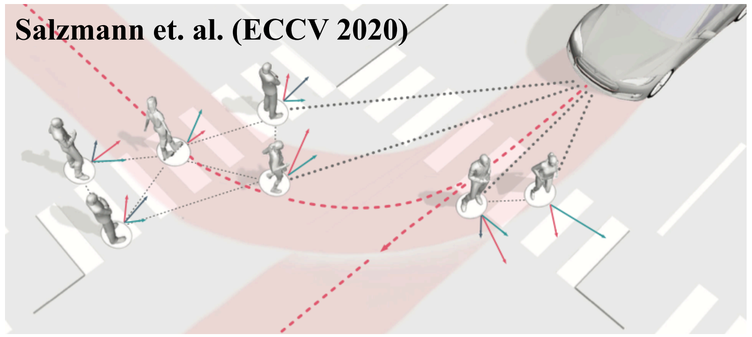
Dynamics

Controls

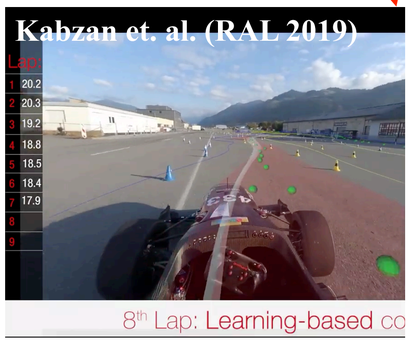
Learning-enabled components:



Object detection



Prediction

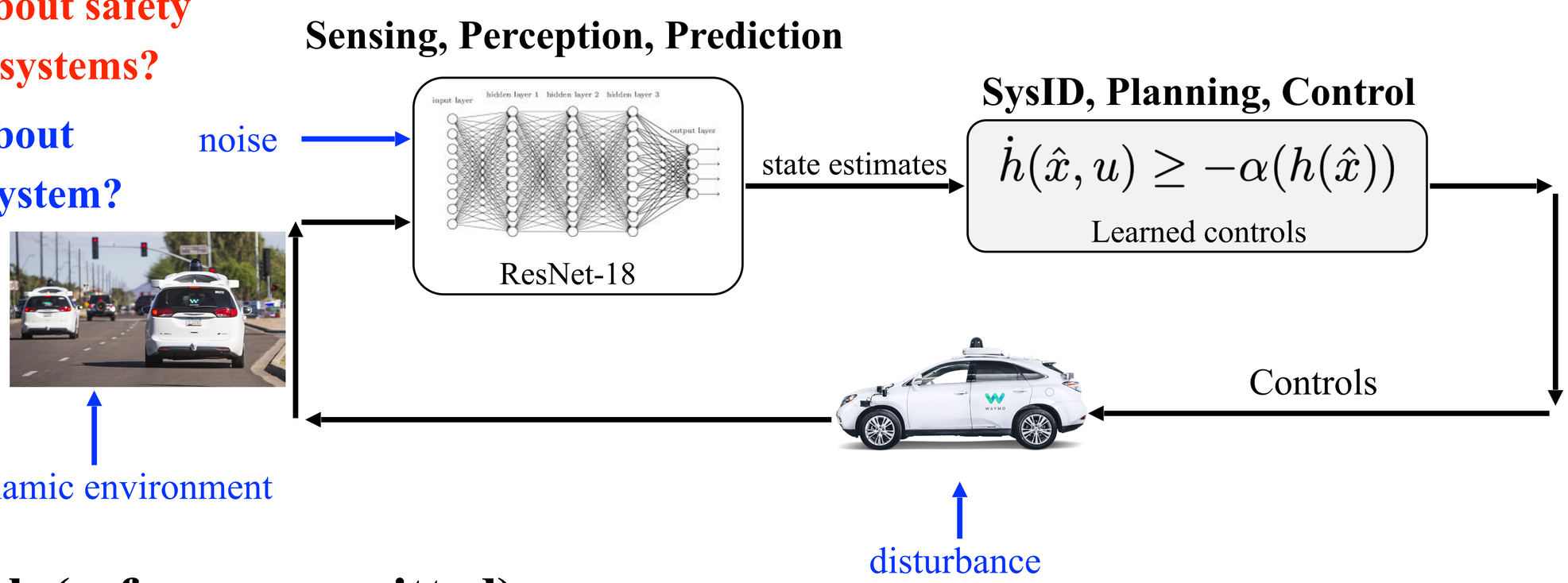


Modelling and control

Uncertainty quantification for learning-enabled CPS

How do we reason about safety of high-dimensional systems?

How do we reason about safety of stochastic system?



Existing work (references omitted):

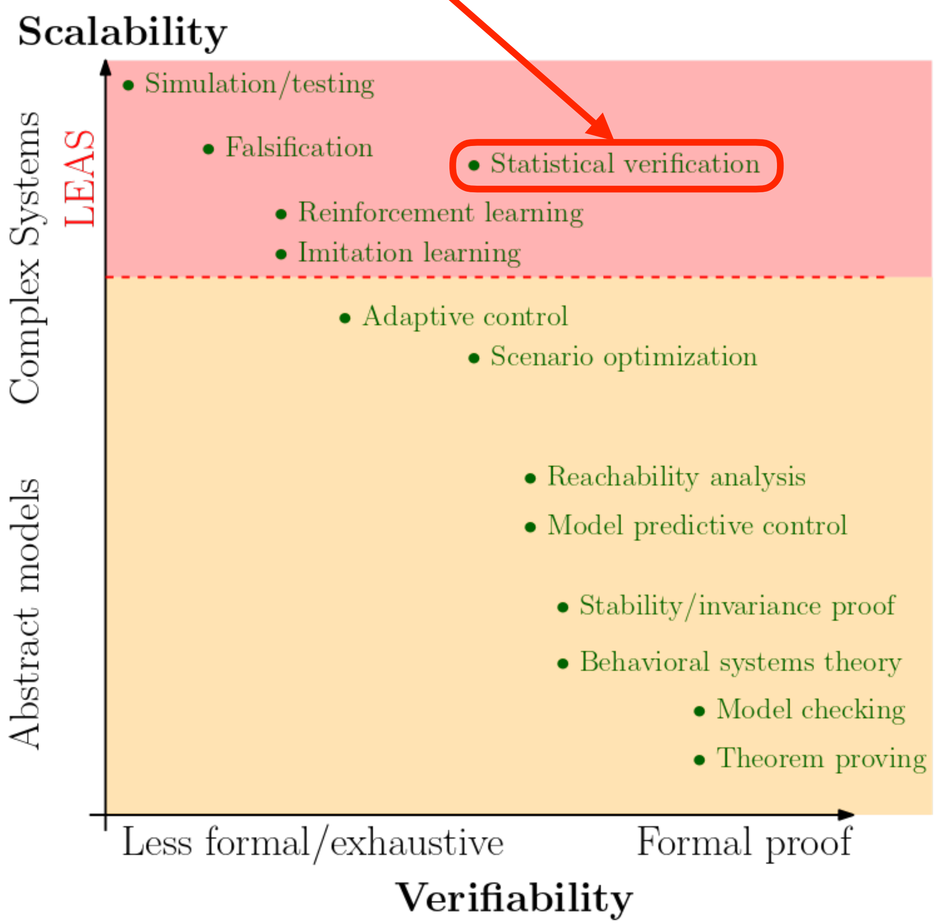
- SMT/MILP techniques or hybrid systems reachability ← **scalability issues**
- SDP/LP relaxations or compositional approaches ← **overly conservative**

Can we **use statistical tools** to formally reason about safety of learning-enabled systems?

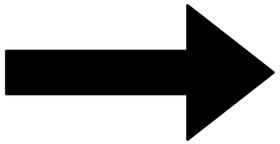
Conformal prediction for learning-enabled CPS

Conformal prediction: simple, general, and efficient

Goal: $\text{Prob}(x \models \phi) \geq 1 - \delta$



survey article



Formal Verification and Control with Conformal Prediction

PRACTICAL SAFETY GUARANTEES FOR AUTONOMOUS SYSTEMS

Authors: Lars Lindemann, Yiqi Zhao, Xinyi Yu, George J. Pappas, and Jyotirmoy V. Deshmukh

https://arxiv.org/pdf/2409.00536

Conformal Prediction

Conformal prediction in a nutshell

Uncertainty quantification under minimal assumptions:

- Distribution-free
- No assumptions on the predictor needed

no assumptions needed

$$(u, z) \sim \mathcal{D}$$

$$\hat{z} = \mu(u)$$

no assumptions needed

Assumption: Availability of i.i.d. calibration data $\{(u^{(i)}, z^{(i)})\}_{i=1}^{k}$

Goal[1]:

For a failure probability of $\delta \in (0, 1)$, we want to obtain a prediction region C s.t.

$$\text{Prob}(z \in C(u)) \geq 1 - \delta$$

Example[2]:

{ fox squirrel 0.99 }

{ fox squirrel 0.82, gray fox, 0.03, bucket, 0.02, rain barrel 0.02 }

{ marmot 0.30, fox squirrel 0.22, mink 0.18, weasel 0.16, beaver 0.03, polecat 0.01 }

[1] Vovk and Shafer, "A Tutorial on Conformal Prediction", *JMLR,* 2008.

[2] Angelopoulos, "A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification", *subm.,* 2021.

Conformal prediction in a nutshell

test data

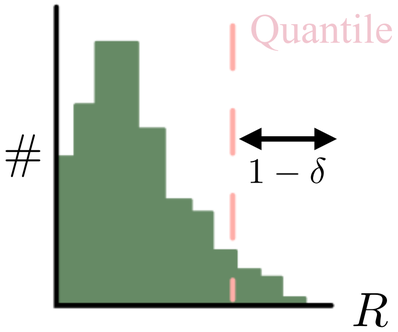
calibration data

nonconformity score (e.g., prediction error)

Quantile lemma:

Let $R^{(0)}, R^{(1)}, \ldots, R^{(k)}$ be $k+1$ i.i.d. random variables. It holds that

$$\text{Prob}\big(R^{(0)} \leq \text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(k)}, \infty)\big) \geq 1 - \delta$$

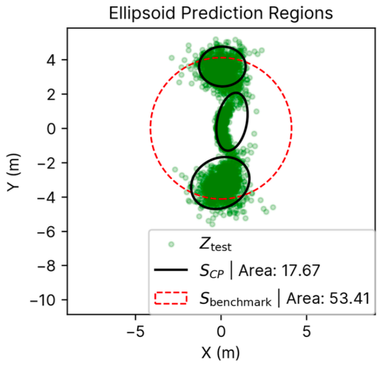


The $p := \lceil (k+1)(1-\delta) \rceil$ -th smallest nonconformity score ➡ computationally efficient

- Implicit **data requirement**: $p \leq k$
- For regression, the nonconformity score $R^{(i)} := |z^{(i)} - \mu(u^{(i)})|$ results in

$$\text{Prob}(z \in [\mu(u) - \text{Quantile}_{1-\delta}, \mu(u) + \text{Quantile}_{1-\delta}]) \geq 1 - \delta$$

- **Simple, general, and efficient** with the right nonconformity score[1]:



[1]Tumu, Cleaveland, Mangharam, Pappas, and **Lindemann**, "Multi-Modal Conformal Prediction Regions by Optimizing Convex Shape Templates", *L4DC,* 2024.

Marginal coverage guarantees

- How tight is the obtained bound?

Let $R^{(0)}, R^{(1)}, \ldots, R^{(k)}$ be $k+1$ i.i.d. **continuous** random variables. It holds that

$$\text{Prob}(R^{(0)} \leq \text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(k)})) \leq 1 - \delta + \frac{1}{k+1}$$

- Conformal prediction provides **marginal coverage** guarantees:

$$\text{Prob}\big(R^{(0)} \leq \text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(k)}, \infty)\big) \geq 1 - \delta$$

captures randomness in the draw over test and calibration data $R^{(0)}, R^{(1)}, \ldots, R^{(k)}$

- This is in contrast to **calibration conditional coverage** guarantees:

conditional probability

$$\text{Prob}\big(R^{(0)} \leq \text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(k)}, \infty) \,\big|\, R^{(1)}, \ldots, R^{(k)}\big) \geq 1 - \delta$$

captures randomness in the draw over test data $R^{(0)}$ → impossible to obtain

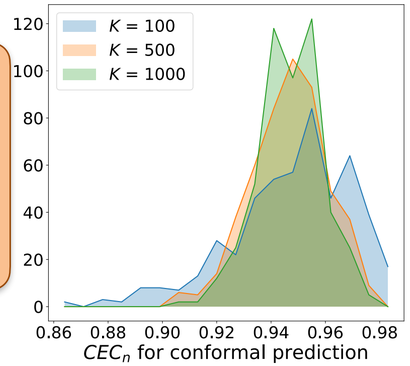
Calibration conditional conformal prediction

Let $R^{(0)}, R^{(1)}, \ldots, R^{(k)}$ be $k+1$ i.i.d. **continuous** random variables. It holds that[1]

$$\text{Prob}(R^{(0)} \leq \text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(k)}, \infty) | R^{(1)}, \ldots, R^{(k)}) \sim \text{Beta}(1-\delta, 1/k)$$

conditional probability

mean variance



- **(Pseudo) calibration conditional conformal prediction:**

Let $R^{(0)}, R^{(1)}, \ldots, R^{(k)}$ be $k+1$ i.i.d. random variables. It holds that[1]

$$\text{Prob}_K\left(\text{Prob}(R^{(0)} \leq \text{Quantile}_{1-\bar{\delta}}(R^{(1)}, \ldots, R^{(k)}, \infty) \geq 1-\delta\right) \geq 1-\beta$$

tightened quantile $\bar{\delta} = \delta - \sqrt{\frac{\ln(1/\beta)}{2k}}$

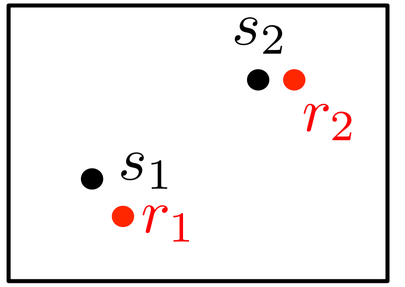
confidence $\beta \in (0,1)$

captures randomness in the draw over calibration data $R^{(1)}, \ldots, R^{(k)}$

[1]Vovk, "Conditional Validity of Inductive Conformal Predictors", *ACML*, 2012.

An academic example: sensor calibration

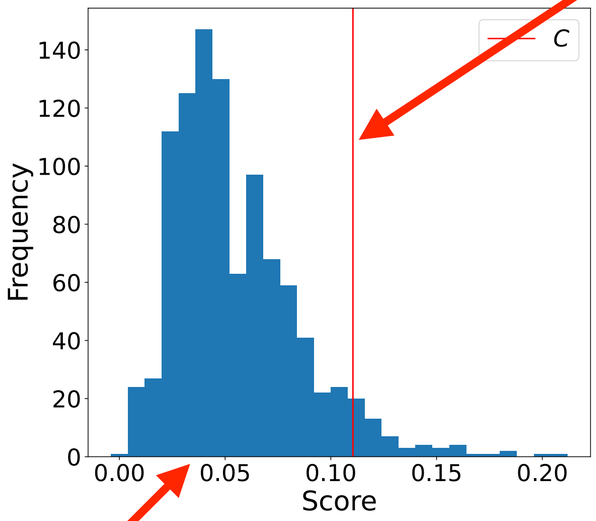
- Unknown regions: $r_1 \sim \mathcal{U}([1.5, 2.5] \times [0.5, 1])$ and $r_2 \sim \mathcal{U}([2.5, 3.5] \times [4, 4.5])$
- Sensor measurements: $s_1 \sim \mathcal{L}(r_1, 0.025)$ and $s_2 \sim \mathcal{L}(r_2, 0.025)$
- Sensor calibration: $\text{Prob}(\|s_l^{(0)} - r_l^{(0)}\| \leq \text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(k)}, \infty), \forall l \in \{1, 2\}) \geq 1 - \delta$



$R^{(i)} := \max_{l \in \{1,2\}} (\|s_l^{(i)} - r_l^{(i)}\|)$

- **Empirical validation:**

$\text{Quantile}_{0.95}$

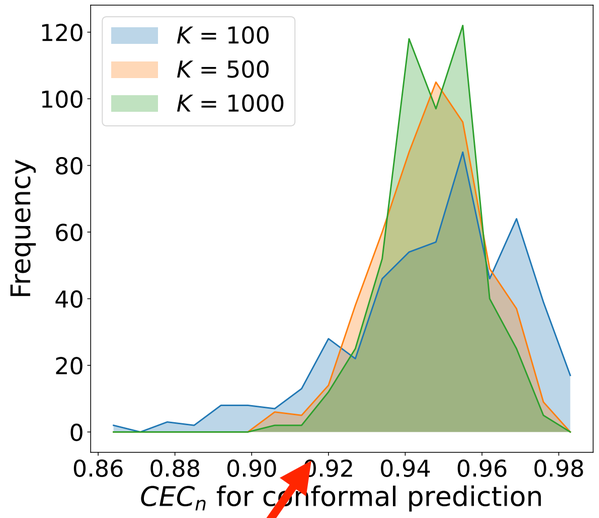


histogram $R^{(1)}, \ldots, R^{(k)}$

empirical coverage

$EC = \{0.963, 0.947, 0.957\}$

for $K \in \{100, 500, 1000\}$

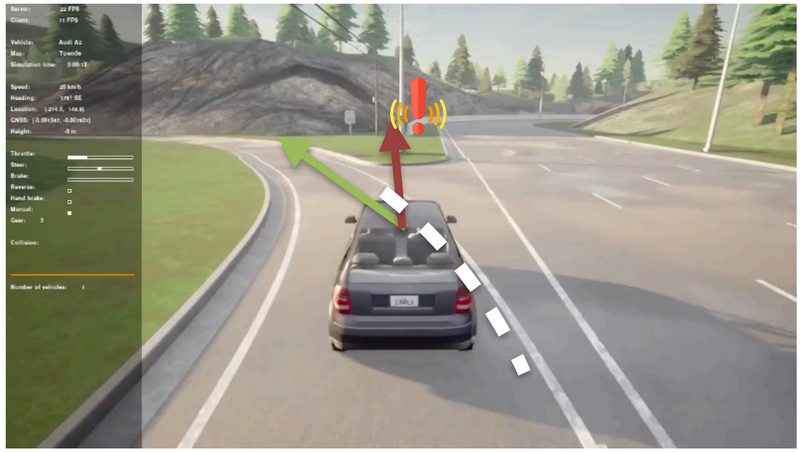


conditional empirical coverage

An academic example: sensor calibration

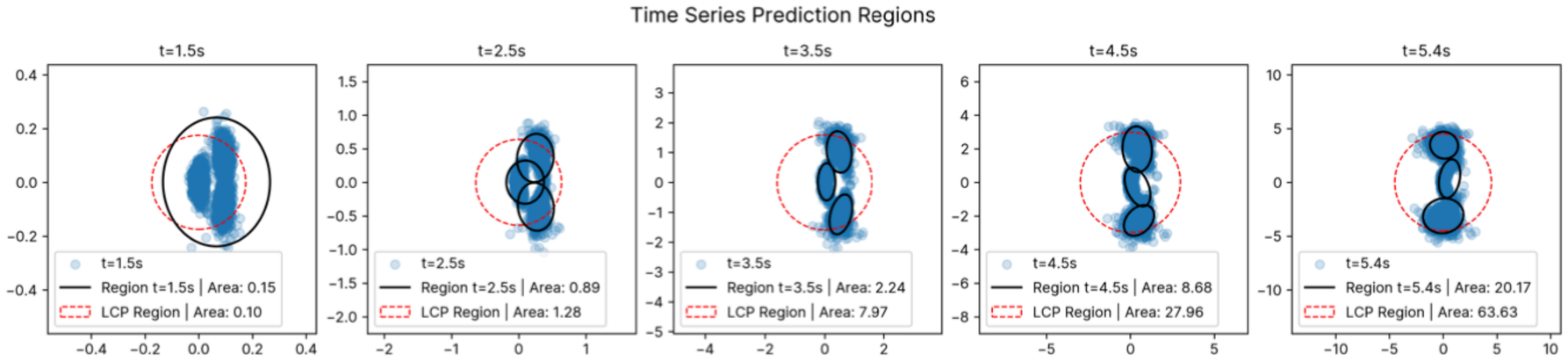
Code available at: https://github.com/zhaoy37/conformal_prediction_survey_codes

Remaining agenda: conformal prediction in autonomy



(1) Predictive Runtime Verification
(20 minutes)

(3) Safe Control in Dynamic Environments
(5 minutes)

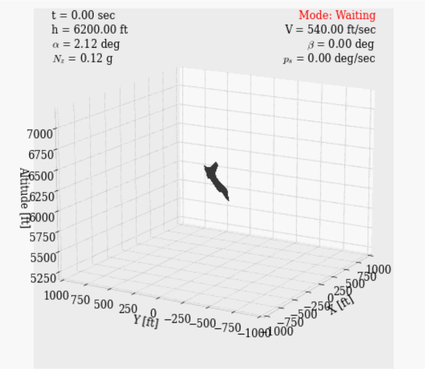


(2) Statistical Reachability Analysis
(20 minutes)

Predictive Runtime Verification

Predictive runtime verification

Motivation:



Stochastic systems

Learning-enabled systems

discrete-time stochastic systems

$$x = (x_0, x_1, \ldots) \sim \mathcal{D}$$

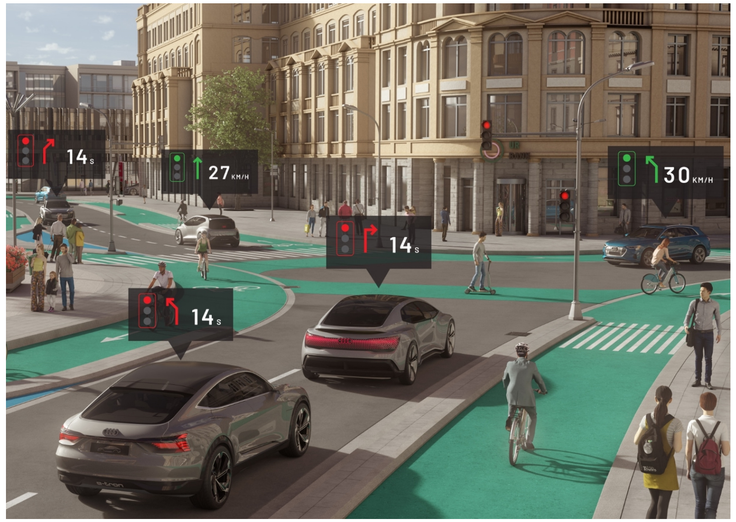
Predictive Runtime Verification:
Given observations $(x_0, \ldots, x_t)$, what is the probability that $\mathrm{Prob}(\rho^{\phi}(x) \geq 0)$?

performance measure

Challenges:

- How do we predict the behavior of **stochastic learning-enabled systems**?
- How can we use these to design **efficient runtime monitors**?
- How can we provide **probabilistic verification guarantees**?

Predictive runtime verification with conformal prediction

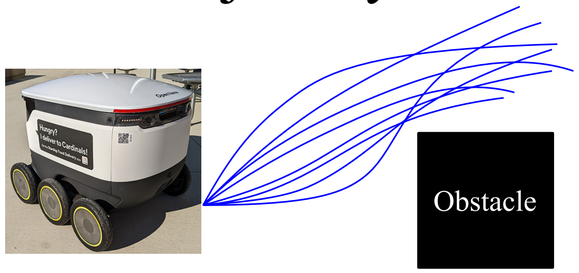


- **System dynamics:** stochastic, learning-enabled, and in general hard to model
- **Data-rich:** large datasets describing motion of stochastic CPS usually available
- **Predictors:** learning-enabled trajectory predictors (e.g., RNNs, LSTMs)

We use **conformal prediction** to quantify uncertainty of trajectory predictors and design **efficient predictive runtime monitors**.

Uncertainty-aware runtime verification

Offline trajectory dataset



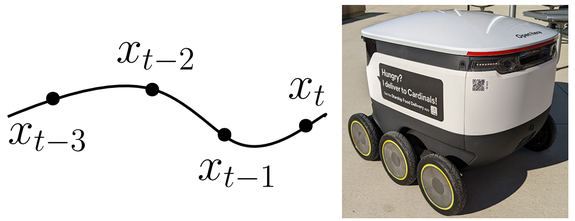
Conformal Prediction

Calibration step

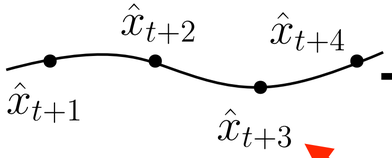
learned, e.g., RNN

$\hat{x} := (x_0, \ldots, x_t, \hat{x}_{t+1}, \ldots, \hat{x}_T)$

Online observations



Trajectory predictor



Performance measure $\rho^{\phi}(\hat{x})$

$\rho^{\phi}(x) \geq \rho^{\phi}(\hat{x}) - C$ with probability $1 - \delta$

How good are these estimates?

failure probability

Contributions:

- We propose two lightweight runtime verification algorithms
- We provide probabilistic correctness guarantees
- We perform a set of illustrative case studies

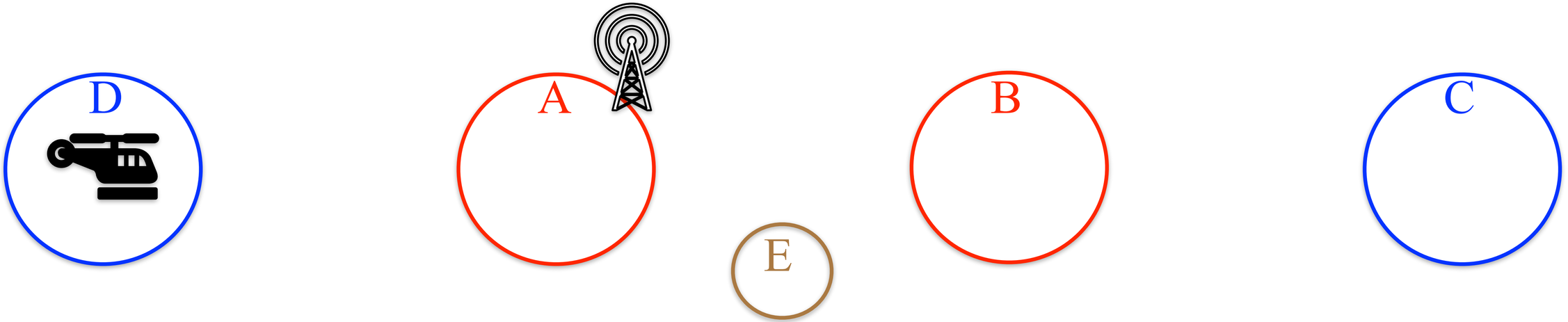
Lindemann, Qin, Deshmukh, and Pappas, "Conformal Prediction for STL Runtime Prediction", ICCPS, 2023.

Temporal logic specifications

A temporal logic formula ϕ has performance measure ρ^{ϕ}.

- Boolean logic equipped with temporal operators (eventually, always, until)

1. *Always* avoid E
2. *Every* hour inspect A *and* B
3. *If* battery level is low, *then* recharge *eventually* in C *or* D
4. *Always* between 8pm and 6am turn on communication



The accurate algorithm

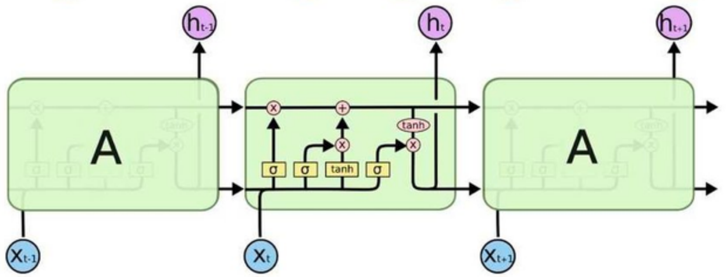
- Consider a trajectory predictor that maps $(x_0, \ldots, x_t)$ to $(\hat{x}_{t+1}, \ldots, \hat{x}_T)$

mission horizon

observations

predictions of $(x_{t+1}, \ldots, x_T)$

- For instance, long short-term memory (LSTM) networks



- Define the nonconformity score $R^{(i)} := \rho^{\phi}(\hat{x}^{(i)}) - \rho^{\phi}(x^{(i)})$ and apply conformal prediction $C := \text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(k)}, \infty)$

calibration data

$$\Rightarrow \text{Prob}(\rho^{\phi}(\hat{x}) - \rho^{\phi}(x) \leq C) \geq 1 - \delta$$

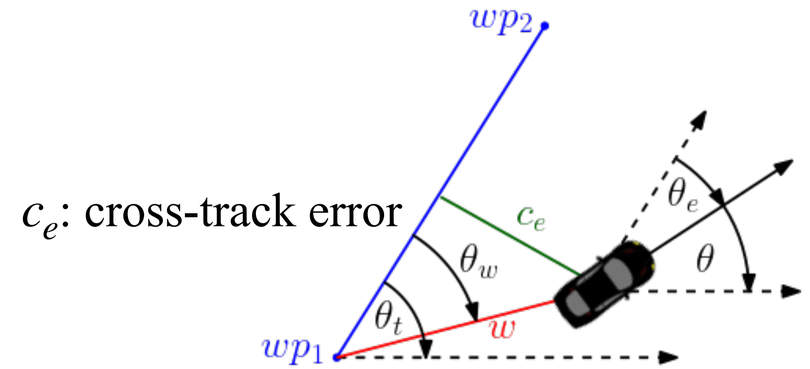
Given a finite-horizon specification ϕ, it holds that $\text{Prob}(\rho^{\phi}(x) \geq \rho^{\phi}(\hat{x}) - C) \geq 1 - \delta$

Lindemann, Qin, Deshmukh, and Pappas, "Conformal Prediction for STL Runtime Prediction", ICCPS, 2023.

CARLA: learning-enabled lane keeping controller

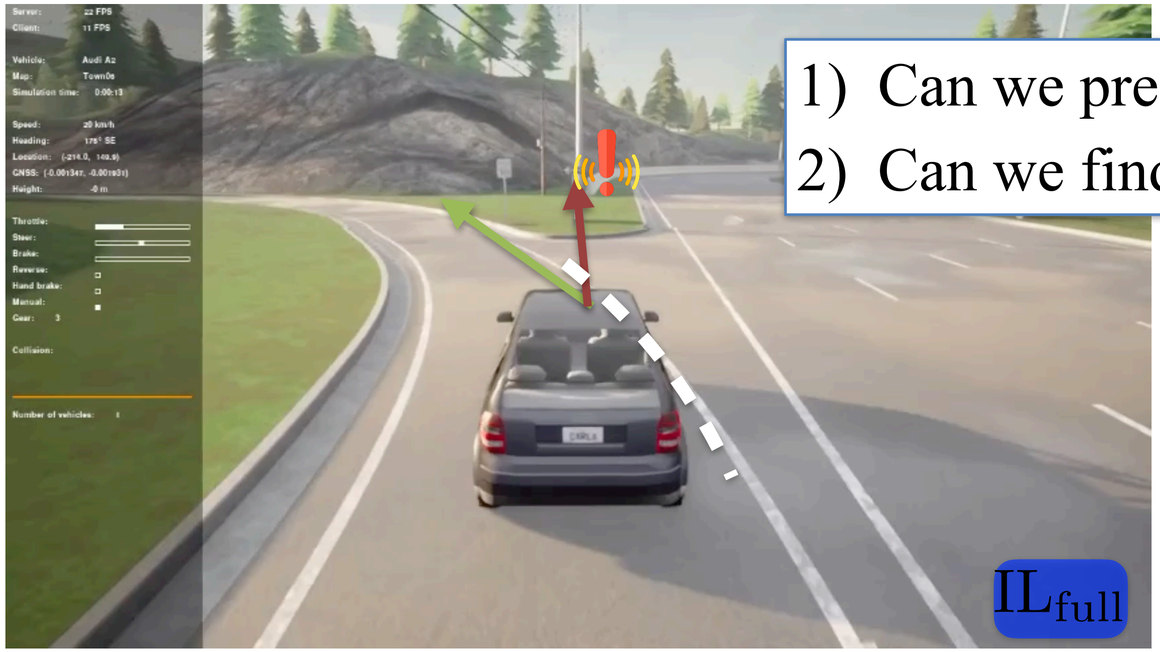
We use two different **learning-enabled lane keeping controllers**:

- Imitation learning[3] (IL_{full})
- Learned control barrier functions[4] (CBF_{full})



c_e: cross-track error

1) Can we predict unsafe behavior?
2) Can we find the safest controller?



[3]Ross and Bagnell, "Efficient reduction for imitation learning", *AISTATS*, 2010.

[4]Lindemann, Robey, Jiang, Tu, and Matni, "Learning Robust Output Control Barrier Functions from Expert Demonstrations", *OJ-CSYS*, 2024.

Accurate method: empirical evaluation

- Recall that we aim for: $\text{Prob}(\rho^{\phi}(x) \geq \rho^{\phi}(\hat{x}) - C) \geq 1 - \delta$ $\quad \delta := 0.05$

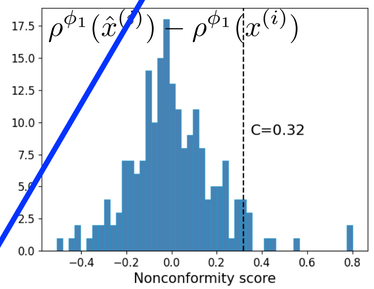
IL_{full} | Empirical coverage | CBF_{full}

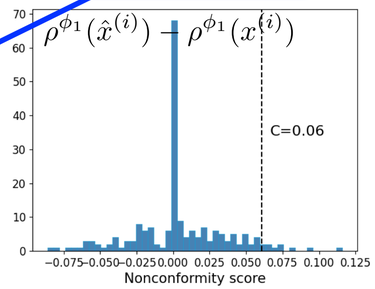
Two specifications:

1. Overall safety

   $G_{[10,\infty)} |c_e| \leq 2.25$

$C = 0.32$	95/100	$C = 0.06$	95/100

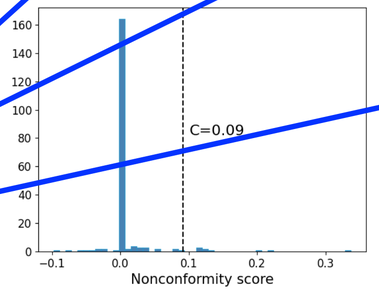


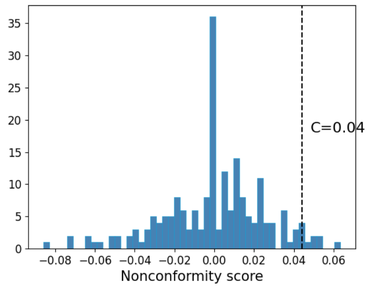


2. Controller reactivity

   $G_{[10,\infty)} (|c_e| \geq 1.25$
   $\implies F_{[0,5]} G_{[0,5]} |c_e| \leq 1.25)$

$C = 0.09$	98/100	$C = 0.04$	92/100





Measure of uncertainty and trust during runtime

The need for interpretability

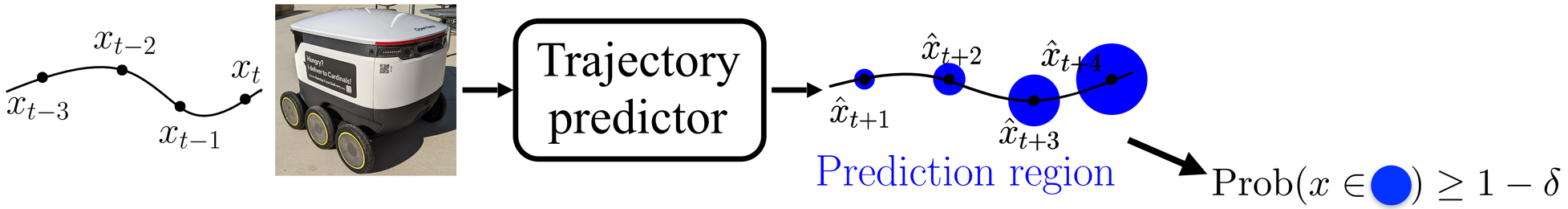
- Monitors can detect safety violations, but **fail to identify causes of violations**

Alert!!!

$\rho^{\phi}(\hat{x}) - C > 0$

Where and when is the car unsafe?

- **Idea**: obtain interpretability by quantifying uncertainty at the state prediction level



Uncertainty representation of state predictions

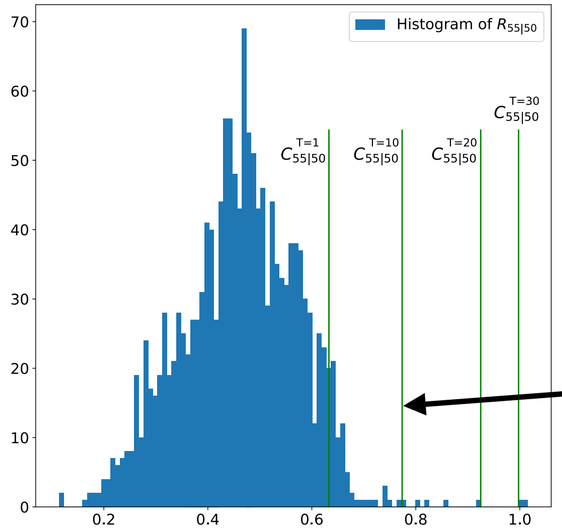
- Define the nonconformity score $R_\tau^{(i)} := \|x_\tau^{(i)} - \hat{x}_\tau^{(i)}\|$

  and apply conformal prediction $C_\tau := \text{Quantile}_{1-\delta/(T-t)}(R_\tau^{(1)}, \ldots, R_\tau^{(k)}, \infty)$

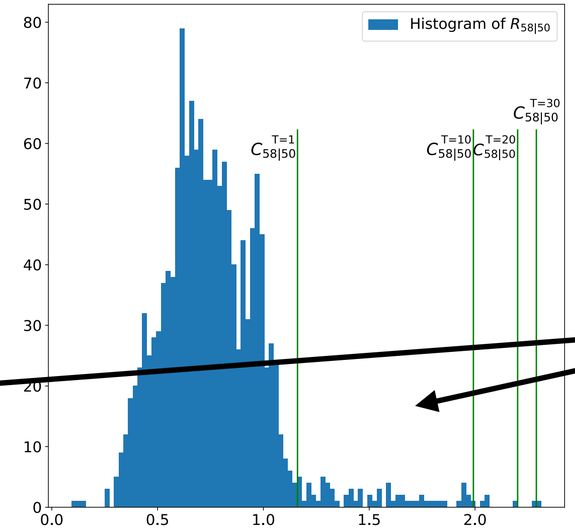
calibration data

Multi-step prediction regions: $\text{Prob}(\|x_\tau - \hat{x}_\tau\| \leq C_\tau,\ \forall \tau \in \{t+1, \ldots, T\}) \geq 1 - \delta$

Example: Intersection with pedestrians in CARLA



2.5 second ahead prediction error



4 second ahead prediction error

Size of prediction regions scales with T.

Lindemann, Cleaveland, Shim, and Pappas, "Safe Planning in Dynamic Environments using Conformal Prediction", *RA-L,* 2023.

Efficient uncertainty representations

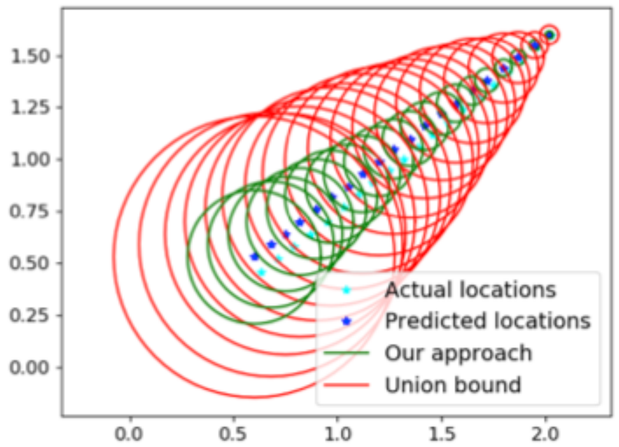
- Recall, we compute $C_\tau := \text{Quantile}_{1-\delta/(T-t)}(R_\tau^{(1)}, \ldots, R_\tau^{(k)}, \infty)$
- Conservatism can be avoided using the nonconformity score[1]

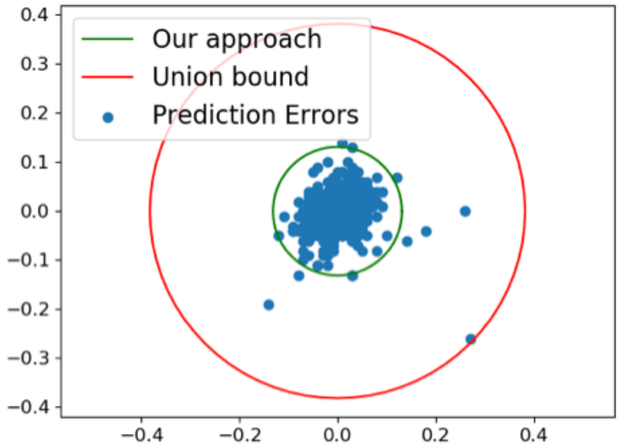
$$R^{(i)} := \max(\alpha_{t+1} R_{t+1}^{(i)}, \ldots, \alpha_T R_T^{(i)}) \quad \Rightarrow \quad \text{Prob}(\|x_\tau - \hat{x}_\tau\| \leq C/\alpha_\tau, \ \forall \tau \in \{t+1, \ldots, T\}) \geq 1 - \delta$$

normalization constant (next slide)

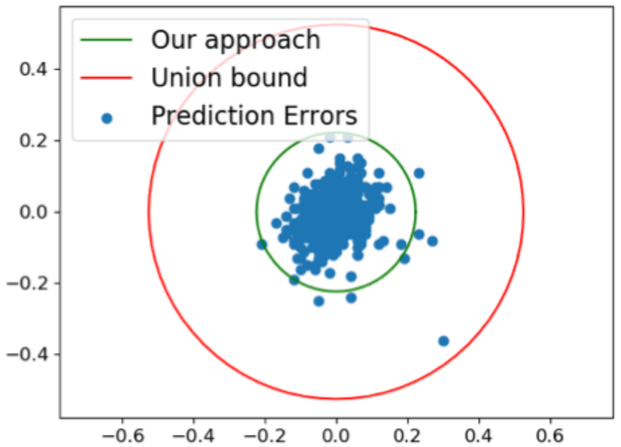
$$C = \text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(k)}, \infty)$$

Example:





1.25 second ahead prediction error



1.8 second ahead prediction error

[1]Cleaveland, Lee, Pappas, and **Lindemann**, "Conformal Prediction for Time Series using Linear Complementarity Programming", *AAAI*, 2024.

Efficient uncertainty representations

- Recall the nonconformity score $R^{(i)} := \max(\alpha_{t+1} R^{(i)}_{t+1}, \ldots, \alpha_T R^{(i)}_T)$

How to choose these optimally?

- Normalization constants $\alpha_1, \ldots, \alpha_T$ computed as the solution of

$$\begin{aligned} \min_{\alpha_1,\ldots,\alpha_T} \quad & \text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(k)}, \infty) \\ \text{s.t.} \quad & R^{(i)} = \max(\alpha_1 R^{(i)}_1, \ldots, \alpha_T R^{(i)}_T) \\ & \sum_{j=1}^{T} \alpha_j = 1 \\ & \alpha_j \geq 0 \end{aligned}$$

minimize quantile

nonconformity score

a linear complementarity program

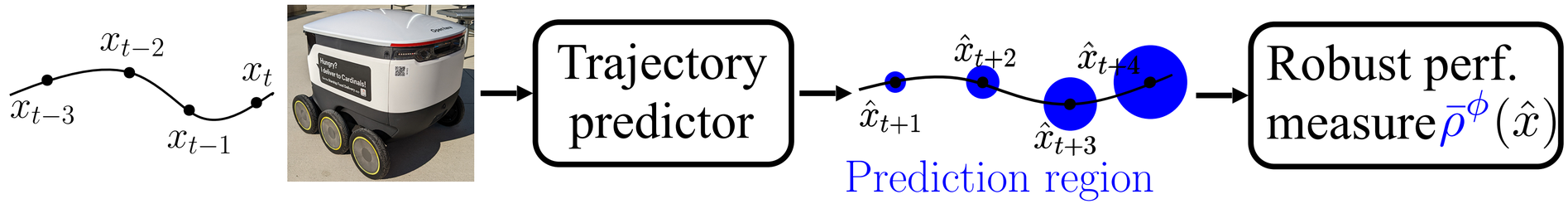
a mixed integer linear program

"normalization budget"

This optimization problem is a **mixed integer linear complementarity program**.

Cleaveland, Lee, Pappas, and **Lindemann**, "Conformal Prediction for Time Series using Linear Complementarity Programming", *AAAI*, 2024.

The interpretable algorithm



Compute worst case performance: $\bar{\rho}^{\phi}(\hat{x}) := \sup_{x_{t+1} \in \mathcal{B}_{t+1}, \ldots, x_T \in \mathcal{B}_T} \rho^{\phi}(x)$

$\mathcal{B}_\tau := \{x \text{ s.t. } \|x - \hat{x}_\tau\| \leq C_\tau\}$

Given a finite-horizon specification ϕ, it holds that $\text{Prob}(\rho^{\phi}(x) \geq \bar{\rho}^{\phi}(\hat{x})) \geq 1 - \delta$

➡ we can compute $\bar{\rho}^{\phi}(\hat{x})$ efficiently

Predicates:

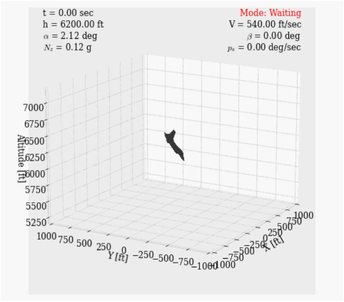
$$\bar{\rho}^{\mu}(\hat{x}, \tau) := \begin{cases} h(x_\tau) & \text{if } \tau \leq t \\ \inf_{\zeta \in \mathcal{B}_\tau} h(\zeta) & \text{otherwise} \end{cases}$$

x_τ already observed

x_τ not yet observed

Boolean/temporal operators: Defined in their standard way using nested min/max

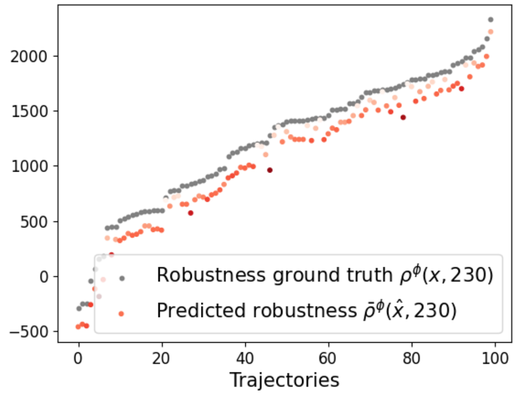
Lindemann, Qin, Deshmukh, and Pappas, "Conformal Prediction for STL Runtime Prediction", ICCPS, 2023.



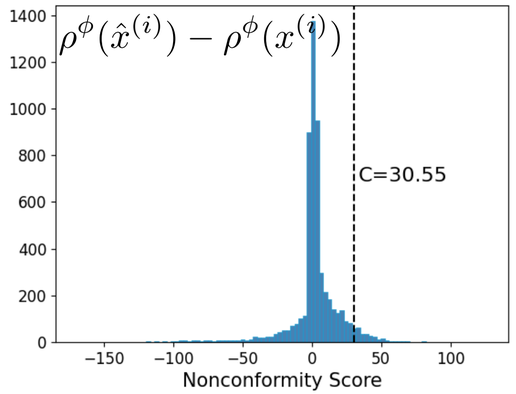
F-16 case study: accurate vs interpretable method

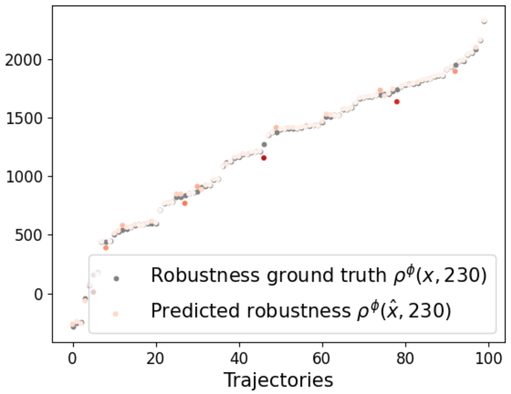
→ High-fidelity F-16 simulator with ground collision avoidance[5]

Interpr. method



Accurate method





- We aim for: $\text{Prob}(\rho^\phi(X) \geq \bar{\rho}^\phi(\hat{x})) \geq 1 - \delta$

→ Empirically: 100/100

- We aim for: $\text{Prob}(\rho^\phi(X) \geq \rho^\phi(\hat{x}) - C) \geq 1 - \delta$

→ Empirically: 96/100

Indirect: (-) Conservative
(-) May require lots of data
(+) Interpretable

Direct: (+) Accurate
(+) Data-efficient
(-) Not interpretable

Both: (+) No retraining when specifications change
(+) Easy to implement and understand

[5]Heidlauf, Collins, Bolender, and Bak, "Verification challenges in F-16 ground collisions avoidance and other automated maneuvers", 2018.

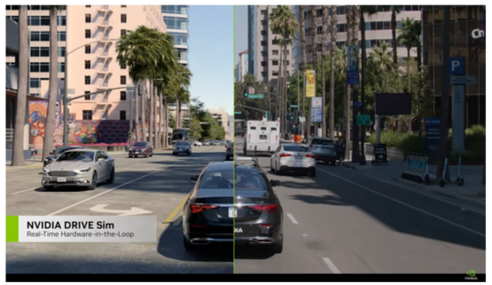
Distribution Shifts

Safe control when the distribution shifts

- **Design and deployment conditions** may be different, i.e., the distribution may "shift"

Examples: changing environments

data from simulators



← no safety guarantees

Assumption: design and deployment conditions are ϵ-close in **f-divergence**

- **Robust conformal prediction[1]:** Let $R^{(1)}, \ldots, R^{(k)} \sim \mathscr{R}$ and $R^{(0)} \sim \mathscr{R}_0$ as well as $D_f(\mathscr{R}_0, \mathscr{R}) \leq \epsilon$.

It holds that $\text{Prob}(R^{(0)} \leq \tilde{C}) \geq 1 - \delta$ with $\tilde{C} := \text{Quantile}_{1-\tilde{\delta}(\epsilon)}(R^{(1)}, \ldots, R^{(k)})$.

adjusted confidence level $\tilde{\delta}(\epsilon) < \delta$
(solution of convex optimization problem)

→ Increased data requirements

[1]Cauchois, Gupta, Ali, and Duchi, "Robust validation: Confident predictions even when distributions shift", JASA, 2024.

Robust predictive runtime verification

- Bounding the distribution shift:

follows by data processing inequality

nonconformity score R

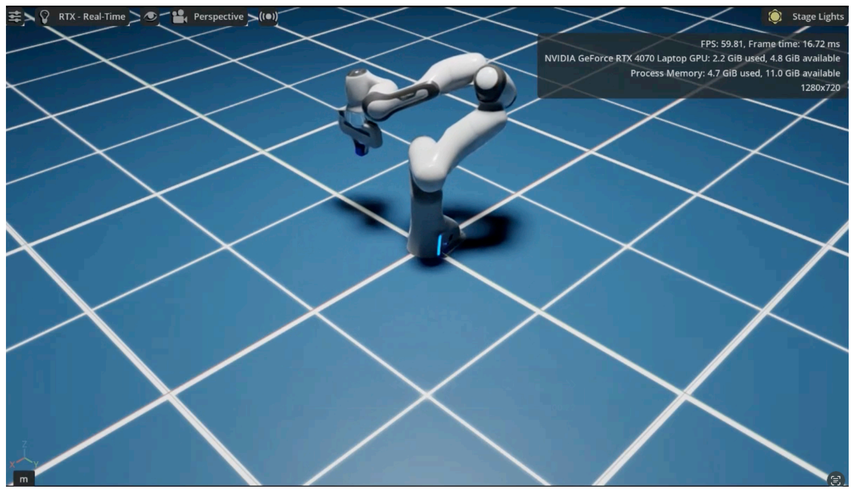
$$D_f(\mathcal{D}_0, \mathcal{D}) \leq \epsilon \quad \longrightarrow \quad D_f(\mathcal{R}_0, \mathcal{R}) \leq \epsilon$$

test trajectory x_{real}

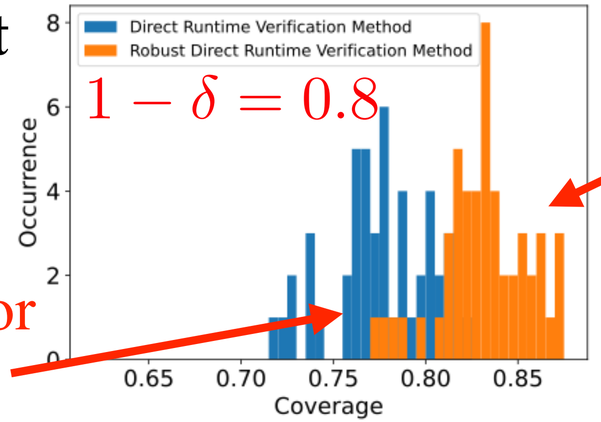
calibration trajectories $x^{(1)}, \ldots, x^{(k)}$

$R(x_{\text{real}})$ $R(x^{(1)}), \ldots, R(x^{(k)}),$

- Enables the design of **robust runtime verification algorithms**[2,3]:



Direct



robust monitor achieves coverage

vanilla monitor undercovers

Best Paper Award Finalist

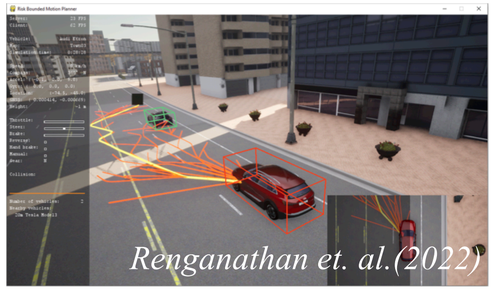
[2] Zhao, Hoxha, Fainekos, Deshmukh, and **Lindemann**, "Robust Conformal Prediction for STL Runtime Verification under Distribution Shift", ICCPS, 2024.

[3] Zhao, Zhu, Hoxha, Fainekos, Deshmukh, and **Lindemann**, "Distributionally Robust Predictive Runtime Verification under Spatio-Temporal Logic Specifications", TCPS (subm), 2025.

Safe Control in Dynamic Environments

The safe control problem in dynamic environments

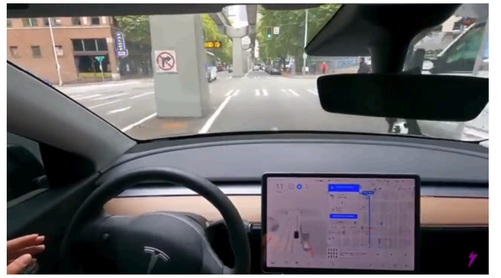
Compute **control inputs** so that the system avoids **dynamic agents** with a probability of at least $1 - \delta$.



Challenges:

- Stochastic and dynamic agents
- Complex learning-enabled predictors

uncertainty quantification within control difficult



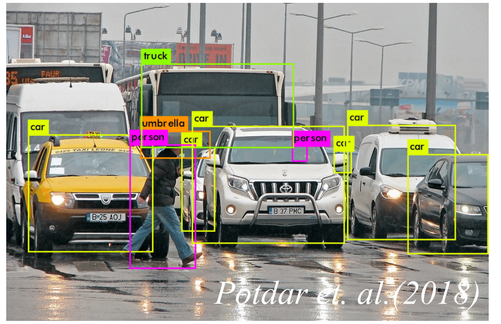
Existing work (references omitted):

- "Vanilla" motion planner (e.g., graph/sampling-based)
- Reactive control (e.g., navigation functions, DWA, CBF)
- Predictive interactive and non-interactive methods (e.g., deep RL, GP-based control)

ignores dynamic agents

no predictions, simple systems

no safety guarantees, or simplifying assumptions

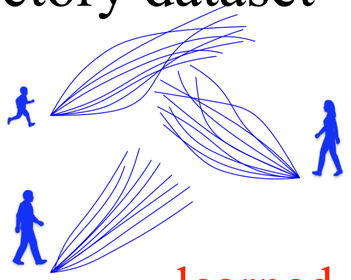


How can we quantify uncertainty of **learning-enabled predictors** and design **safe control laws**?

Safe Control in Dynamic Environments

Safe control:
Design controllers that avoid dynamic agents with a probability of at least $1-\delta$.

Offline trajectory dataset



an efficient statistical tool

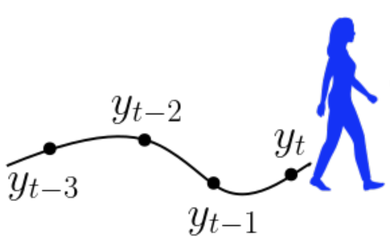
Conformal Prediction

Dynamics

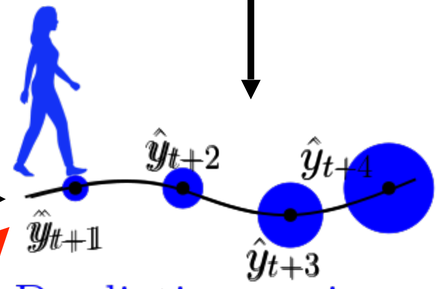
$x_{t+1} = f(x_t, u_t)$

learned, e.g., RNN

Online observations



Trajectory predictor



Prediction region

$\text{Prob}(y \in \bigcirc) \geq 1-\delta$

How good are these estimates?

safety constraint

Model Predictive Control

$Prob(c(x,y) \geq 0) \geq 1-\delta$

Contributions:

- Computationally lightweight algorithm
- Probabilistic safety guarantees
- Deals with distribution shifts

Case Study: CARLA

Results:

- 99.985 % correct one-step ahead predictions (at least 99.83 % expected)
- 99/100 test runs without constraint violation (at most 5 expected)

Comparison with Gaussian Process:

- Largely undercovers

Lindemann, Cleaveland, Shim, and Pappas, “Safe Planning in Dynamic Environments using Conformal Prediction”, *RA-L,* 2023.

Part Two:

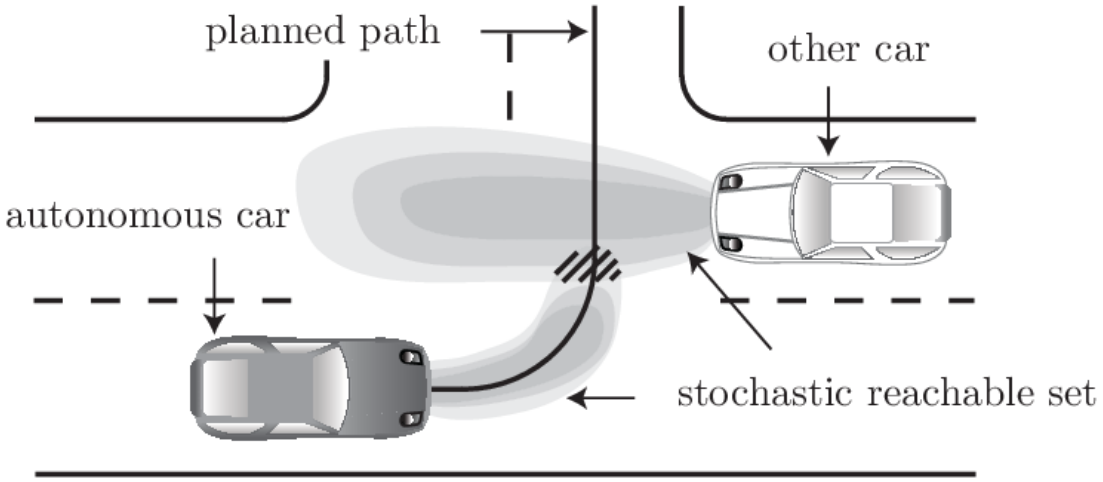
Model-free Methods for Verification using Conformal Prediction

Jyo Deshmukh,

CS, ECE @ School of Advanced Computing,
Viterbi School of Engineering
University of Southern California (USC)

Why should we care about reachability analysis?

- Computationally promising method for safety proofs of dynamical systems
- Active area of research in hybrid systems/formal methods community
  - Many tools such as SpaceEx, C2E2, DryVr, Cora, Flow*, ReachNN, Polar, Sherlock, …
- Useful tool to understand model behavior (*simulation on steroids*):
  - not just simulated system behavior, but behavior in an uncertain neighborhood



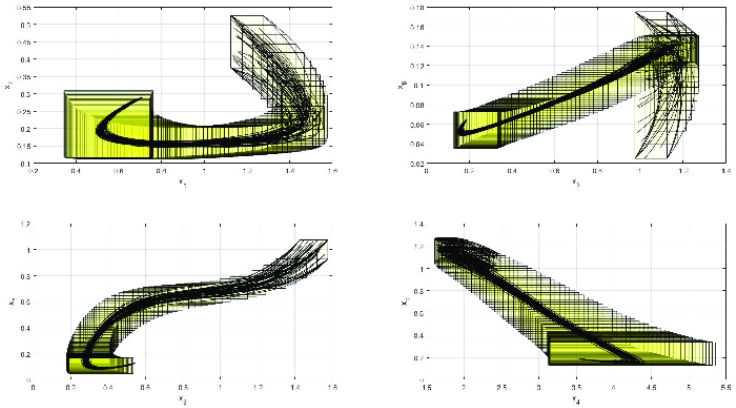


Image credit: M. Althoff, Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars

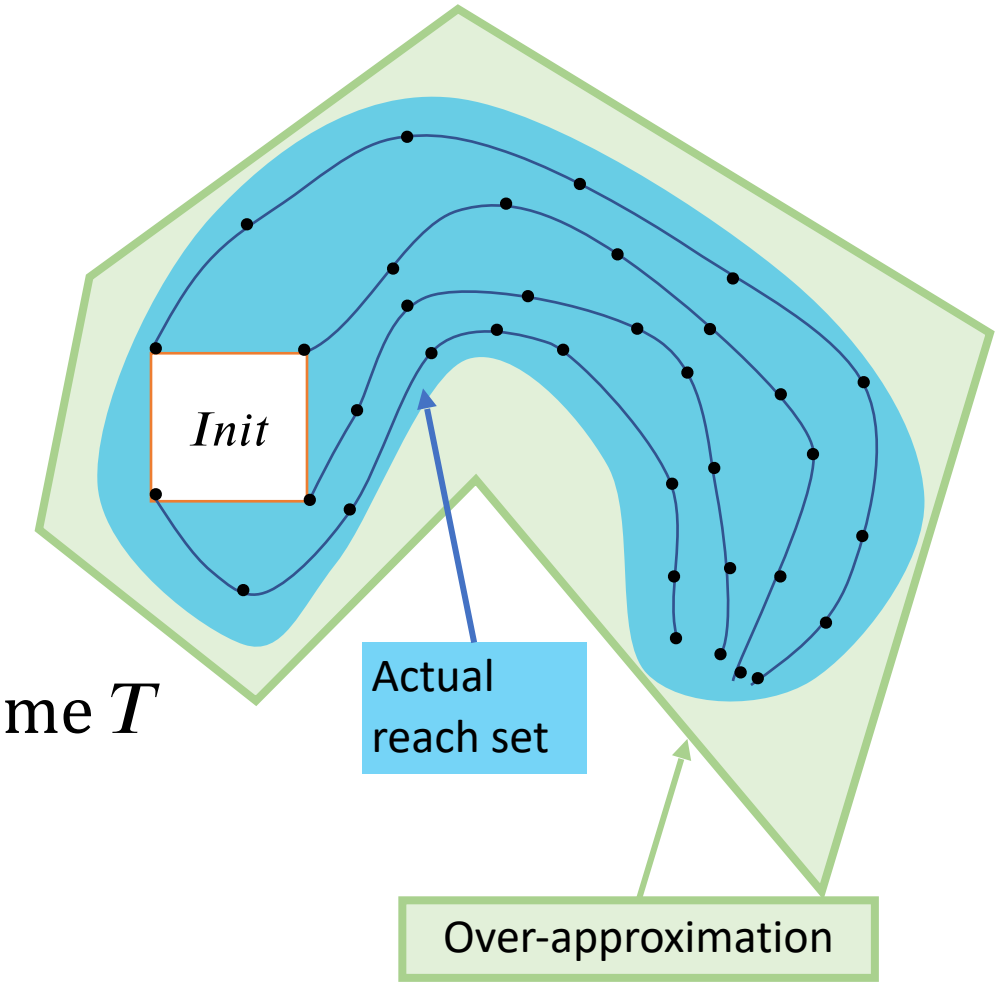
Reachability Analysis

Given

- set of initial states $Init$
- system dynamics: $s_{t+1} = f(s_t)$
- time horizon T

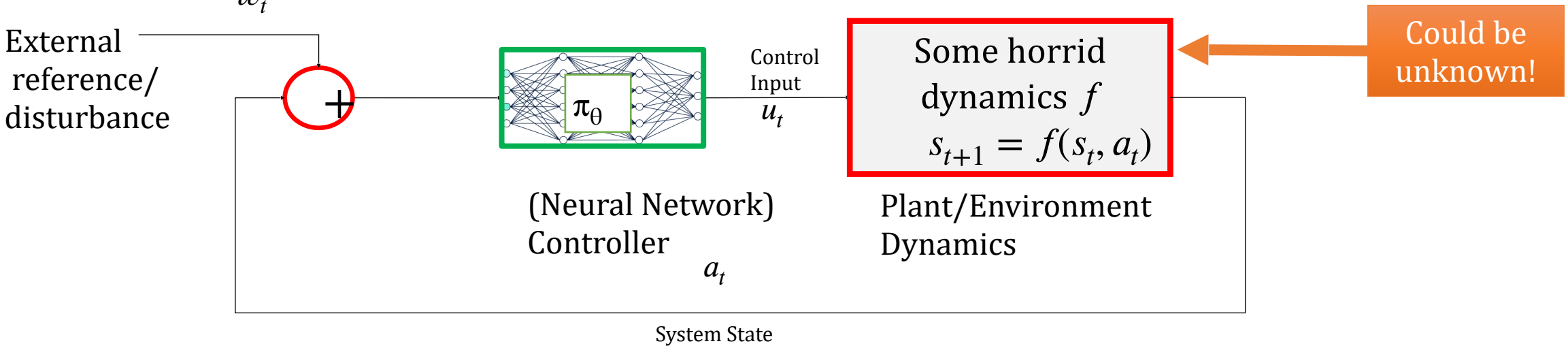
Compute

- over-approximation of reachable states over time T
- compute $Reach_T(Init)$
  - $\forall t \in [0,T], \forall s(0) \in Init,\ s(t) \in Reach_T(Init)$
  - where: $\forall t \in [1,T]\, s_t = f(s_{t-1})$



Reachability for deterministic dynamical systems

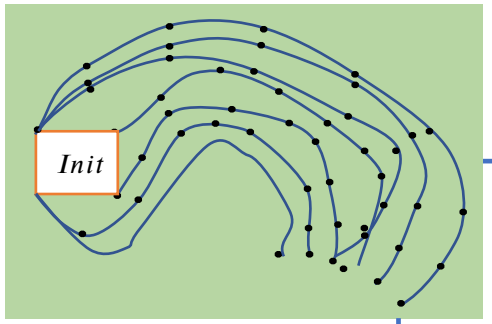
- Rich literature in the hybrid systems community
  - Represent set of reachable states with a convenient geometric shape
  - Algorithms to propagate geometric shapes through dynamics and guards
- What do we do when the controller is a neural network?
  - Tools like Sherlock, Verisig, ReachNN, Polar, Alpha-Beta-Crown, etc.



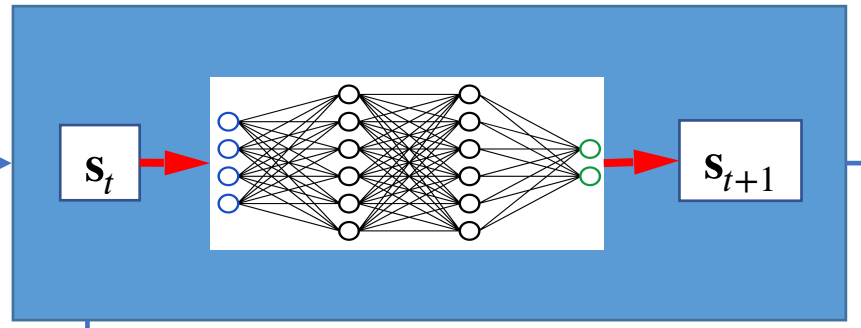
- What do you do when you do not know the dynamics?

Data-driven reachability[1]: High-level idea

1. Hashemi, Lindemann, Deshmukh, Data-Driven Reachability Analysis of Stochastic Dynamical Systems with Conformal Inference, Proc. of CDC 2023



Learn surrogate model



Set Propagation through Time-unrolled NN dynamics (NN reachability)

Conformal Prediction to calculate probabilistic bounds on prediction error of NN dynamics

R_*

$\oplus$

Over-approximated flowpipe or reachset of the time-unrolled NN

Probabilistic Reach Set

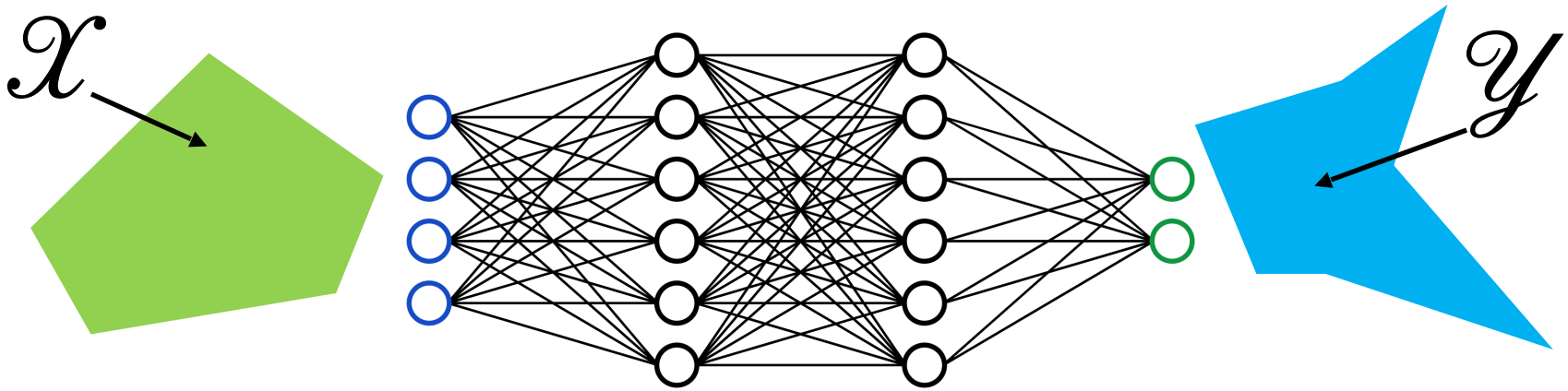
Training the neural surrogate model

- Sample initial state $s_0 \in Init$
- Obtain trajectory $\sigma_{s_0} = s_0,\ s_1, , \ldots s_{T-1},\ s_T;\ D = \{\langle s_t,\ s_{t+1}\rangle\}$ for all t
- Define NN $\hat{f}(s;\theta)$: input s_t and output s_{t+1}; parameters: θ
- Train NN to minimize standard MSE loss function

$$\theta^* = \text{argmin}\ \frac{1}{|D|} \sum_{(\mathbf{s}_t, \mathbf{s}_{t+1}) \in D} \left\| \mathbf{s}_{t+1} - \hat{f}(\mathbf{s}_t;\theta) \right\|_2$$

- Training itself: standard back-propagation
- Can also train to minimize error for the entire T-step prediction, i.e.,
  - $D = \left\{ \langle s_0,\ (s_1, \ldots,\ s_T) \rangle_i \right\}$ [Avoids error propagation of the one-step predictions]

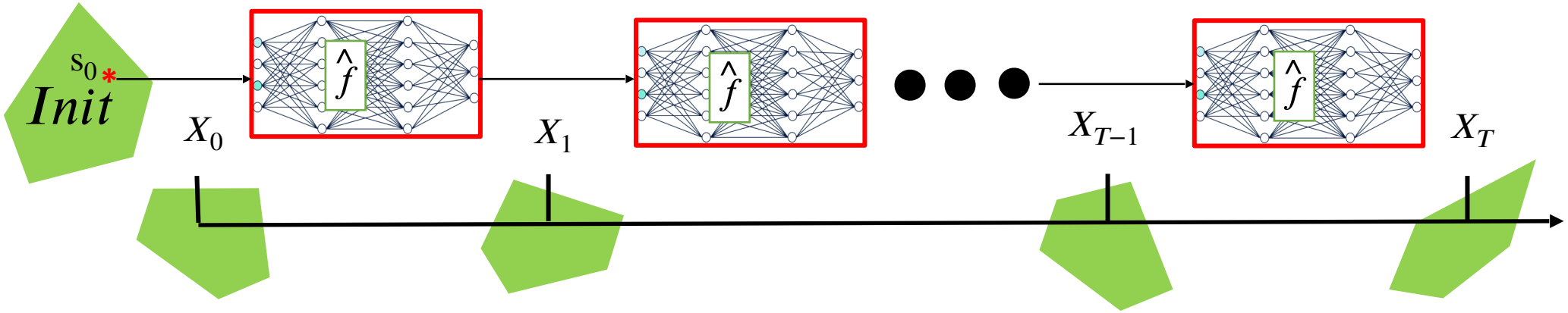
NN reachability (forward image computation)



- Many tools: NNV[1], alpha-beta-CROWN[2], Marabou[3], …
  - VNN-comp: several tools that compete on reachability benchmarks
- NNV :
  - star sets (efficient representation of polyhedral sets)
  - layer-by-layer forward propagation of sets
  - has both an exact (slow but accurate) and an abstract (fast but inaccurate) method
  - Navid Hashemi and Dung Tran were interns at Toyota R&D at the same time ☺

[1] D. Tran, et al. "NNV: the neural network verification tool for deep neural networks and learning-enabled cyber-physical systems." *CAV 2020*
[2] S. Wang, et al., " Beta-CROWN: Efficient bound propagation with per-neuron split constraints for complete and incomplete neural network verification, Neurips 2021
[3] Katz, Guy, et al. "The marabou framework for verification and analysis of deep neural networks, CAV 2019.

NN reachability to compute reach-set



- Reach-set $= X_0 \cup X_1 \cup \ldots \cup X_T$
- NN reachability guarantee:
  - for every trajectory $\hat{\sigma}_{\mathbf{s}_0} = \mathbf{s}_0, \hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_T$ s.t. $\hat{\mathbf{s}}_{t+1} = \hat{f}(\mathbf{s}_t)$
  - $\hat{\sigma}_{\mathbf{s}_0} \in$ Flow-pipe X

NN model is not the actual system: use Conformal Prediction

Conformal prediction[1,2]:

- By this time, you are experts in CP

1. Shafer, G., & Vovk, V. (2008). A tutorial on conformal prediction. Journal of Machine Learning Research, 9(3).
2. Qin, X., Xia, Y., Zutshi, A., Fan, C., & Deshmukh, J. V. Statistical verification of cyber-physical systems using surrogate models and conformal inference. Proc. of ICCPS 2022

Conformal prediction to compute $(1-\delta)$-confidence flowpipes

- Collect m trajectories $\sigma_{\mathbf{s}_{0,1}}, \sigma_{\mathbf{s}_{0,2}}, \ldots, \sigma_{\mathbf{s}_{0,m}}$
- R_i^j : prediction error for the j^{th}component in the i^{th} trajectory
  - $R_i^j = \left\| \sigma_{s_0,i}(j) - \hat{\sigma}_{s_0,i}(j) \right\|$ *component = dimension + time-step*
- Define $R_i = \max_j \alpha_i R_i^j$ as the nonconformity score
  - max normalized error over any component for i^{th} trajectory
  - normalizing with α_i's helps take care of unequal error scales across components
- $D_{cal} = \{R_1, \ldots, R_m\}$, compute $(1-\delta)$-quantile of $D_{cal} = R_*$
- Let X be the flow-pipe computed by NNV
- $X^* = X \oplus \left[\frac{R_*}{\alpha_1}, \ldots, \frac{R_*}{\alpha_{nT}}\right]$ is a $(1-\delta)$-confidence flow-pipe,
- i.e., $P\left(\sigma_{s_0} \in X^*\right) > 1-\delta$

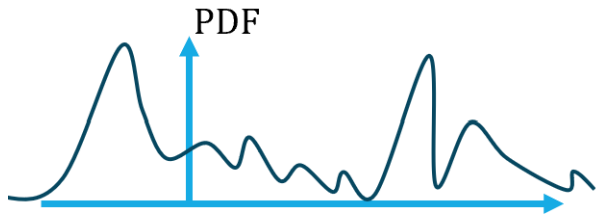
What's under the rug?

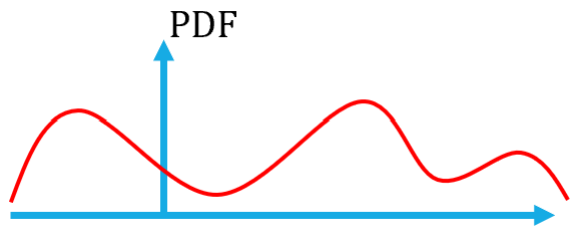
- Assumes that the trajectory distribution remains the same
  - Big assumption: *distributions* often *shift* between the lab and the real-world
- How to reason about the distribution shift:
  - In robust control, amount of noise is an input parameter,
  - Distribution shift can be an input parameter! [a burden on the designer]
- But can we estimate it?
  - Compute kernel density approximation using data
  - Compute your favorite divergence metric on the empirical PDFs
  - Approximate the distribution shift

1. Hashemi, Lindemann, Deshmukh, Data-Driven Reachability Analysis of Stochastic Dynamical Systems with Conformal Inference, Proc. of CDC 2023

Robust Conformal Prediction[1]

By this time, you are also experts in robust CP

$TV($ 

, 

$) = \tau$

- If total variation distance between $\mathscr{R}_{\text{training}}$ and $\mathscr{R}_{\text{deploy}}$ is τ, then:

$$\ell = \left\lceil \left(1 + \frac{1}{m}\right)(1 - \delta + \tau)(m + 1) \right\rceil$$

1. Cauchois, Maxime, Suyash Gupta, Alnur Ali, and John C. Duchi. "Robust validation: Confident predictions even when distributions shift."

Training Better Surrogate Models

- NN surrogate challenges:
  - How do we decide structure of the NN?
  - How do we limit the undesirable sensitivity of NNs?
  - How do we really train it for *less conservatism* in CP-based reach-set inflation*?*
- Structural constraints: Bound the Lipschitz constant of the NN
- What loss to use? [MSE has issues!]
  - Many errors are very low, but one very high error ⇒ mean value **is** low, but the quantile is bad!
  - To reduce conservatism when inflating reach sets: $(1-\delta)$-quantile of error should be small
  - Then why not just use $(1-\delta)$-quantile as the objective function?
  - We can …

Quantile loss

- Idea inspired by work in quantile regression (QR)
- Define $\mathscr{L} = c\mathscr{L}_1 + \mathscr{L}_2$: c is some large number
- $\mathscr{L}_1$ sets up a trainable parameter[1] q

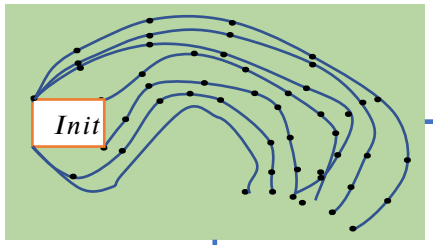
$$\mathscr{L}_1 = \sum_{i=1}^{m} \underbrace{(1-\delta)\cdot \mathrm{RELU}(R_i - q)}_{R_i \text{ if } R_i > q,\quad 0 \text{ otherwise}} + \underbrace{\delta \cdot \mathrm{RELU}\left(q - R_i\right)}_{q \text{ if } q > R_i,\quad 0 \text{ otherwise}}$$

  - QR result: minimizing $\mathscr{L}_1$ makes q the $(1-\delta)$-quantile of $R_1, \ldots, R_{nT}$
- $\mathscr{L}_2$ sets up minimizing q [Technically the $(1-\delta)$-quantile of the *sum* of all component-wise errors]

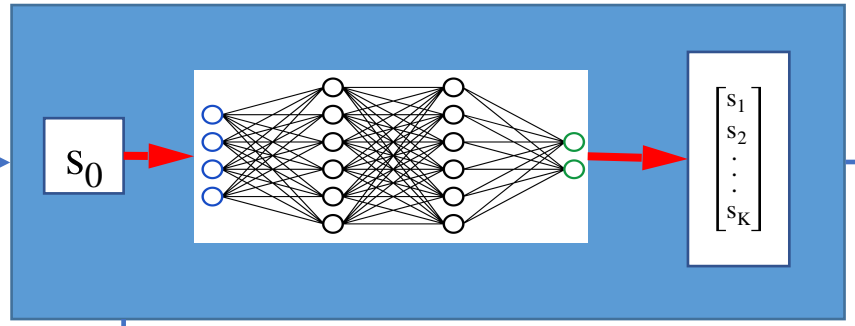
$$\mathscr{L}_2 = q\left(\frac{1}{\alpha_1} + \ldots + \frac{1}{\alpha_{nT}}\right)$$

[1] In the paper, this formula looks different. In this tutorial, to be uniform, we call the mis-coverage level $1-\delta$, while in the paper it is called $\bar{\delta}$

Solution: Robust Model-free Reachability Analysis[1]



Learn surrogate model



Challenges:

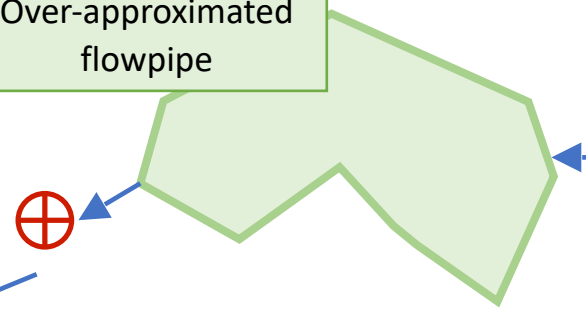
- Training the model well
  - Can use quantile loss
- Reducing conservatism
  - Please read the papers ☺

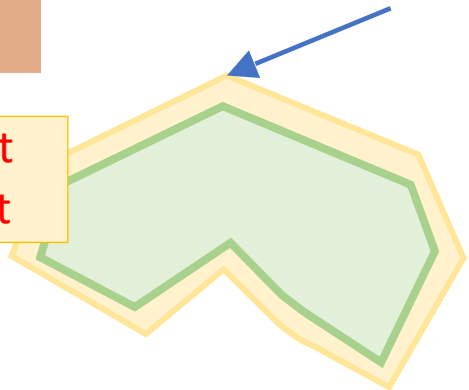
Set Propagation using **NNV**[2]

Robust Conformal Prediction to calculate probabilistic bounds on residual error for trained NN

δ

Over-approximated flowpipe





Distributionally Robust Probabilistic Reach Set

1. Hashemi, N., Lindemann, L., & Deshmukh, J. V. Statistical reachability analysis of stochastic cyber-physical systems under distribution shift, EMSOFT/IEEE TCAD 2024.

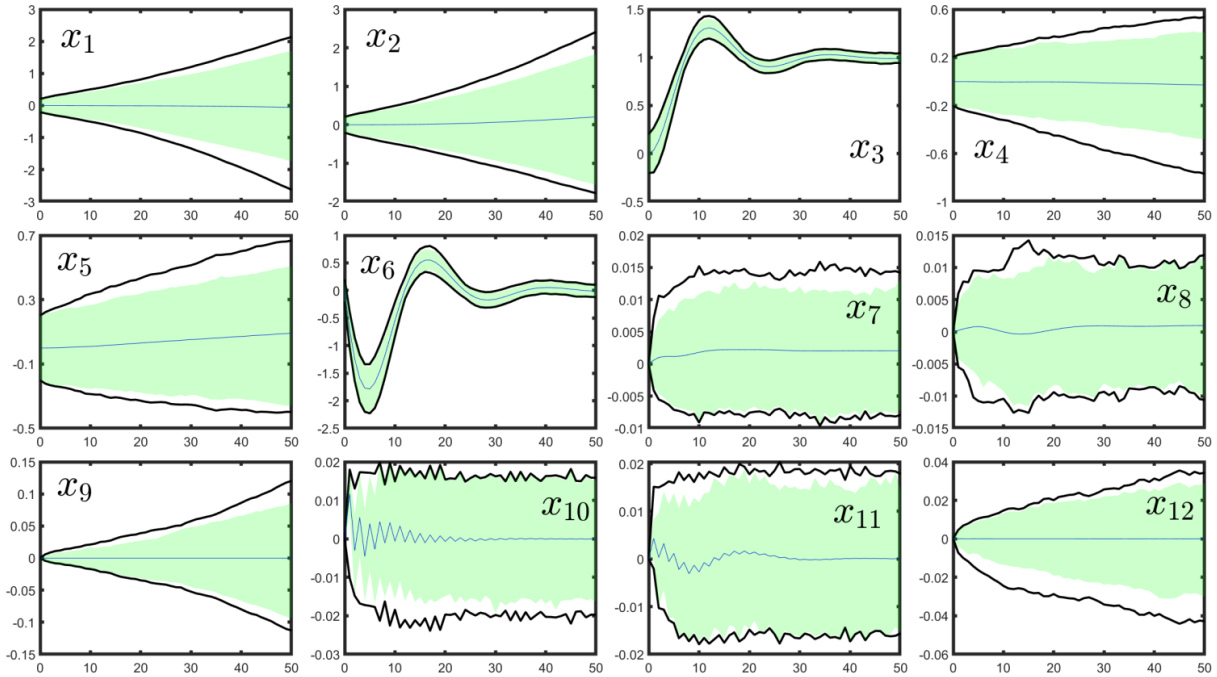
Model-free Reachability Results[1]

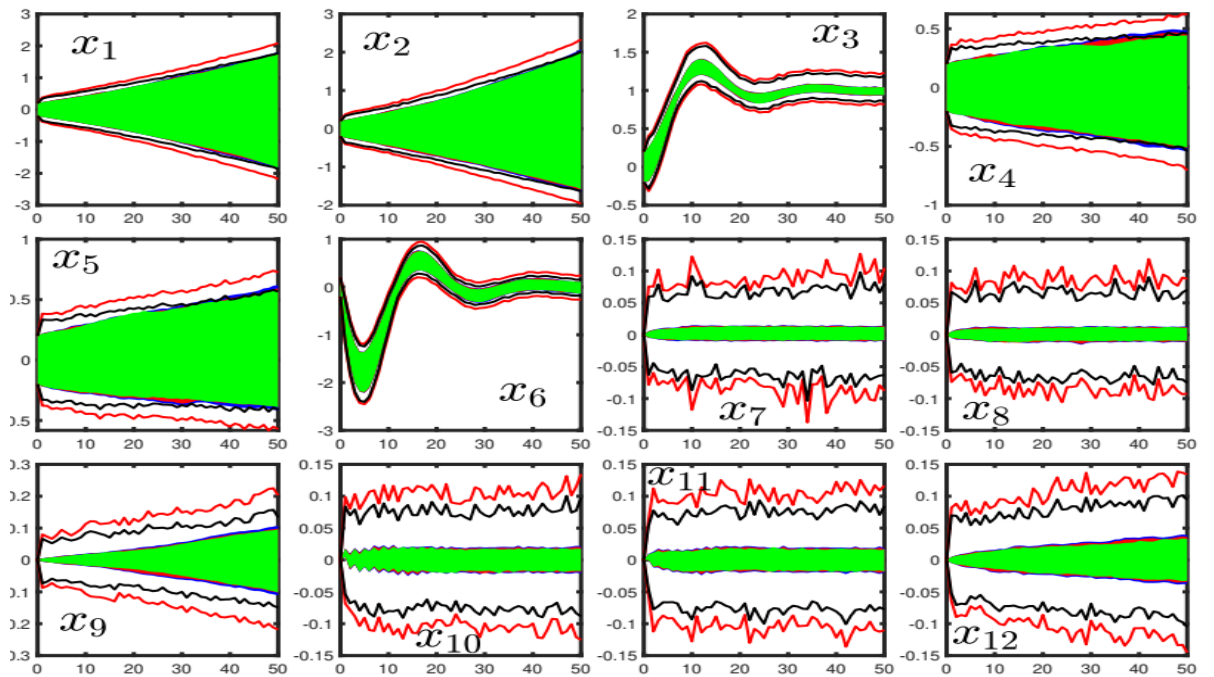
$$
\begin{cases}
\dot{x}_1 = \cos(x_8)\cos(x_9)x_4 + (\sin(x_7)\sin(x_8)\cos(x_9) - \cos(x_7)\sin(x_9))x_5 \\
\quad +(\cos(x_7)\sin(x_8)\cos(x_9) + \sin(x_7)\sin(x_9))x_6 + v_1 \\
\dot{x}_2 = \cos(x_8)\sin(x_9)x_4 + (\sin(x_7)\sin(x_8)\sin(x_9) + \cos(x_7)\cos(x_9))x_5 \\
\quad +(\cos(x_7)\sin(x_8)\sin(x_9) - \sin(x_7)\cos(x_9))x_6 + v_2 \\
\dot{x}_3 = \sin(x_8)x_4 - \sin(x_7)\cos(x_8)x_5 - \cos(x_7)\cos(x_8)x_6 + v_3 \\
\dot{x}_4 = x_{12}x_5 - x_{11}x_6 - 9.81\sin(x_8) + v_4 \\
\dot{x}_5 = x_{10}x_6 - x_{12}x_4 + 9.81\cos(x_8)\sin(x_7) + v_5 \\
\dot{x}_6 = x_{11}x_4 - x_{10}x_5 + 9.81\cos(x_8)\cos(x_7) - 9.81 - u_1/1.4 + v_6 \\
\dot{x}_7 = x_{10} + (\sin(x_7)(\sin(x_8)/\cos(x_8)))x_{11} + (\cos(x_7)(\sin(x_8)/\cos(x_8)))x_{12} + v_7 \\
\dot{x}_8 = \cos(x_7)x_{11} - \sin(x_7)x_{12} + v_8 \\
\dot{x}_9 = (\sin(x_7)/\cos(x_8))x_{11} + (\cos(x_7)/\cos(x_8))x_{12} + v_9
\end{cases}
$$

$$
\mathcal{I} = \left\{ s_0 \;\middle|\; \begin{bmatrix} -0.2 \\ -0.2 \\ -0.2 \\ -0.2 \\ -0.2 \\ -0.2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \leq s_0 \leq \begin{bmatrix} 0.2 \\ 0.2 \\ 0.2 \\ 0.2 \\ 0.2 \\ 0.2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\} \quad (7)
$$

- Trained ReLU NN (MSE vs. Quantile Loss)
- Failure probability = 0.01
- Exact star Reachability with NNV
- Calibration dataset = 40K trajectories

Distribution shift





1. Hashemi, Navid, Xin Qin, Lars Lindemann, and Jyotirmoy V. Deshmukh. "Data-Driven Reachability Analysis of Stochastic Dynamical Systems with Conformal Inference." In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 3102-3109. IEEE, 2023.

Related work on data-driven reachability

Alanwar et al.

- identify Markovian dynamics and then learn do reachability on identified models
- do not currently quantify uncertainty in model learning

Devonport, Arcak, ACC 2020:

- Use Gaussian Processes to separate reachable states from unreachable ones
- May require solving high-dimensional optimization problems over GPs

Lin, Bansal, ICRA 2023:

- Use neural PDE solvers to do Hamilton-Jacobi method-based reachability

Fan, Qi, Mitra, Vishwanathan, CAV 2017:

- Use PAC-learning methods to learn discrepancy functions and use simulation-guided reachability

Devonport, Yang, El Ghaoui, Arcak, CDC 2021:

- Model nonlinear systems using level sets of Christoffel functions and give high-accuracy probabilistic reach sets

Dynamical Systems Reachability: Challenge Problems

- Scaling to 1000s of states
  - May need to scale NN set propagation to 1000s of inputs
  - May need to identify *important* states and prune away others (dimensionality reduction such as PCA)
- Sensors!
  - System state may not be observable directly except through sensors
  - Might need to model system dynamics with Recurrent Neural Networks (RNNs)
- What if sampling is expensive?
  - Need to learn from limited data, give guarantees from limited data: very hard problems
- Conformal prediction approach is a frequentist approach
  - How can we combine with Bayesian reasoning?

Reflections

- Conformal Prediction is a versatile tool to get probabilistic guarantees
  - Define non-conformity score
  - Sample calibration data in an iid fashion
  - Get guarantees in $O(m\log m)$, where m = size of calibration dataset
  - Calibration set size scales linearly with $\sim\frac{1}{\delta}$, for a $(1-\delta)$-guarantee
- Can reason about distribution shifts (if is given or can be estimated)
- Versatile to solve many different problems
  - Predictive monitoring
  - Verification/Reachability analysis
  - Planning
- Marginal guarantees:
  - Guarantees to be interpreted over an implicit marginalization over all possible calibration sets
  - Can get conditional guarantees in some cases

Thank You!