Neural Network Verification in Autonomous Cyber-Physical Systems (CPS)

85th Meeting of IFIP Working Group 10.4 on Dependable Computing and Fault Tolerance: Workshop on Trustworthy AI-Enabled Cyber-Physical Systems
February 2, 2024 (Groundhog Day)

Taylor T. Johnson, PhD, PE, Associate Professor & A. James and Alice B. Clark Foundation Chancellor Faculty Fellow

VeriVITAL - the **Veri**fication and **V**alidation for **I**ntelligent & **T**rustworthy **A**utonomy **L**aboratory

Institute for Software Integrated Systems

Departments of Computer Science & Electrical and Computer Engineering

Vanderbilt University

http://www.taylortjohnson.com/
taylor.johnson@vanderbilt.edu

VANDERBILT UNIVERSITY


PUNXSUTAWNEY, PA
WEATHER CAPITAL OF THE WORLD
PHIL
AP

You

Show me a groundhog day picture illustrating the AI winters

DALL·E

Here are the illustrations depicting Groundhog Day in the context of AI winters. Each image captures the essence of periods of reduced interest and funding in AI research, symbolized through a groundhog, vintage computers, and scattered AI research papers in a snowy landscape.

You

Do you think Phil saw his shadow?

DALL·E

Based on the metaphorical context of these illustrations, if Phil (the groundhog) represents the cyclical nature of AI research interest and funding, seeing his shadow could symbolize predicting a prolonged "winter" or period of stagnation in AI advancements. Given the imagery of a cautious groundhog in a snowy landscape surrounded by symbols of past AI research efforts, it suggests a cautious outlook on the future of AI progress. So, metaphorically speaking, it might imply that Phil saw his shadow, indicating more challenges ahead before a new "spring" of AI innovation.

DALL-E Conversation 2/1/2024, GPT-4
https://chat.openai.com/g/g-2fkFE8rbu-dall-e/c/3bd7bfaa-1de6-437f-b619-6528195dd0ca

Safe, Secure, & Trustworthy AI

WHITE HOUSE

OCTOBER 30, 2023

Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence

BRIEFING ROOM › PRESIDENTIAL ACTIONS

By the authority vested in me as President by the Constitution and the laws of the United States of

- My perspective & background
- Come from the Hybrid Systems: Computation & Control (HSCC) part of CPS/IoTWeek community
- Developing formal verification methods for AI and machine learning (ML) since ~2016
- DARPA Assured Autonomy & ANSR, NSA SoS, NSF FMitF, AFOSR/AFRL, ONR, Toyota, Mathworks, …
- Co-led SafeTAI workshop feeding into NSF Safe-Learning Enabled Systems (SLES) creation, contributed to DARPA AI Forward events, AISoLA'23, …

VeriVITAL Members & Alumni

Current PhD Students & Postdocs


Judy Nguyen


Anne Tumlin


Samuel Sasaki


Preston Robinette
2021 NDSEG


Serena Serbinowska


Dr. Diego
Manzanas Lopez


Dr. Tianshu Bao


Dr. Neelanjana Pal
MathWorks

Postdoc / PhD / Research Scientist Alumni


Dr. Nate Hamilton
2019 NDSEG
Parallax Research


Dr. Xiaodong
Yang
Visa Research


Dr. Patrick Musau
Google


Prof. Weiming Xiang
Augusta University
2022 NSF CAREER


Prof. Hoang-Dung Tran
U Nebraska Lincoln
2021 IEEE TCCPS
Outstanding Dissertation


Prof. Joel Rosenfeld
USF
2021 AFOSR YIP


Prof. Luan Nguyen
U Dayton
2023 NSF CRII


Prof. Omar Beg
U Texas PB
2019 UT System
Rising STARs


Dr. Shafiul Chowdhury
Meta, ML Senior
Research Scientist

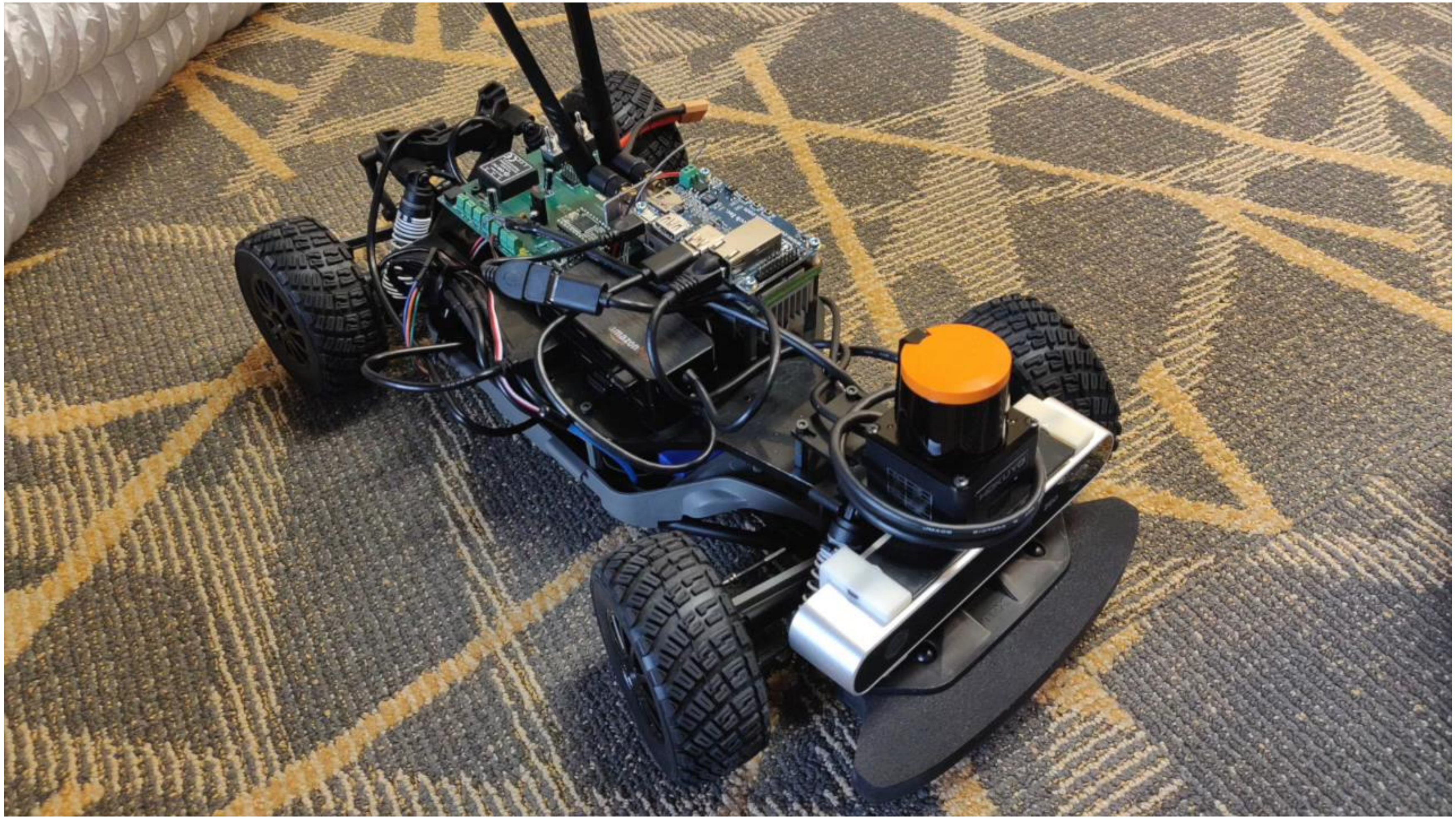

Prof. Khaza Hoque
U Missouri


Dr. Andrew
Sogokon
Southampton /
Lancaster

MSc Thesis / Undergrad Researcher Alumni: at Google, Meta, Microsoft, Amazon, Qualcomm, Rivian, etc.

Motivation: Autonomous Cyber-Physical Systems (CPS)

Patrick Musau

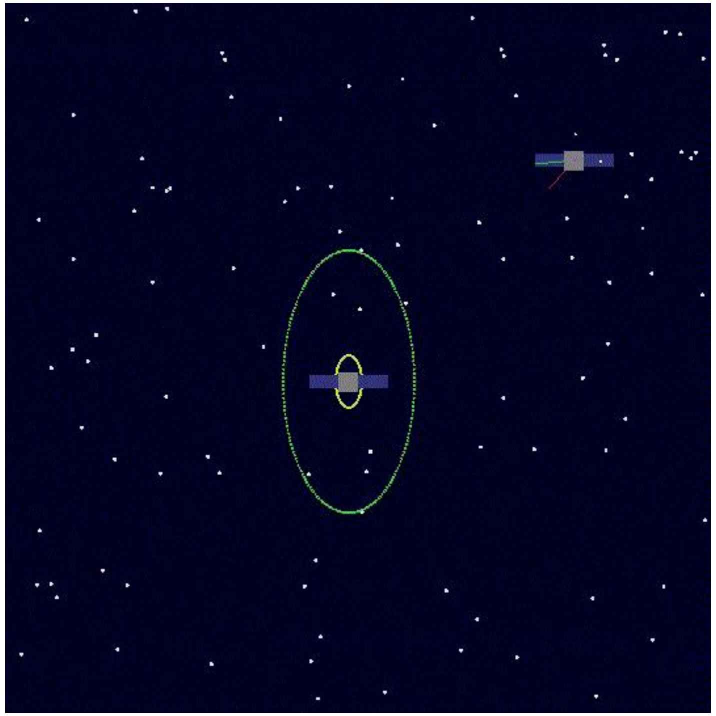


https://f1tenth.dev/

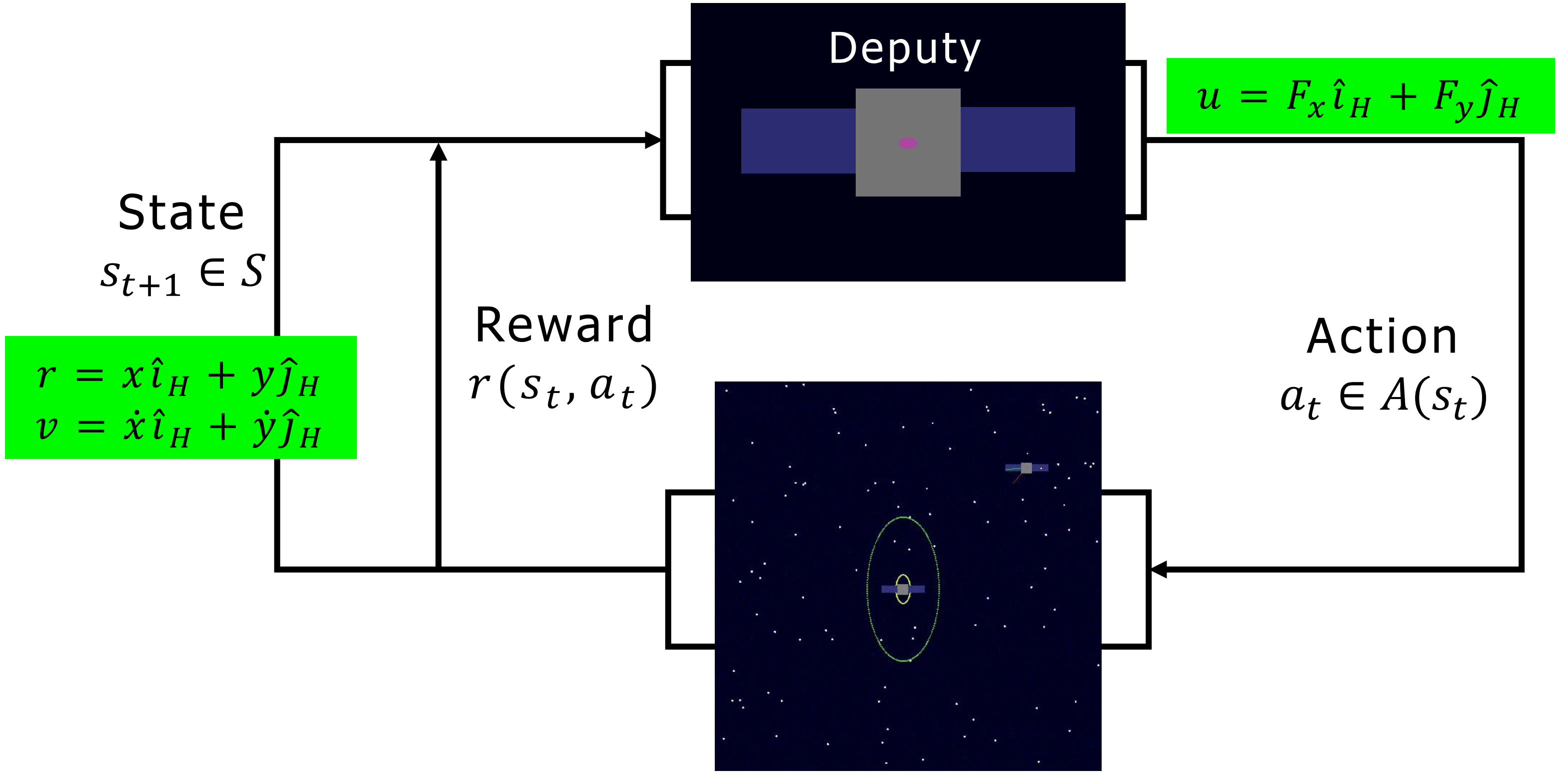
- CPS: modern embedded control systems, specifically where there is a tight coupling between software (cyber) and physical processes
- Examples: cars, aircraft, IoT devices, etc.
- Most CPS involve networking and increasingly involve machine learning components, such as neural networks (NNs)
- F1/10 Architecture: LIDAR and stereo camera sensory data processed by NNs on NVIDIA Jetson TX2

Motivation: 2D Spacecraft Docking

Preston Robinette



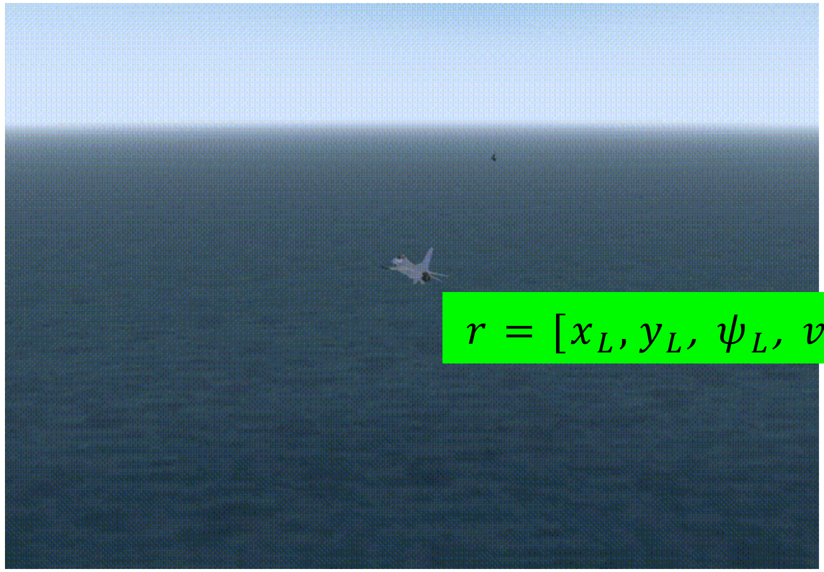
$\varphi_{docking}: (r \leq \rho_d)$

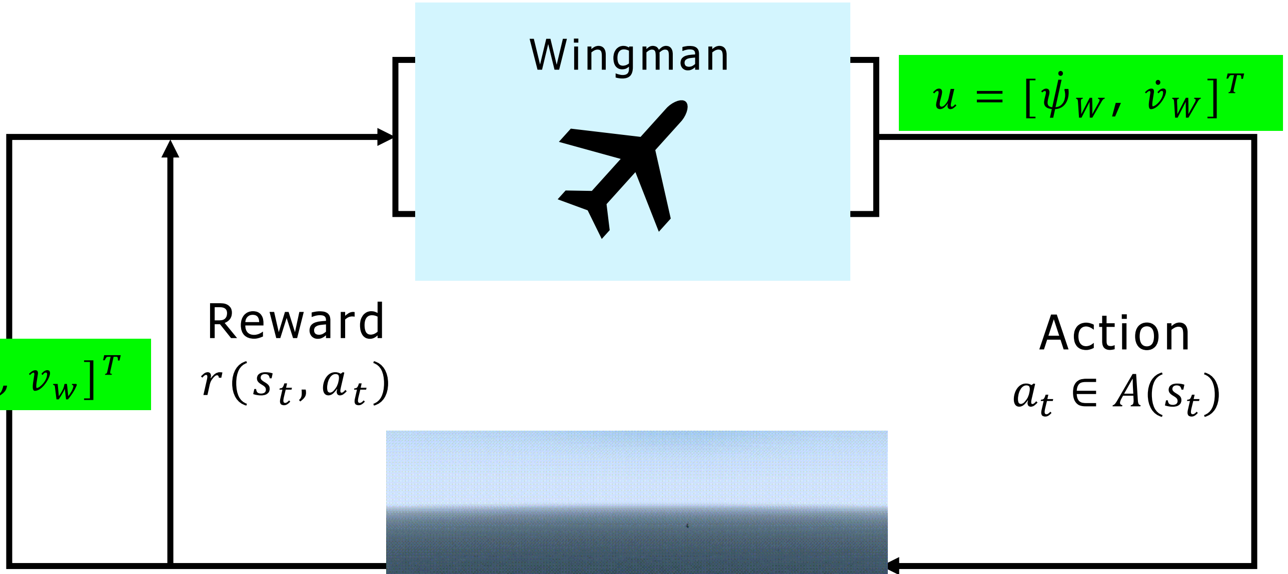


Developed with Kerianne Hobbs AFRL/RQ&RY, along with Umberto Ravioli, Preston Robinette, Nate Hamilton
https://github.com/act3-ace/SafeRL and https://github.com/act3-ace/aerospaceRL

Motivation: 2D Dubins Rejoin

Preston Robinette





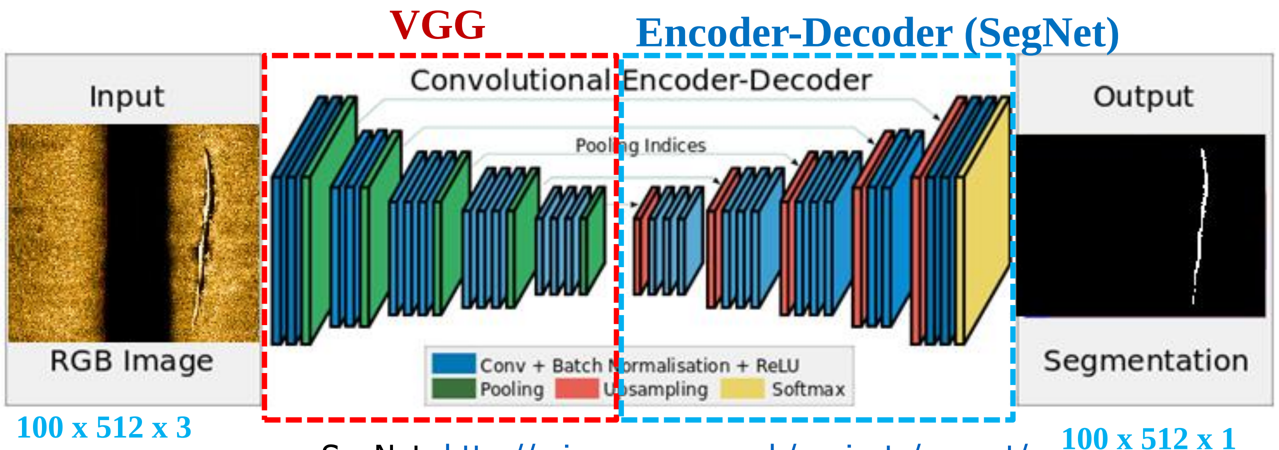
$\varphi_{formation}: \left(\sqrt{(x_W - x_r)^2 + (y_W - y_r)^2} \leq \rho_e\right) \wedge (t_{rejoin} \geq t_{success})$

Developed with Kerianne Hobbs AFRL/RQ&RY, along with Umberto Ravioli, Preston Robinette, Nate Hamilton
https://github.com/act3-ace/SafeRL and https://github.com/act3-ace/aerospaceRL

Motivation: Autonomous Underwater Vehicles (AUVs)

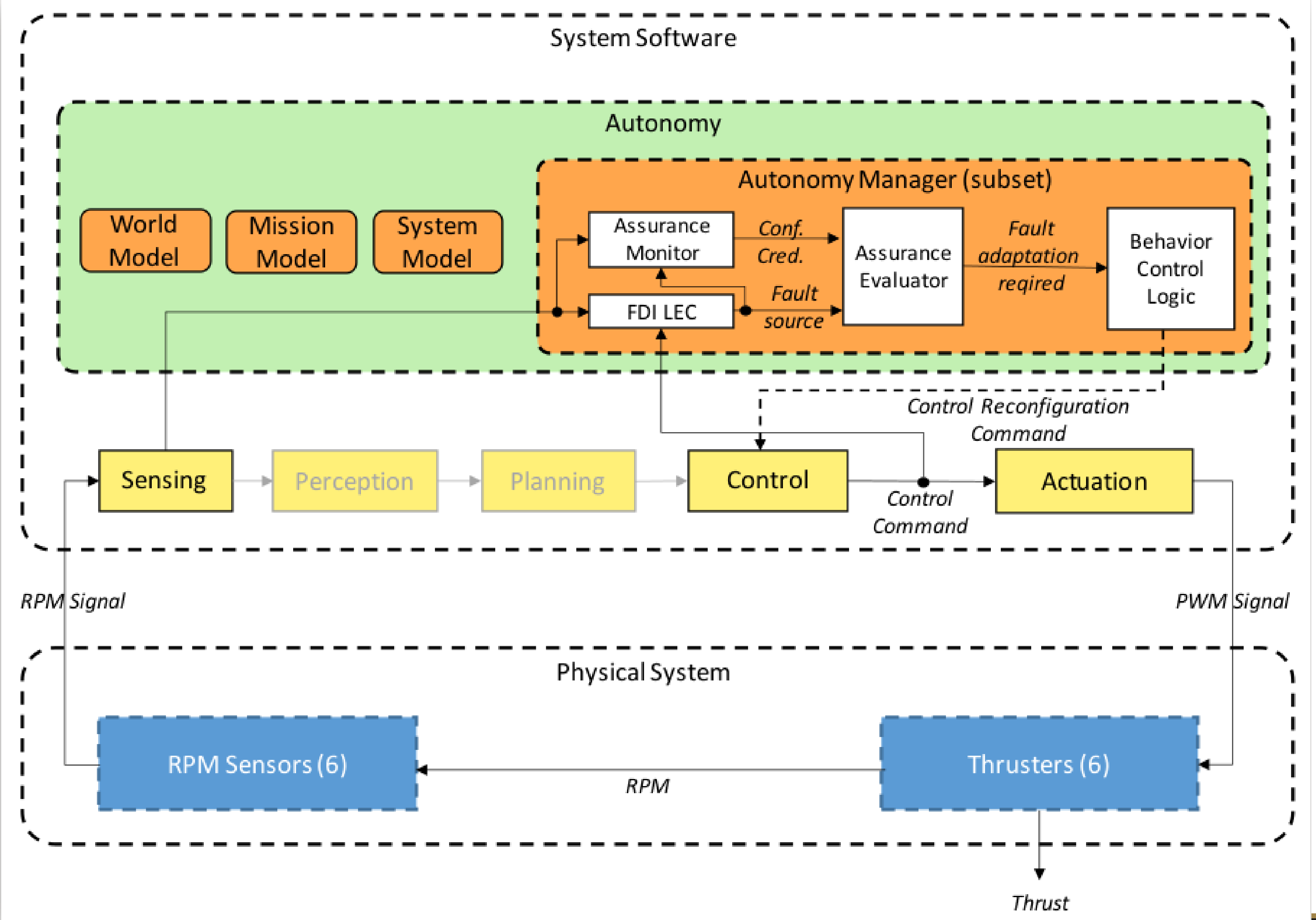
- Paired with Northrop Grumman in **DARPA Assured Autonomy** program
  - Assured Autonomy: “The goal of the Assured Autonomy program is to create technology for continual assurance of Learning-Enabled, Cyber Physical Systems (LE-CPSs).”
- Our role: develop verification methods for autonomous systems, work with Northrop to apply them to AUV scenarios
- A specific challenge problem: robustness verification of neural networks used for **semantic segmentation**, specifically processing sonar data for identifying obstacles and targets to track (e.g., pipes/cables) underwater
  - Can think of sonar data as images (just acquired slowly)

https://www.l3harris.com/all-capabilities/iver3-ep-open-system-uuv

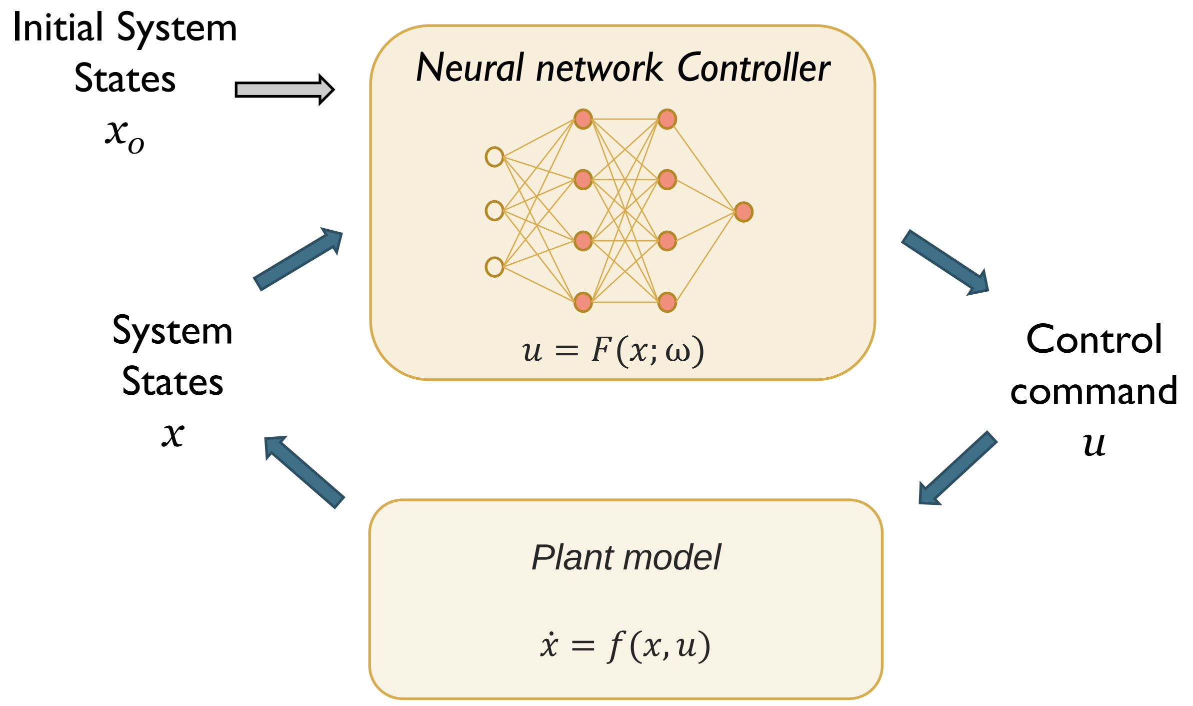


SegNet: http://mi.eng.cam.ac.uk/projects/segnet/

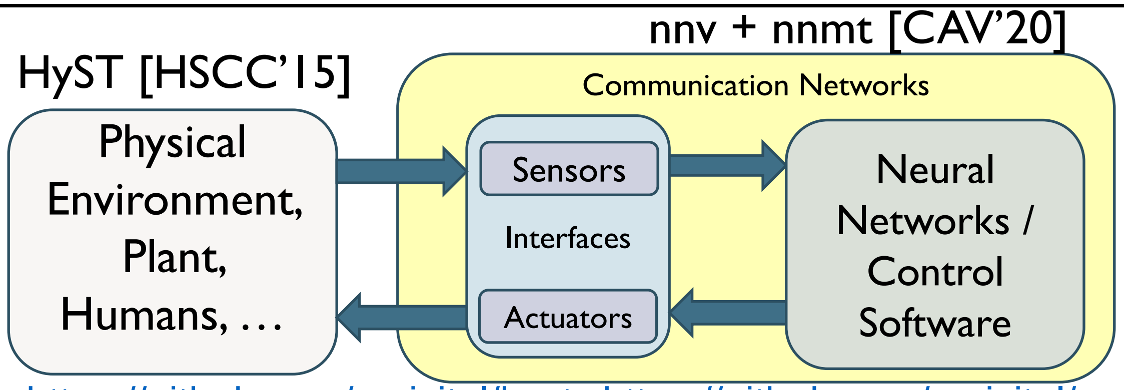
Representative AUV Architecture (BlueROV)



Neural Network Control Systems (NNCS)

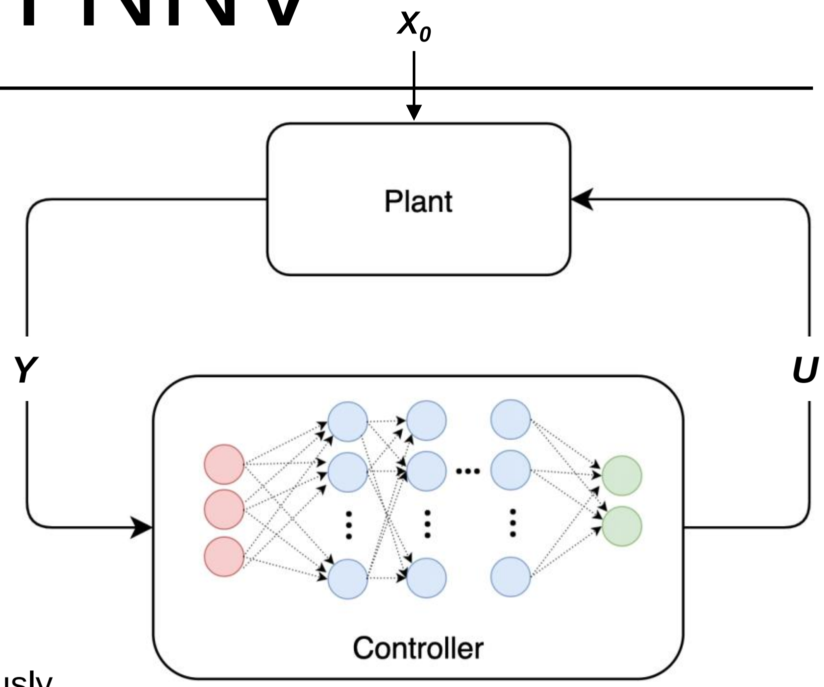


Closed-Loop Verification with NNV



https://github.com/verivital/hyst https://github.com/verivital/nnv

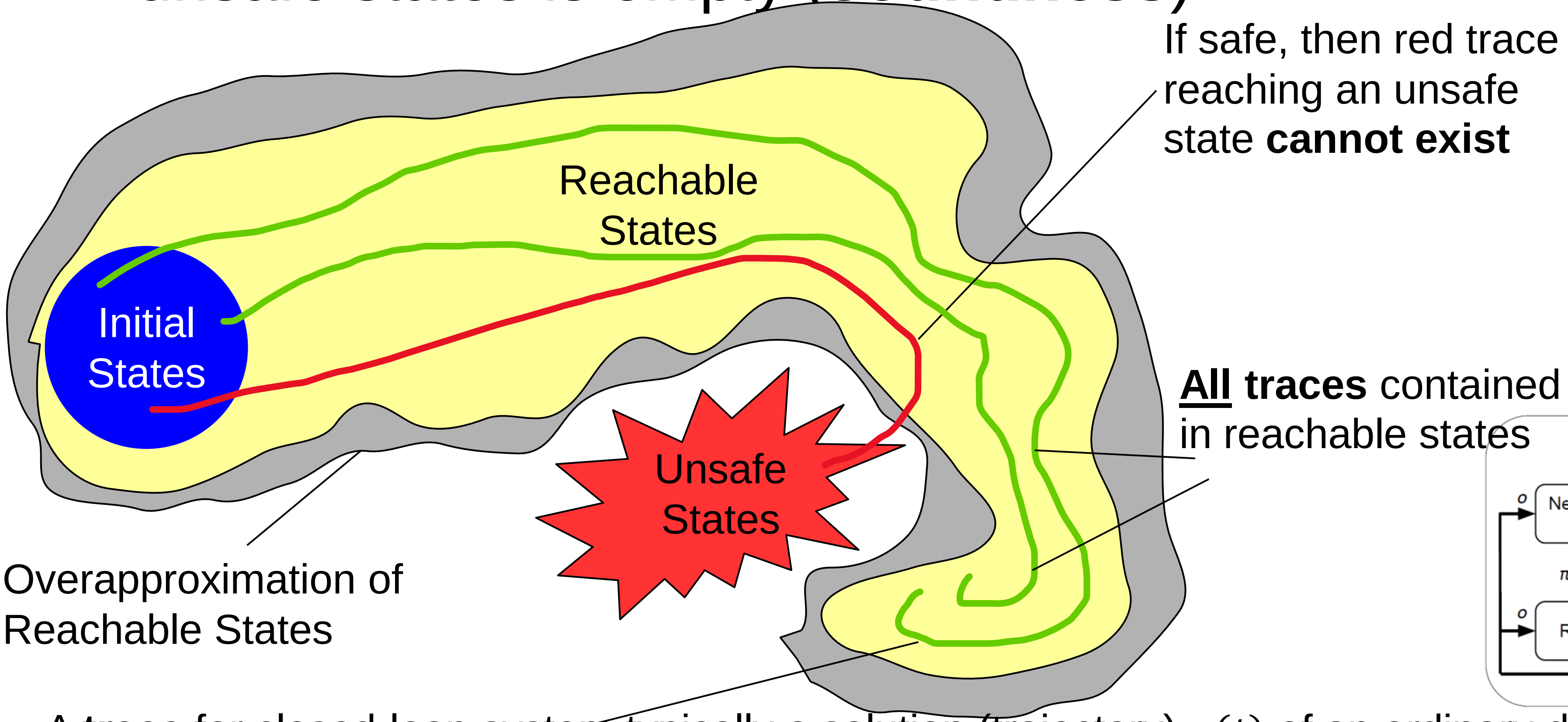
https://cps-vo.org/group/hyst https://github.com/verivital/nnmt

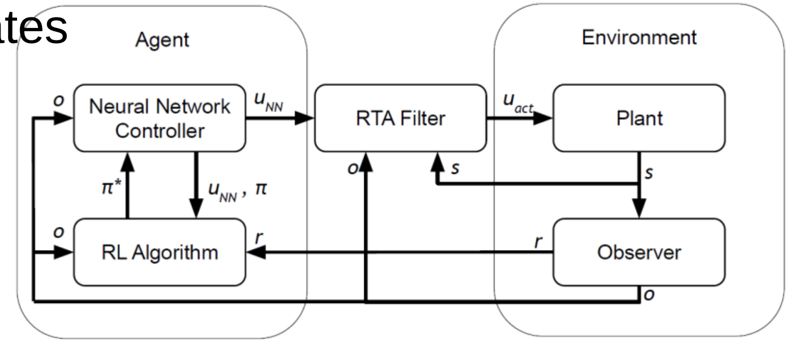


- Plant models: **hybrid automata**, or networks thereof, represented in HyST/SpaceEx/CIF formats
  - Hybrid automaton: **finite state machine** + set of real-valued variables that evolve continuously over intervals of real time according to **ordinary differential equations (ODEs)**
  - **Hybrid** behaviors: discrete transitions and continuous trajectories over real time
  - ODEs: linear or nonlinear (uses CORA for nonlinear)
- LEC and cyber models: for now, feedforward **neural networks**, represented in **ONNX** format (compatible with Keras, Tensorflow, Matlab, etc.)
  - Primarily focused on ReLUs, but recent support for nonlinear activations
- Specifications: primarily **safety properties** for now, some reachability properties
- Verification: composed LEC and plant analysis
  - **Bounded model checking**: k control periods, alternating reachability analysis of controller and plant

Safety Verification of Closed-Loop Autonomous Systems with Reachability

- **Safe** if intersection of overapproximation of **reachable states** with unsafe states is empty (**soundness**)



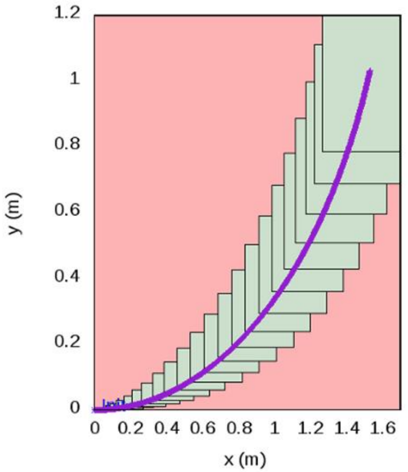


A **trace** for closed-loop system typically a solution (trajectory) $x(t)$ of an ordinary differential equation (ODE) $\dot{x} = f(x, u)$ or generalization thereof (hybrid automata, differential inclusion, etc.)

Monitoring: Runtime (Online) Verification of Autonomous Systems with Real-Time Reachability


Patrick Musau

- *Several orders of magnitude* progress in past few years made analyzing **learning-enabled components (LECs)** like neural networks and usage in autonomous CPS at ***design-time*** with NNV and other approaches (***scalability***, ***layer types***, ***closed-loop interaction***, etc.)
- However, while improving confidence of such LECs before they are deployed is important, ***online monitoring at runtime*** is essential
- How can we provide formal and provable guarantees of system-level behaviors, such as safety, **online at runtime**?
  - Key idea: abstract LEC behaviors (see other approaches on out of distribution detection, etc.) and simply ***observe the influence of their behavior on plant/system-level at runtime***
  - Necessary technology: **online reachability analysis** of plant models, ideally with worst-case execution time (**WCET**) guarantees for implementation in embedded hardware
  - Builds on **real-time reachability** of linear/nonlinear ordinary differential equations (ODEs) and hybrid automata with WCET guarantees, implemented as an **anytime** algorithm [FORTE'19, TECS'16, RTSS'14]
  - Based on **mixed face lifting reachability** [Dang and Maler, HSCC'98 & HSCC'19 Test of Time Award Winner], using hyperrectangles (intervals) as state-space representation

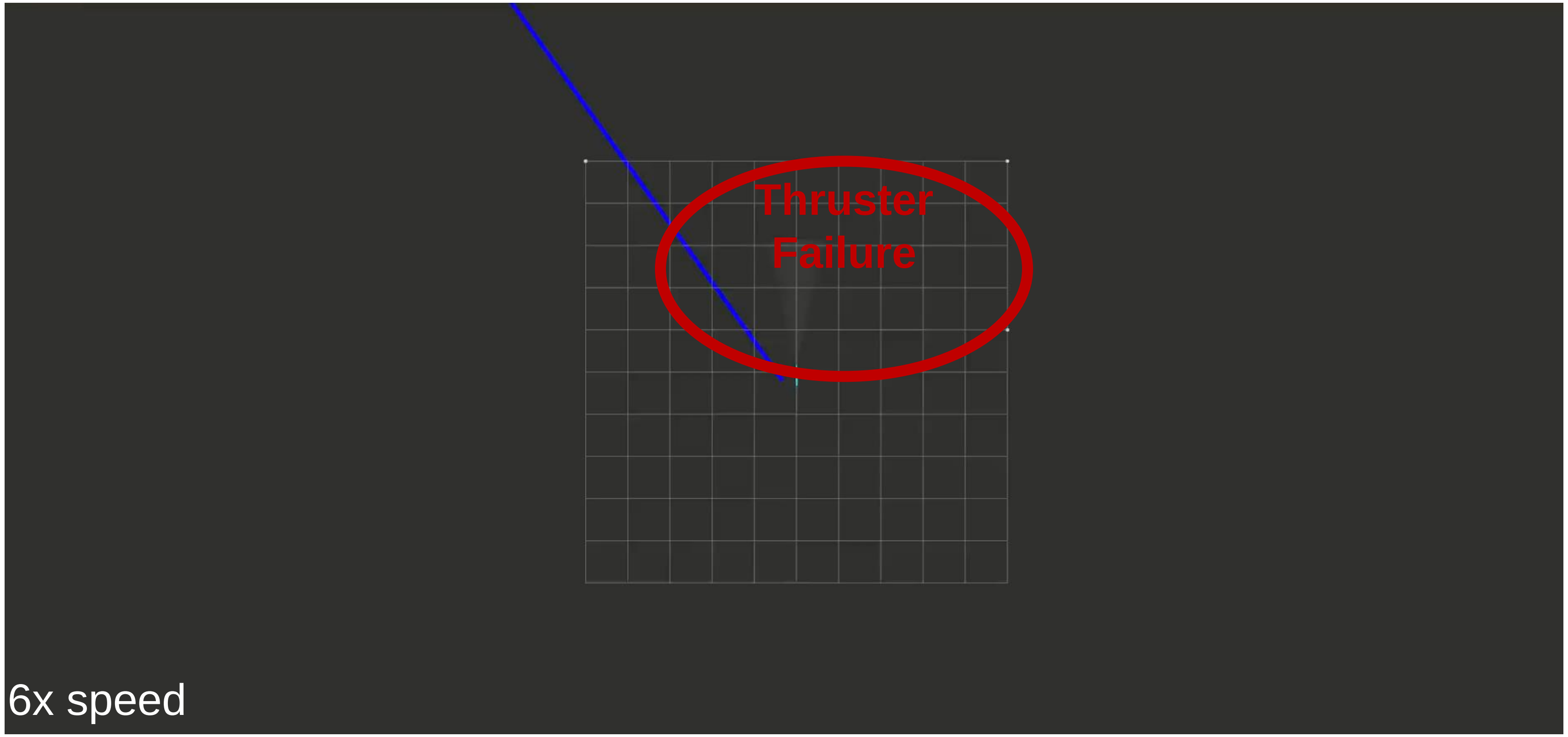


[Musau et al, "On Using Real-Time Reachability for the Safety Assurance of Machine Learning Controllers", **ICAA'22**]
[Tran et al, "Decentralized Real-Time Safety Verification for Distributed Cyber-Physical Systems", **FORTE'19**]
[Johnson et al, "Real-Time Reachability for Verified Simplex Design", **TECS'16**]
[Bak et al, "Real-Time Reachability for Verified Simplex Design", **RTSS'14**]

http://www.verivital.com/rtreach/

Monitoring AUV Waypoint Following Mission with Real-Time Reachability: Degraded Operation and Obstacle Avoidance

Patrick Musau



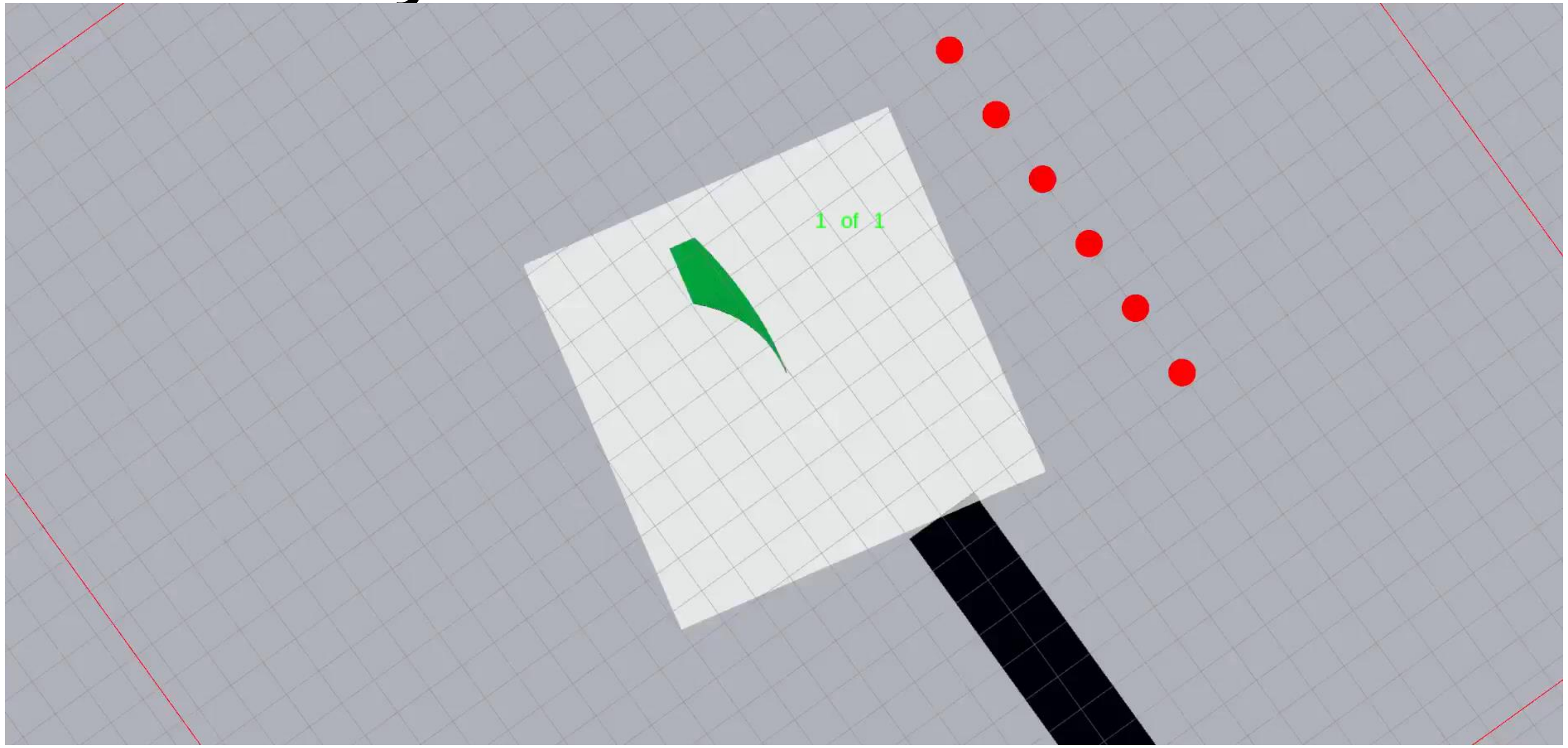
Legend
- Green curve: past trajectory
- Light-blue set: nominal control reachable set projected forward in time
- Red set: degraded control reachable set projected forward in time
- Gray dots: waypoints
- Darker blue: pipeline
- Red/blue around obstacle: sonar detection
- Light gray cone: forward looking sonar

Based on ROS2 / UUVSim framework: https://uuvsimulator.github.io/

[Musau et al, "On Using Real-Time Reachability for the Safety Assurance of Machine Learning Controllers", **ICAA'22**]
[Tran et al, "Decentralized Real-Time Safety Verification for Distributed Cyber-Physical Systems", **FORTE'19**]
[Johnson et al, "Real-Time Reachability for Verified Simplex Design", **TECS'16**]
[Bak et al, "Real-Time Reachability for Verified Simplex Design", **RTSS'14**]
http://www.verivital.com/rtreach/

Monitoring AUV Geo-fencing under Uncertainty

Patrick Musau

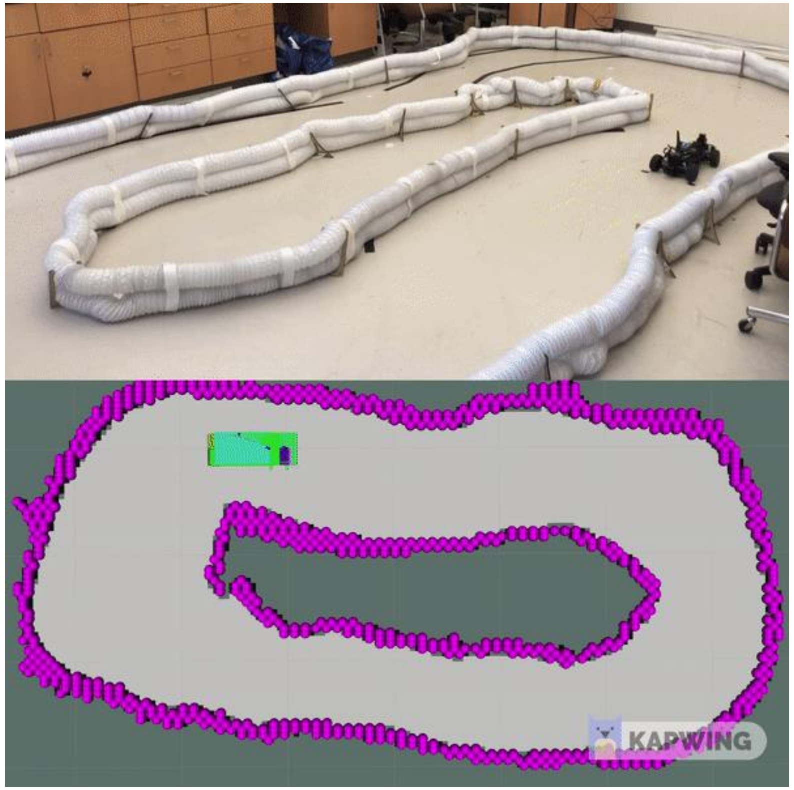


- Green Boxes: Safe Trajectory
- Red Boxes: Unsafe Trajectory
- Red Dots: Static Obstacles
- White box: local costmap
- Black squares: no-go zone or possible location of obstacles
- Red Box: Geo-Fenced area boundary

- Localization Uncertainty Considered in Above Video
    - x +- 0.025 %
    - y +- 0.025 %
    - speed +- 0.025 %
    - heading +- 0.025 %
- Unsafe Event Handling via behavior Tree

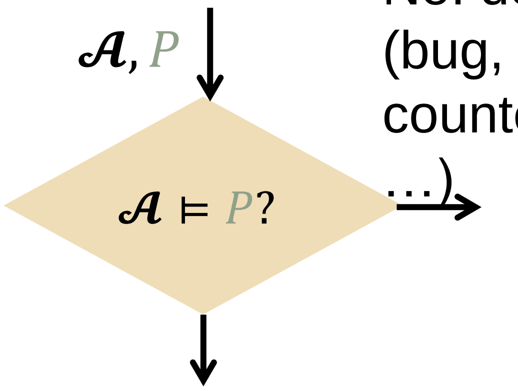
Outline for Remainder

- Why formal methods for AI/ML/NNs?
  - Approach for safe & trustworthy AI
- NNV: Verifying neural networks in autonomous cyber-physical systems
  - Focusing mostly on neural networks in these learning-enabled systems (open loop vs. closed loop)
- Conclusions



Formal Verification Challenge

- ***Formal verification problem:*** Given a **system model** $\mathcal{A}$ and a **specification (requirement) *P***, prove that $\mathcal{A}$ **satisfies *P***
- Automated formal verification: model checking
- Model checking ***algorithms*** return:
  - $\mathcal{A}$ satisfies P and give ***proof*** or
  - $\mathcal{A}$ violates P and why (***bug***)
  - With abstraction, possibly unknown
- Engineering / CS grand challenge
  - Debugging and verification: ***about 50–75%*** engineering cost [Beizer, 1990]
  - Expensive and life-threatening defects: ***about $60 billion/year*** [NIST, 2002]
  - State-space explosion (“curse of dimensionality”) and undecidability
  - Related to simulation/testing, but different: “Program testing can be used to show the presence of bugs, but never their absence!”- Edsger W. Dijkstra
  - These days at intersection of software engineering and theory



No: defect (bug, counterexample, …)

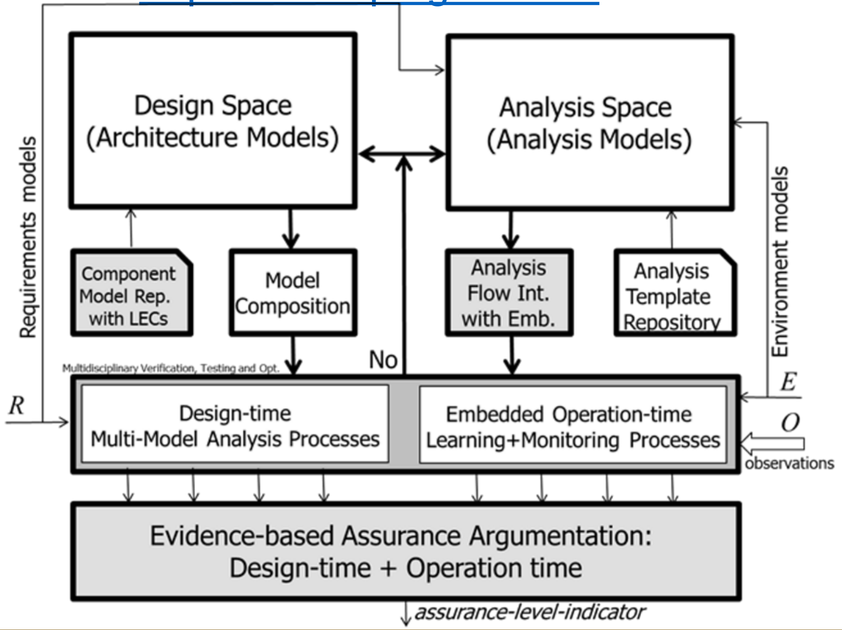
Yes: proof (mathematical and typically machine checkable, evidence for certification authority or regulator, …)

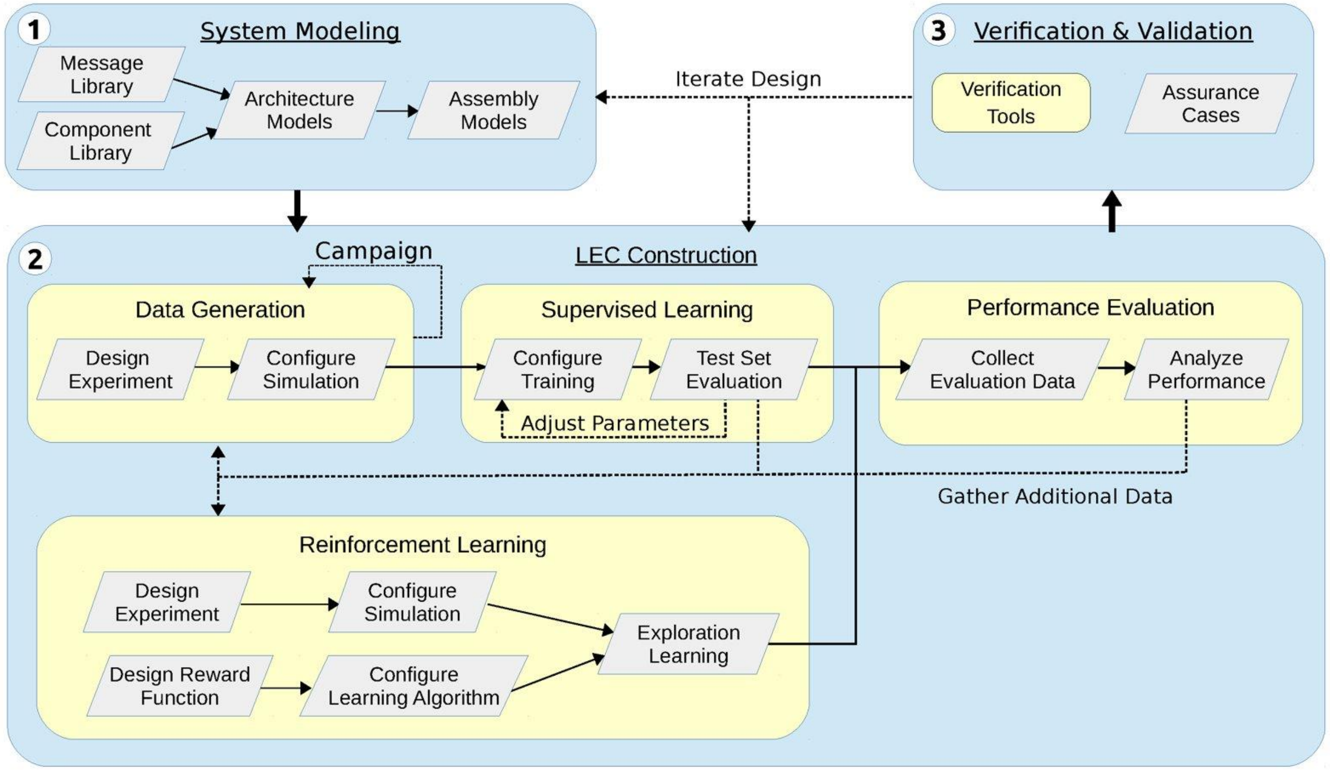
Overview and Motivation for Formal Verification in Machine Learning

- Machine learning (ML) components, such as neural networks, are increasingly being deployed as subcomponents in safety-critical autonomous/semi-autonomous cyber-physical systems (CPS) that have strict regulatory requirements
  - Tasks for these ML components, which we also call **learning-enabled components (LECs)**, range from **sensing, estimation, and perception** to **planning and control**
- Challenge: Strict regulatory requirements for these systems, including for software/computer components and their safety
  - Examples: Aerospace: **DO-178C**, **DO-333**; Automotive: **ISO 26262**; Medical devices: **IEC 62304**, …
    - Advocates usage of formal methods and verification in software development processes
  - No one fully knows what to do for machine learning components yet…
    - Based on our ongoing interactions with companies (Boeing, Northrop, Toyota, Collins, GM, etc.), regulatory/standards bodies (**FAA**, **NHTSA**, **NRC/IAEA**, **SAE**, etc.)
    - But there are significant concerns in safety-critical domains
  - Ongoing accepted best practices
    - Analyze as much as possible at design time (formal verification, augmentation, simulation, etc.)
    - Monitor at runtime (runtime verification, supervisory control, dataset shift, OOD detection, etc.)

Assurance-based Learning-enabled Cyber-Physical Systems (ALC) Toolchain

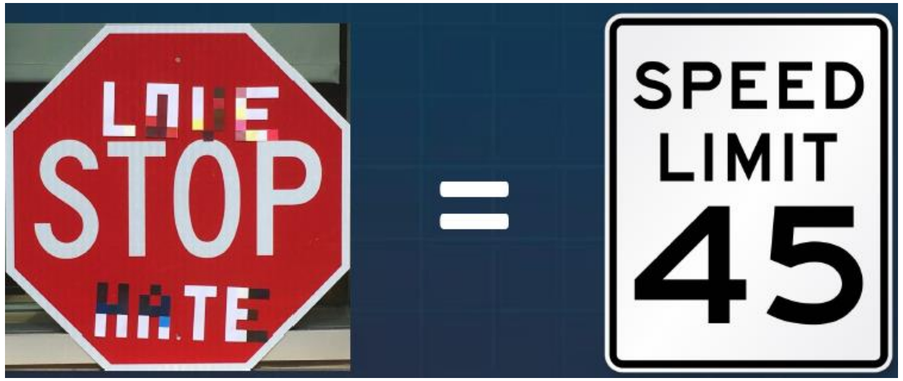
- Focusing on verification efforts, but everything is integrated into a broader toolchain covering modeling, training, verification, assurance, runtime monitoring, etc.
- Latest release:
  - https://github.com/AbLECPS/alc
  - https://ablecps.github.io/





Challenge: Robustness of Neural Networks, Especially when used in CPS

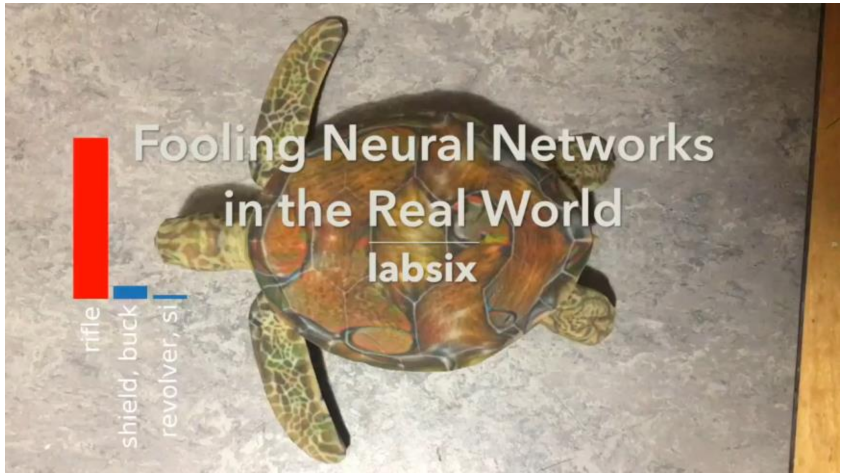
- Neural networks are susceptible to adversarial perturbations: they are not robust to small changes in inputs, as small changes in inputs may cause drastic changes in outputs
- Example: noise of various forms applied to inputs may cause networks to change classification results for image classifiers
- Many forms and types: single pixel attacks, physical perturbations, etc.
- Additional risks and challenges when used within CPS
- Typical formalization: local adversarial robustness under L-infinity (or other) norm
  - Intuition: for image classification, nearby test data yield the same classes (so adversarial perturbation or noise does not change class)

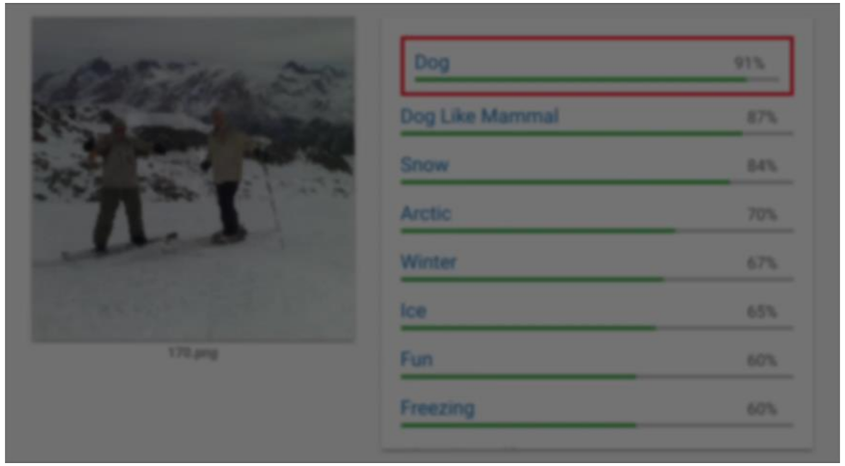


[Eykholt et al, CVPR 2018]

Challenges for Assurance of Learning-Enabled Components (LECs)

- Nontransparency
  - LECs encode information in a complex manner, and it is hard for humans to reason about the encoding
- Error rate
  - LECs typically exhibit some nonzero error rate
  - True error rate unknown, and only estimates from statistical processes known
- Training-based
  - Training dataset is necessarily incomplete
- Potentially unpredictable behavior
  - Training based on nonconvex optimization algorithms and may converge to local minima
  - Changing training dataset may change behaviors
- LECs can exhibit unique hazards
  - ***Adversarial examples*** (incorrect output for a given input that cannot be discovered at design time): whole field of ***adversarial machine learning***
  - May be always possible to find adversarial examples
  - Perception of environment difficult to specify



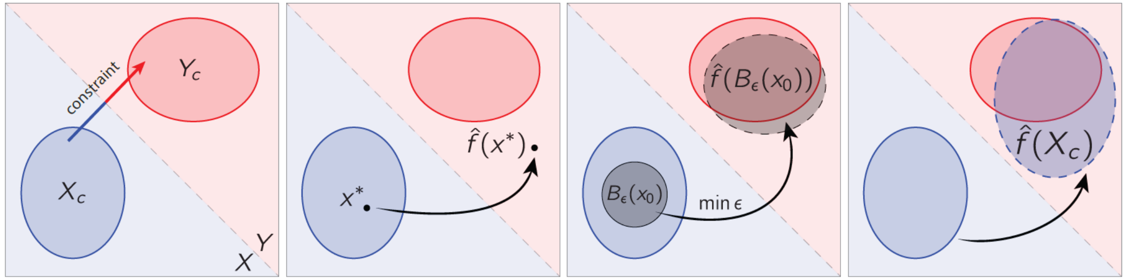


https://www.labsix.org

Why Formal Verification in ML?

- Advocated by DARPA Assured Autonomy / ANSR, NSF FMitF / SLES, by safety organizations (EASA), showing up in NIST AI RMF / reports, etc.
- If we can specify precisely what ML components should/should not do, this may aid in **understandability** and **explainability**
- Analyzing these specifications for ML components can **address concerns related to lack-of robustness**
- Needs to be automated (model checking-like approaches, not theorem proving)
- Challenges
  - Specifications, …
  - Scalability, …
  - Evaluation w.r.t. data sets, …

EASA European Union Aviation Safety Agency *Daedalean – EASA CoDANN IPC Extract – Chapter 6* 56



(a) Legend (b) Counterexample result (c) Adversarial result (d) Reachability result

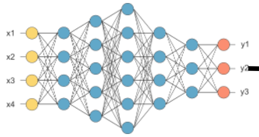
Figure 6.6: Three types of formal verification results. (a): notation. Light-blue and light-red areas X, Y depict input and output spaces; blobs X_c, Y_c depict the constraint sets. (b): a counterexample result represents a single datapoint x^* that violates the constraint. (c): an adversarial result represents the minimum perturbation ε of the input around x_0 that still violates the constraint. (d): reachability result represents the image $\hat{f}(X_c)$ in the output space.

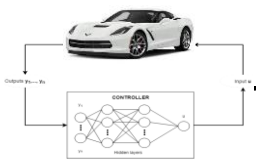
“Concepts of Design Assurance for Neural Networks (CoDANN)”, EASA, 2020
https://www.easa.europa.eu/document-library/general-publications/concepts-design-assurance-neural-networks-codann
SAE G-34 / EUROCAE WG-114

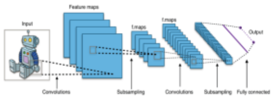
Neural Network Verification (NNV) Software Tool

Hoang-Dung Tran

$M \triangleq$







*The Neural Network Verification (**NNV**) Tool*

https://github.com/verivital/nnv

Feedforward Neural Networks (FFNN)

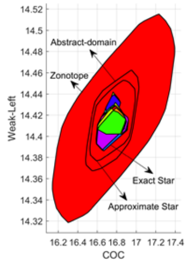
Neural Network Control Systems (NNCS)

Convolutional Neural Networks (CNN)

Reachability Solvers

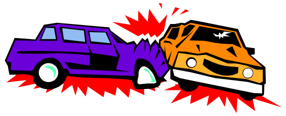
Visualizer

Verifier
$M \models S$?



No: bug

Yes: proof

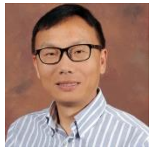
$S \triangleq \neg$ 

$S \triangleq \neg$ 

[Eykholt et al, CVPR 2018]

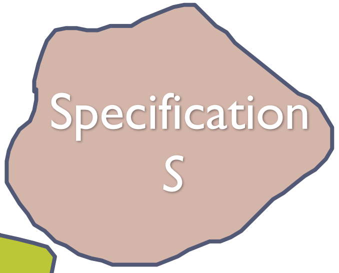
[Xiang et al, “Output Reachable Set Estimation and Verification for Multi-Layer Neural Networks”, **TNNLS’18**]
[Tran et al, “Star-Based Reachability Analysis for Deep Neural Networks”, **FM’19**]
[Tran et al, “Safety Verification of Cyber-Physical Systems with Reinforcement Learning Control”, **EMSOFT’19**]
[Tran et al, “NNV: The Neural Network Verification Tool for Deep Neural Networks and Learning-Enabled Cyber-Physical Systems”, **CAV’20**]
[Tran et al, “Verification of Deep Convolutional Neural Network using ImageStars”, **CAV’20**]
[Bak et al, “Improved Geometric Path Enumeration for Verifying ReLU Neural Networks”, **CAV’20**]
[Xiang et al, “Reachable Set Estimation for Neural Network Control Systems: A Simulation-Guided Approach”, **TNNLS’20**]
[Tran et al, “Robustness Verification of Semantic Segmentation Neural Networks using Relaxed Reachability”, **CAV’21**]
[Lopez et al, “Evaluation of Neural Network Verification Methods for Air to Air Collision Avoidance”, **AIAA JAT’22**]
[Lopez et al, “Reachability Analysis of a General Class of Neural Ordinary Differential Equations”, **FORMATS’22**]
[Lopez et al, “NNV 2.0: The Neural Network Verification Tool”, **CAV’23**]

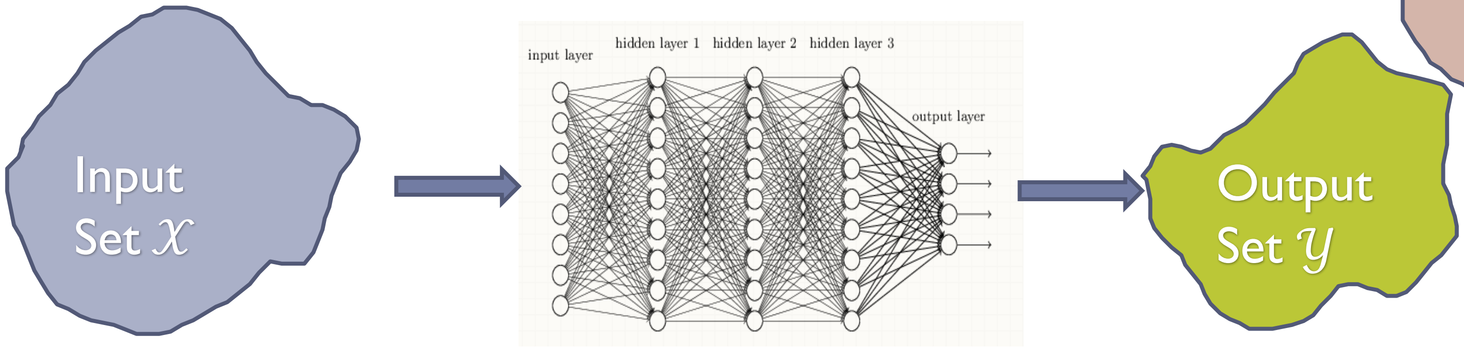
Neural Network Verification with Reachability



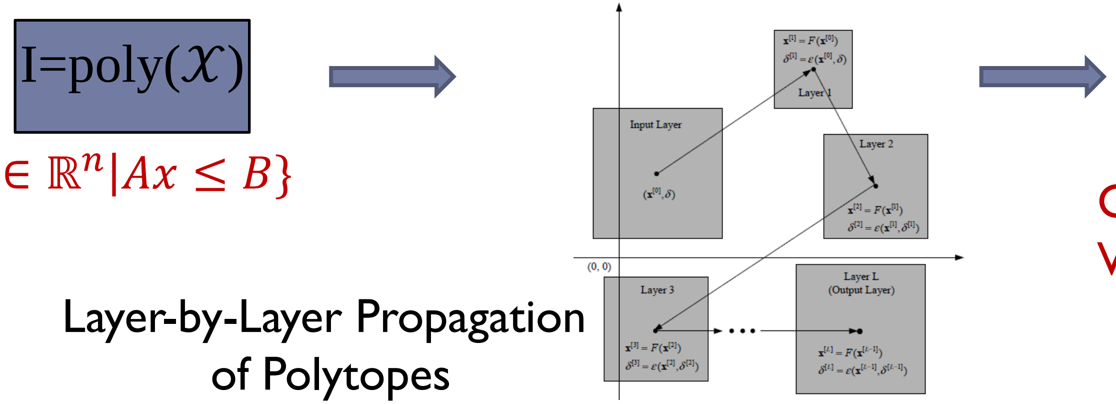
Weiming Xiang

- Given a NN $M: \mathbb{R}^n \mapsto \mathbb{R}^m$ & an input set $\mathcal{X} \subseteq \mathbb{R}^n$, the **output reachable set** of M is $Y = \{y \mid y = M(x), \forall x \in \mathcal{X}\} \subseteq \mathbb{R}^m$



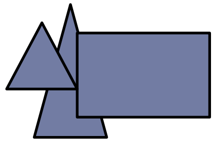


- Computationally: Given a NN M, a convex initial set of inputs I represented as a polytope poly($\mathcal{X}$), compute the output set $Y = M(I)$ of the network



$I = \{x \in \mathbb{R}^n | Ax \leq B\}$

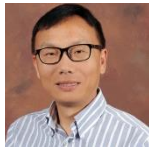
Layer-by-Layer Propagation of Polytopes



Compute: $Y = M(I)$

Verify: $M(I) \cap S = \emptyset$?

Neural Network Reachability Illustrative Example



Weiming Xiang

Given a NN $M: \mathbb{R}^n \mapsto \mathbb{R}^m$ & an input set $\mathcal{X} \subseteq \mathbb{R}^n$, the **output reachable set** of M is $\mathrm{Y} = \{y \mid y = M(x), \forall x \in \mathcal{X}\} \subseteq \mathbb{R}^m$

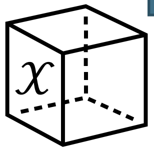
M: simple feedforward NN with 3 inputs, 2 outputs, 7 hidden layers of 7 neurons each, ReLU activations; $M: \mathbb{R}^3 \to \mathbb{R}^2$

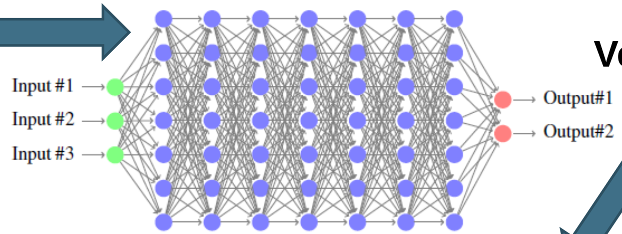
Input set: $\mathcal{X} \triangleq \{x \in \mathbb{R}^3 \mid \|x\|_\infty \leq 1\}$

Specification:

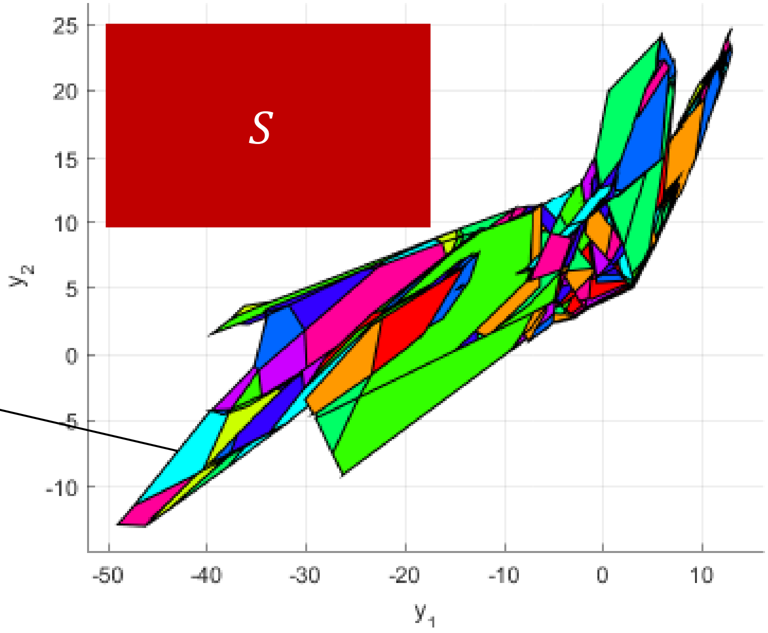
$S \triangleq \{y \in \mathbb{R}^2 \mid -50 \leq y_1 \leq -20 \wedge 10 \leq y_2 \leq 25\}$

Verify: $M(\mathcal{X}) \cap S = \emptyset$?

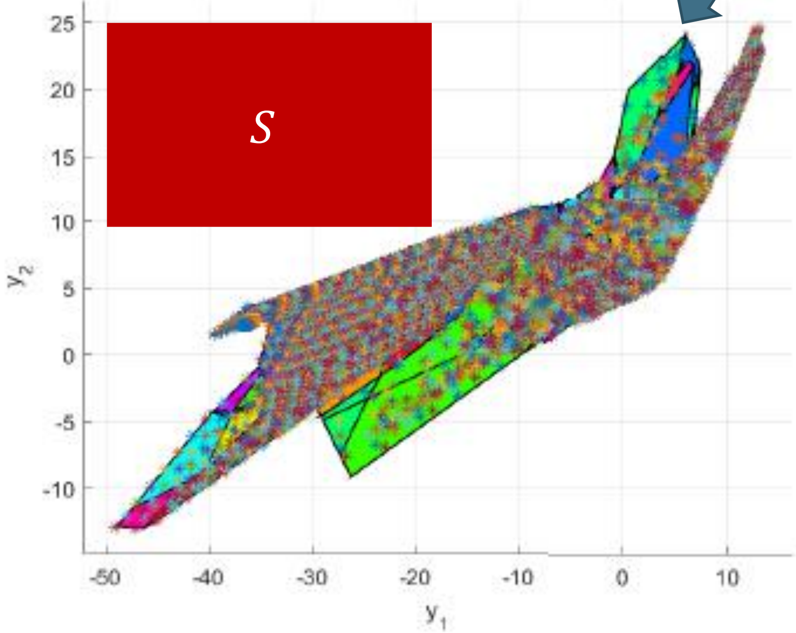




$M(\mathcal{X})$



Output reachable set $Y = M(\mathcal{X})$: union of 1250 polytopes, shown in different colors

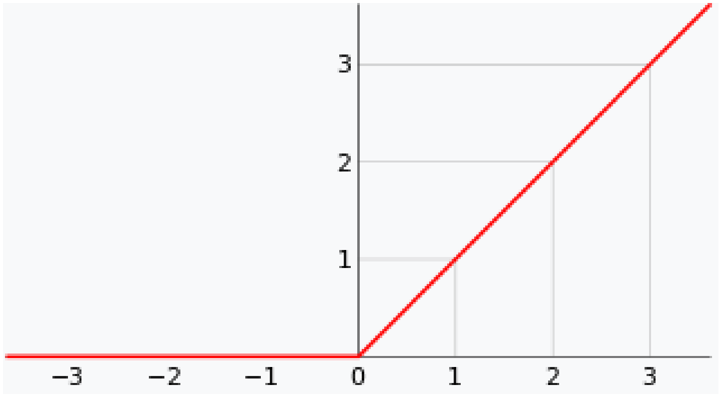


8000 randomly generated outputs (evaluating M on points, e.g., $M(x)$ for 8000 points $x \in \mathcal{X}$)

Scalability Challenge: for ReLU activations, this problem is NP-complete

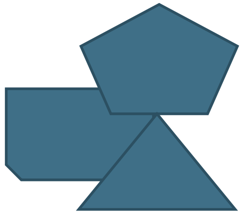
Intuition: number of polytopes may grow exponentially in number of ReLUs due to case splitting

ReLU (Rectified Linear Unit) Neural Network



$f(x) = \max(0, x)$

Input set:



Union of polytopes

$$\mathcal{X}^{[0]} = \bigcup_{s=1}^{N_0} \mathcal{X}_s^{[0]}$$

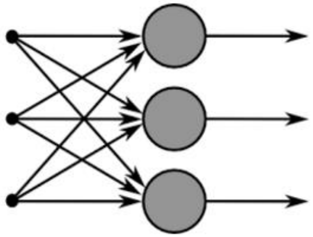
$$\mathcal{X}_s^{[0]} \triangleq \left\{ \mathbf{x}^{[0]} \mid \mathbf{A}_s^{[0]} \mathbf{x}^{[0]} \leq \mathbf{b}_s^{[0]},\ \mathbf{x} \in \mathbb{R}^{n^{[0]}} \right\}$$

For single neuron:

$$y_j = f\left(\sum_{i=1}^{n} \omega_i x_i + \theta_i\right) = \max\left(0, \sum_{i=1}^{n} \omega_i x_i + \theta_i\right)$$

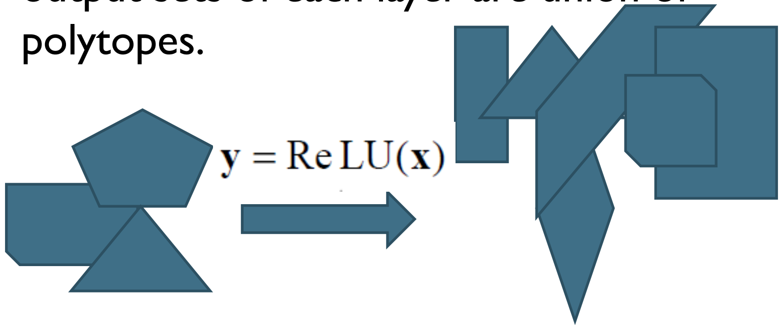
For single layer:

$\mathbf{x}$



$\mathbf{y} = \max(0, \mathbf{W}\mathbf{x} + \boldsymbol{\theta})$

Theorem: For ReLU neural networks, if input set is a union of polytopes, then output sets of each layer are union of polytopes.



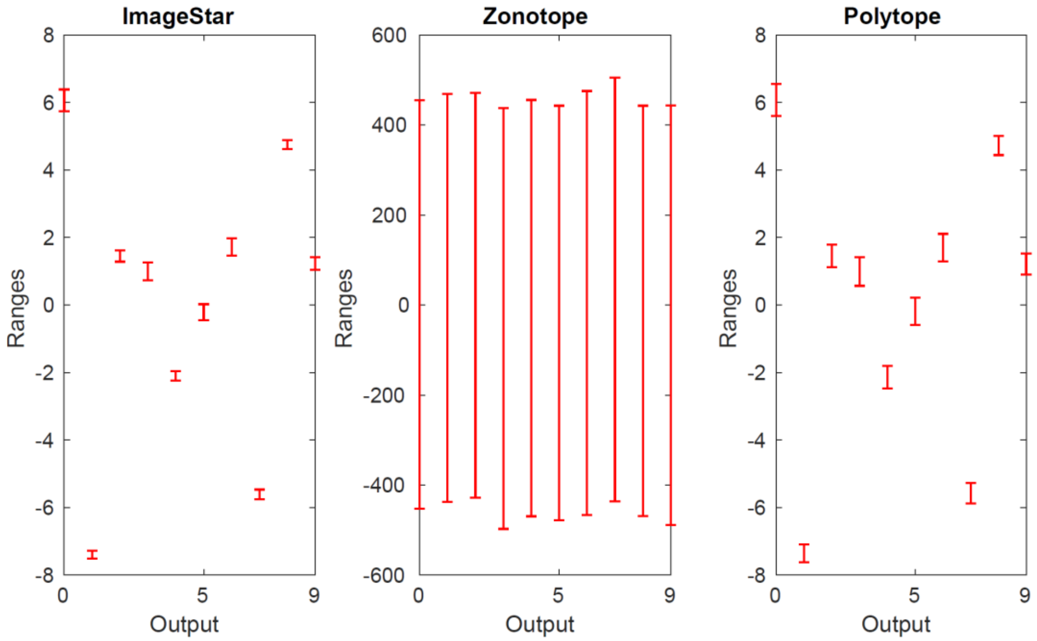
Union of polytopes

We can compute layer-by-layer.

MNIST Robustness Verification: Comparison of Set Representations

Hoang-Dung Tran

http://yann.lecun.com/exdb/mnist/

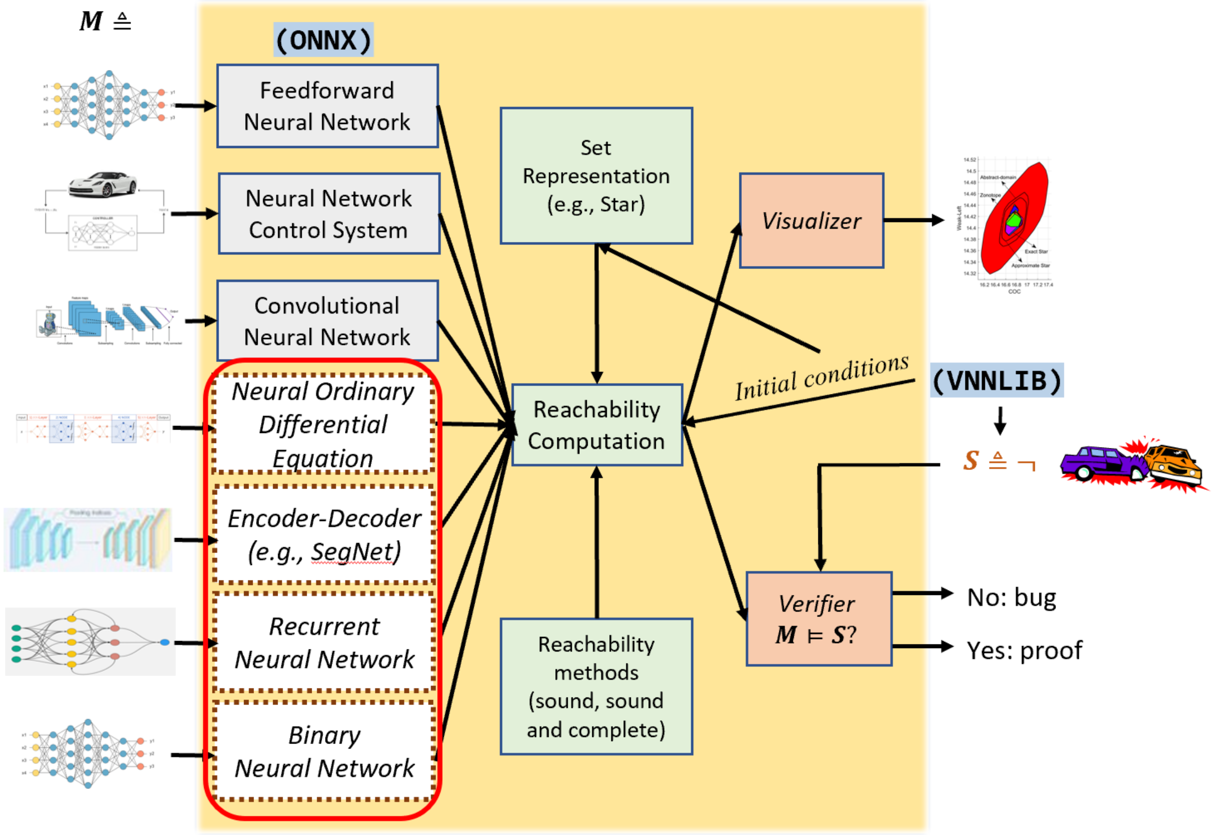


- MNIST classifier is a function from images to classes,M: $\mathbb{R}^{28\times 28} \mapsto \{0,\ldots,9\}$
- Input: $\mathbb{R}^{28\times 28}$; input set: a convex subset in $\mathbb{R}^{28\times 28}$
- Output prior to softmax/argmax: $\mathbb{R}^{10}$; output set: shape in $\mathbb{R}^{10}$
- Final output: take argmax over these 10 dimensions, this is the identified class
- ImageStar: efficient and accurate set representation developed for NNV, extension of star sets for images
- Zonotopes: symmetric data structure, leads to greater imprecision
- Polytope: better accuracy than zonotope, but worse than ImageStars

[Tran et al, “Verification of Deep Convolutional Neural Networks Using ImageStars,” **CAV’20**]

NNV 2.0

https://github.com/verivital/nnv



[Manzanas Lopez et al, "Verification of Neural Network Compression of ACAS Xu Lookup Tables with Star Set Reachability", **AIAA'21**]
[Xiang et al, "Reachable Set Estimation for Neural Network Control Systems: A Simulation-Guided Approach", **TNNLS'21**]
[Tran et al, "Robustness Verification of Semantic Segmentation Neural Networks using Relaxed Reachability", **CAV'21**]
[Tran et al, "Verification of Piecewise Deep Neural Networks: A Star Set Approach with Zonotope Pre-filter", **FAOC'21**]
[Manzanas Lopez et al, "Reachability Analysis of a General Class of Neural Ordinary Differential Equations", **FORMATS'22**]
[Manzanas Lopez et al, "Evaluation of Neural Network Verification Methods for Air-to-Air Collision Avoidance", **JAT'22**]
[Tran et al, "Verification of Recurrent Neural Networks using Star Reachability", **HSCC'23**]
[Ivashchenko et al, "Verifying Binary Neural Networks on Continuous Input Space using Star Reachability", **FormaliSE'23**]
*[Manzanas Lopez et al, "NNV 2.0: The Neural Network Verification Tool", **CAV'23**]*

<u>Legend</u>

- NNV
- **NNV 2.0 additions**

Features

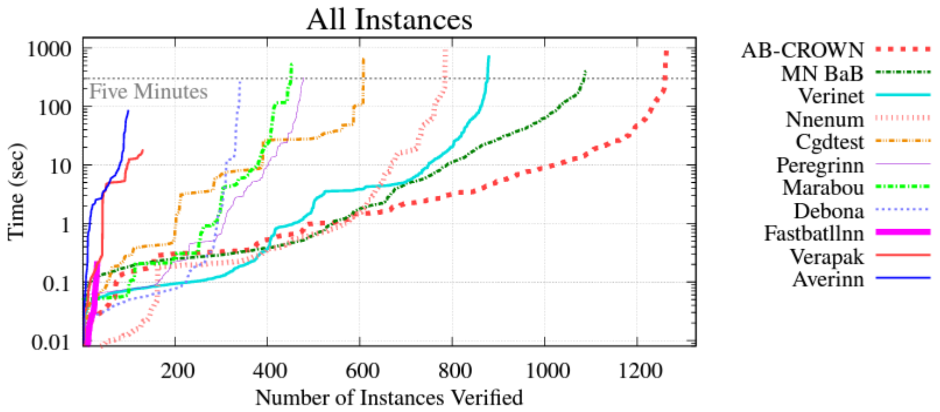
Feature	Supported
Neural Network type	Feedforward, Convolutional, **Encoder-Decoder, Recurrent and Binary Neural Networks, Neural Ordinary Differential Equations**
Layers	MaxPool, Conv, BatchNorm, AvgPool, FullyConnected, **MaxUnpool, Transposed Conv, Dilated Conv, NODE, Recurrent, Sign**
Activation functions	ReLU, Satlin, Sigmoid, Tanh, **Leaky ReLU, Satlins**
Plant dynamics (NNCS)	Linear ODE, Nonlinear ODE, **Hybrid Automata**, Continuous & Discrete Time
Set Representation	Polyhedron, Zonotope, Star, ImageStar
Reachability methods	*Sound and complete*: exact *Sound*: approx, abs-dom, **relax-range, relax-area, relax-random, relax-bound**
Reachable set visualization	Yes, exact and over-approximation
Verification	Safety, robustness, **VNNLIB**
Miscellaneous	Parallel computing, counterexample generation, **ONNX***

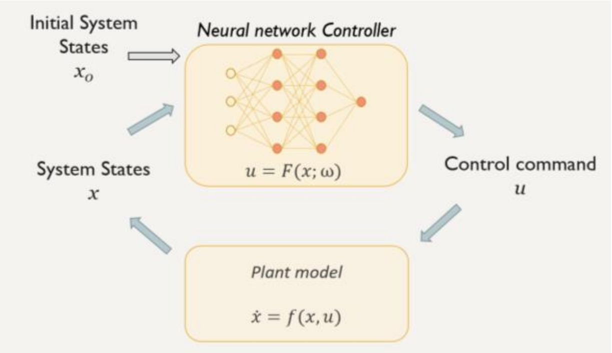
*ONNX support has been improved and extended to other NN types.

Related Work

- NN verification
  - Approaches
    - SMT, MILP, Reachability...
  - Tools
    - α,β-CROWN, MN BaB, Verinet, nnenum, cdgtest
    - Peregrinnm Marabou, Debona, Fastballnn
    - Reluplex, DLV, ReluVal, ERAN, Venus, OVAL
    - DNNF, RPM, NV.jl, MIPVerify, Verapak, Averinn
  - Competition
    - VNN-COMP (participant 2020, 2021, 2023)
- Neural Network Control System (NNCS) verification
  - Also referred to as "Neural Feedback Loops"
  - Linear vs Nonlinear
  - Continuous vs discrete-time
  - Friendly Competition
    - ARCH-COMP AINNCS (participant 2019, 2020, 2021, 2022, 2023)
  - Tools
    - CORA, JuliaReach, Verisig, ReachNN*, POLAR, OVERT, VenMAS, Sherlock
    - RINO, NFL_veripy, DeepNNC, SMC, AutomatedReach

[MN Müller et al, VNN-COMP 2022]





NNCS

(P.S. Apologies if we missed your tool, please come talk to us after the talk and we'll fix it for the next one)

Generalized Star Sets

Hoang-Dung Tran

- **Generalized star set $\mathbf{\Theta} = < \mathbf{c}, \mathbf{V}, \mathbf{P} >$**
  - $\Theta = \{x | x = c + \Sigma_{i=1}^{m} \alpha_i v_i,\ P(\alpha)\}$
  - $c \in R^n$ is the ***center***
  - $V = \{v_1, v_2, \cdots, v_m\}$ is a set of ***basis vectors***
  - $P(\alpha) \triangleq \boldsymbol{C\alpha} \leq \boldsymbol{d}$, is a ***predicate***
  - $\alpha = [\alpha_1, \alpha_2, \cdots, \alpha_m]^T$, is ***predicate variable***
- **Properties**
  - Any bounded convex polyhedron can be represented as a star
  - Affine mapping of a star is also a star

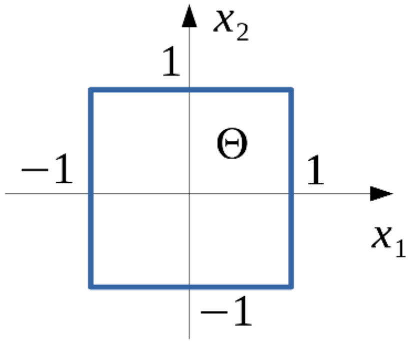
$$x \in \Theta = < \mathrm{c}, \mathrm{V}, \mathrm{P} >, \qquad y = Wx + b \ \rightarrow \mathbf{y} \in \bar{\mathbf{\Theta}} = < \boldsymbol{Wc} + \boldsymbol{b}, \boldsymbol{WV}, \boldsymbol{P} >$$

  - Intersection of a star with a half-space is also a star

$$x \in \Theta = < \mathrm{c}, \mathrm{V}, \mathrm{P} >$$

$$H \triangleq \{x | Gx \leq g\} \ \rightarrow \mathbf{\Theta} \cap \boldsymbol{H} = < \boldsymbol{c}, \boldsymbol{V}, \boldsymbol{P} \wedge \boldsymbol{P'} >$$

$$P'(\alpha) = (G \times V)\, \alpha \leq g - G \times c$$



$$\Theta = \{x \mid x = c + V\alpha, C\alpha \leq d, \alpha = (\alpha_1\ \alpha_2)^T\}$$

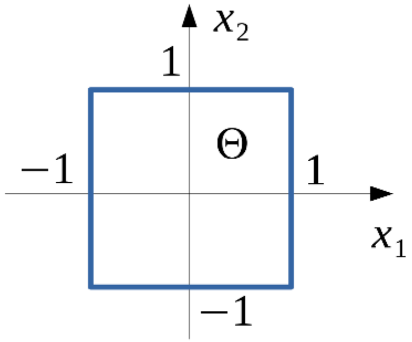
$$c = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, V = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, d = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

[Tran et al, “Star-Based Reachability Analysis of Deep Neural Networks,” **FM’19**]

Generalized Star Sets

Hoang-Dung Tran

- **Why are star sets suitable for reachability analysis of NNs?**
  - Star set is **efficient in affine mapping and intersection with half-spaces**
  - Other argument: very effective in verification of high-dimensional $\mathbb{R}^{10^9}$ hybrid systems: [Bak S, Tran H-D, Johnson TT, “Numerical Verification of Affine Systems With Up to a Billion Dimensions,” **HSCC’19**]; Hylaa tool…
- How do we use these?
  - This is a data structure used to represent **infinite sets of inputs**: considers a symbolic sets of inputs (e.g., an infinite number of images simultaneously) as opposed to testing/evaluating on individual inputs (points)
  - Can lead to scalability improvements vs. trying all inputs (Monte Carlo vs. reachability argument)
- Extended to images efficiently with ImageStars [CAV’21]



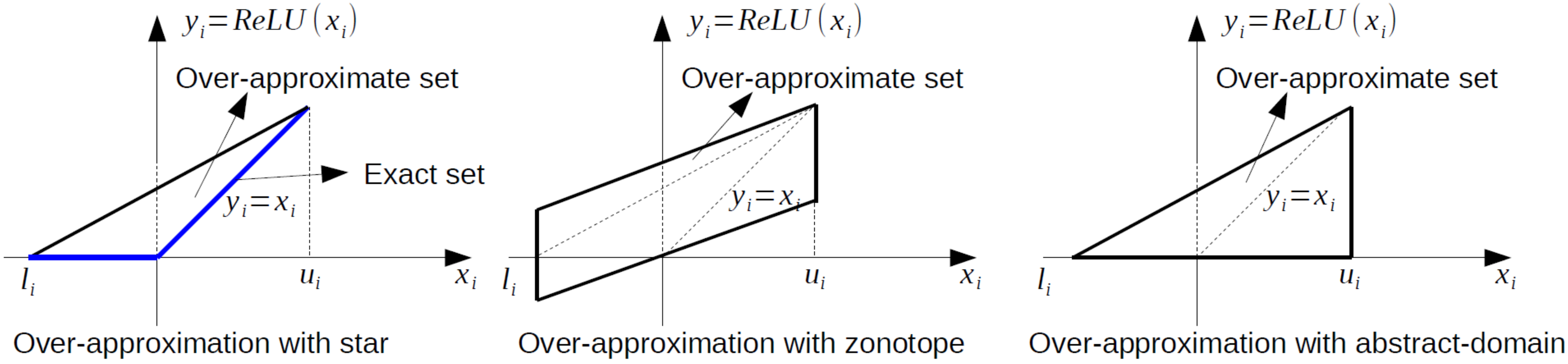
$$\Theta = \{x \mid x = c + V\alpha, C\alpha \le d, \alpha = (\alpha_1\ \alpha_2)^T\}$$

$$c = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, V = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, d = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Symbolic State Space Representation: Star Sets vs. Zonotopes, Part I


Hoang-Dung Tran

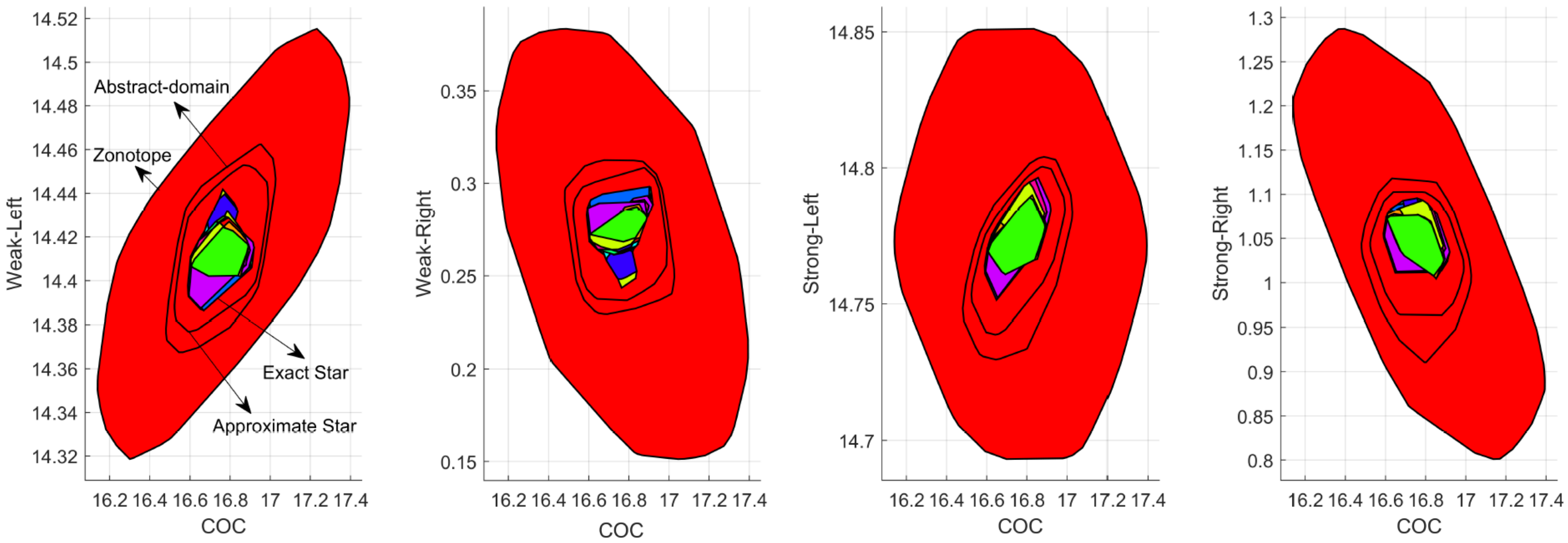
Star set symbolic representation of reachable states in our NNV tool allows improvements varying from 10x to 10,000x speedup vs. existing methods (Reluplex, DeepZ, DeepPoly, ReluVal…) with less conservatism than other overapproximative methods



Symbolic State Space Representation: Star Sets vs. Zonotopes, Part II

Hoang-Dung Tran

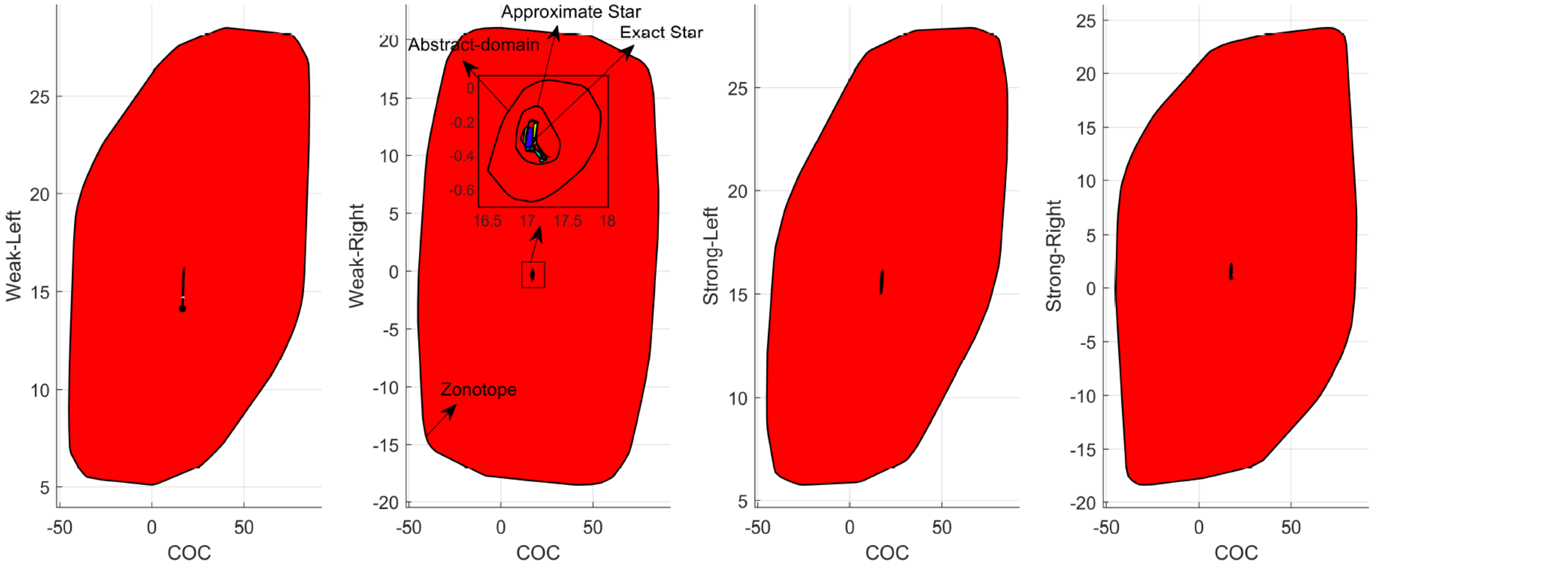
Illustration of overapproximation conservativeness for different symbolic state-space representations (zonotopes, abstract domains, approximate star sets, and exact star sets) within an ACAS Xu benchmark, illustrating the accuracy provided by star set representations, as they are the smallest sets



Symbolic State Space Representation: Star Sets vs. Zonotopes, Part III

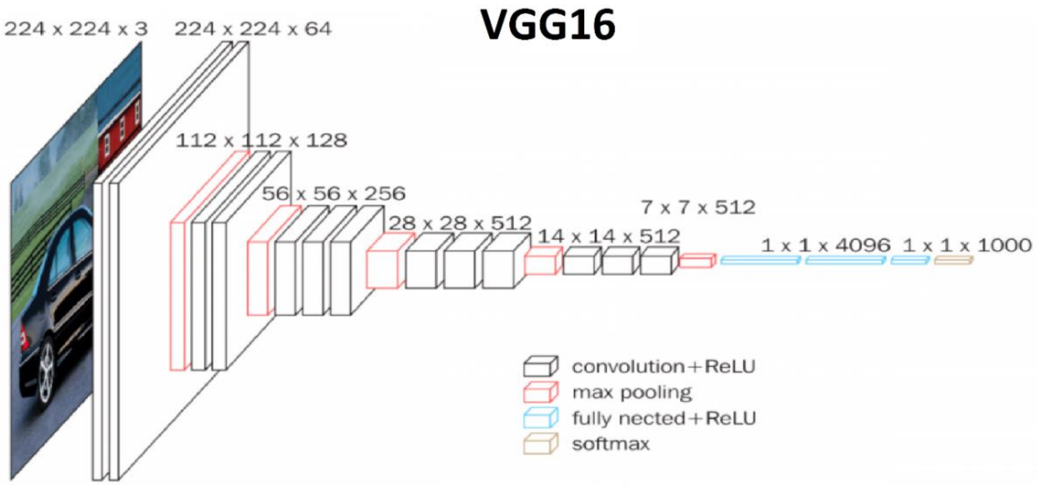
Hoang-Dung Tran

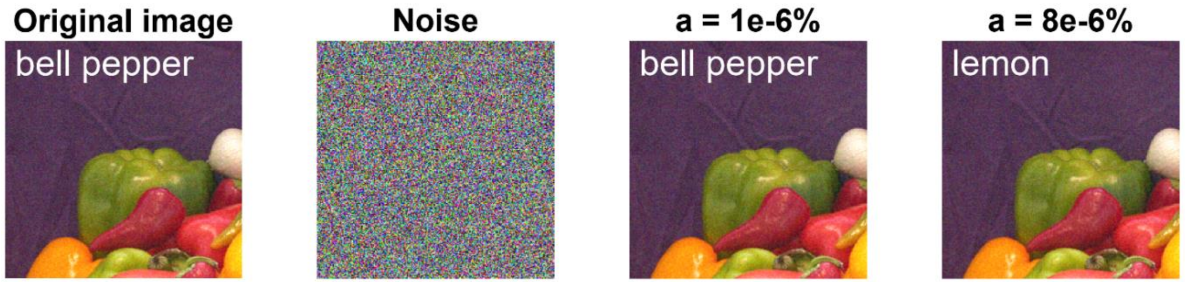
Because star sets minimize overapproximation error, properties may be efficiently verified with them vs. other symbolic state space representations that are too imprecise (zonotopes, abstract domains, polytopes, intervals, etc. as used in DeepZ, DeepPoly, ReluVal…)

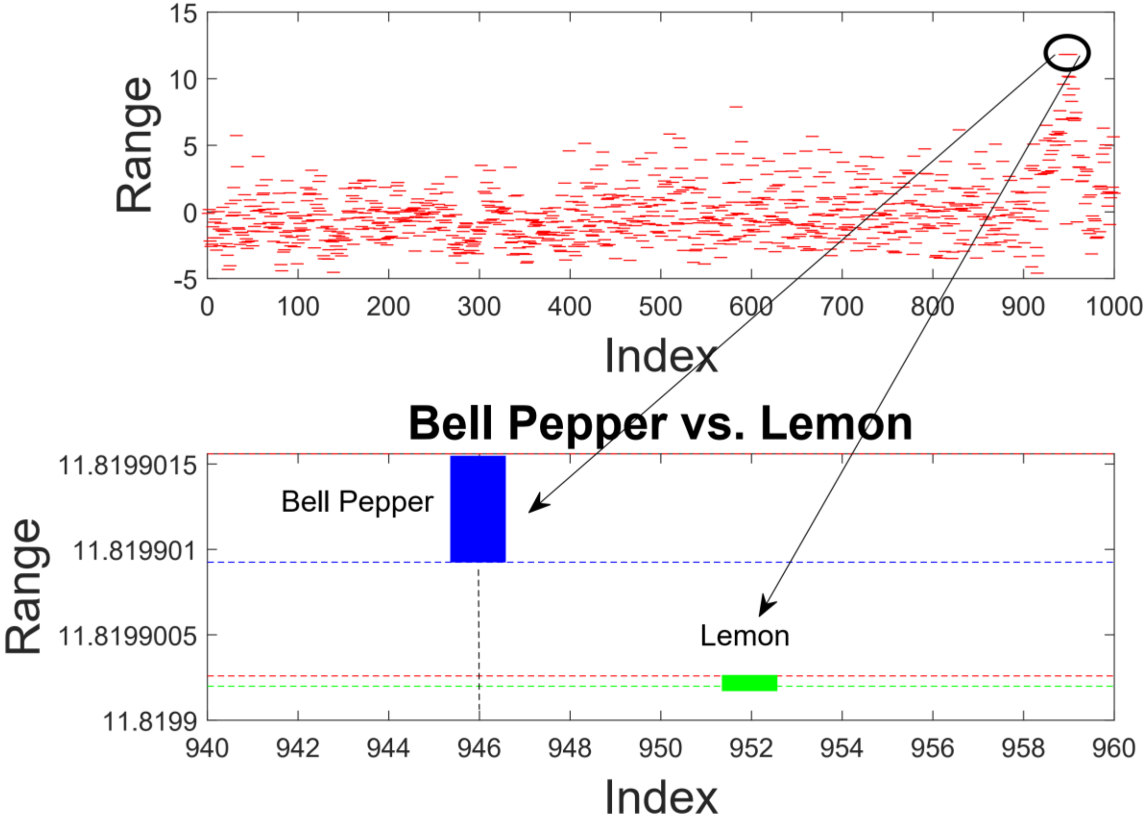


VGG16 Robustness Verification Example

Hoang-Dung Tran







Disturbed images = Original image + *a* * Noise; note a is a set

Is VGG16 robust to an FGSM attack for $a \leq 2 \times 10^{-8}$?

[Tran et al, “Verification of Deep Convolutional Neural Networks Using ImageStars,” **CAV’20**]

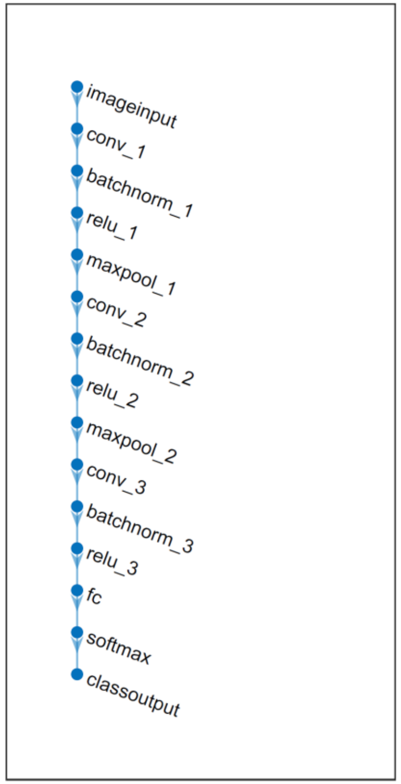
MNIST Robustness Verification across Dataset

Hoang-Dung Tran

- How do we use this?
- Local robustness: with respect to inputs from the test data set
- Current best practice: evaluate across the test data set to provide a robustness measure in addition to accuracy (certified robust accuracy / CRA)

	Robustness Results (in Percent)					
	$\delta = 0.005$		$\delta = 0.01$		$\delta = 0.015$	
	Polytope	*ImageStar*	*Polytope*	*ImageStar*	*Polytope*	*ImageStar*
$d = 250$	85.00	86.00	83.00	86.00	82.00	86.00
$d = 245$	73.00	74.00	68.00	74.00	66.00	73.00
$d = 240$	68.00	69.00	63.00	68.00	59.00	67.00
	Verification Times (in Seconds)					
$d = 250$	6.33	7.80	9.23	19.10	13.97	41.29
$d = 245$	7.25	9.27	11.39	21.01	17.68	63.73
$d = 240$	8.75	11.31	16.32	35.63	26.53	116.04

MNIST_Small



98% accuracy

[Tran et al, “Verification of Deep Convolutional Neural Networks Using ImageStars,” **CAV’20**]

Semantic Segmentation Robustness

Hoang-Dung Tran

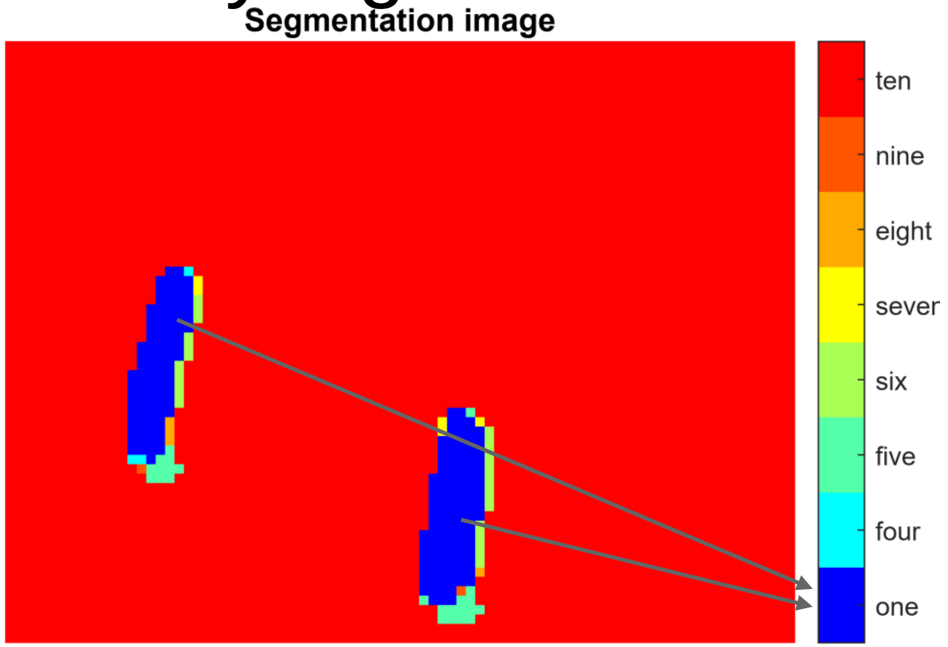
- Semantic segmentation
  - Input: image of width W pixels, height H pixels, and N color channels
  - Output: image specifying class $c \in C$ of every pixel
  - $F: \mathbb{R}^{W \times H \times N} \rightarrow C^{W \times H}$
- What is robustness for semantic segmentation?
  - Extended from classification robustness
  - For a small perturbation of an input image, does the class c of any pixel change (and if so, which ones and how many)?
  - Builds on set-based reachability analysis: again consider a set of input images as an ImageStar, and determine the output reachable set of F
    - More precisely: $F: \mathbb{R}^{W \times H \times N} \rightarrow \mathbb{R}^{W \times H \times C}$ and then take argmax (softmax) over C
  - Output space is significantly more complicated, as for classification, it was just C^1 (or more precisely, $\mathbb{R}^C$ prior to taking the argmax/softmax to determine the class)
  - Major challenge to overcome is handling this upsampling process from the latent space, accomplished e.g. through dilated/transposed convolutions/deconvolutions or unpooling layers

[Tran et al, "Robustness Verification of Semantic Segmentation Neural Networks using Relaxed Reachability," **CAV'21**]

M2NIST Semantic Segmentation Robustness Verification

Hoang-Dung Tran

- Multi-digit variant of MNIST
- Uses background as 10th class
- Segmentation mask defined by digits



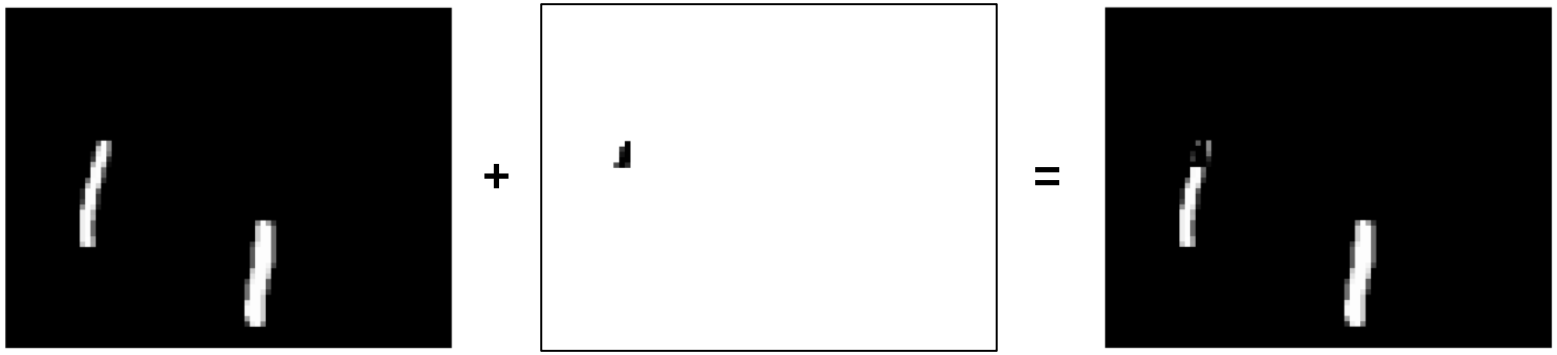
[Tran et al, "Robustness Verification of Semantic Segmentation Neural Networks using Relaxed Reachability," **CAV'21**]

M2NIST Semantic Segmentation Robustness: Attacks

Hoang-Dung Tran

- Simple adversarial attack
  - Brighten/darken some pixels of input image

input image *noise image* *attacked image*

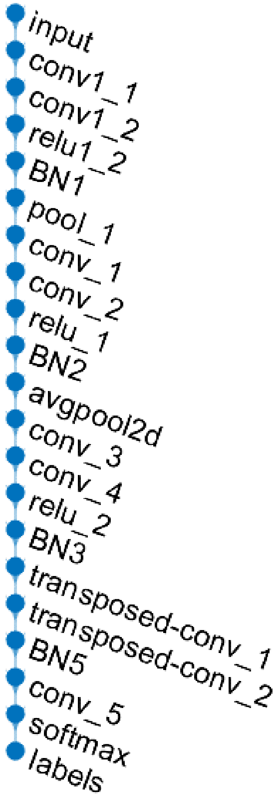


$$Attack\_image = input\_image + \alpha * noise\_image$$

$$\alpha_{min} \leq \alpha \leq \alpha_{max}$$

- Robustness questions
  - How many pixels are robust (correctly classified)?
  - How many pixels are not robust (incorrectly classified)?
  - How does the number of attacked pixels impact robustness?

Example net

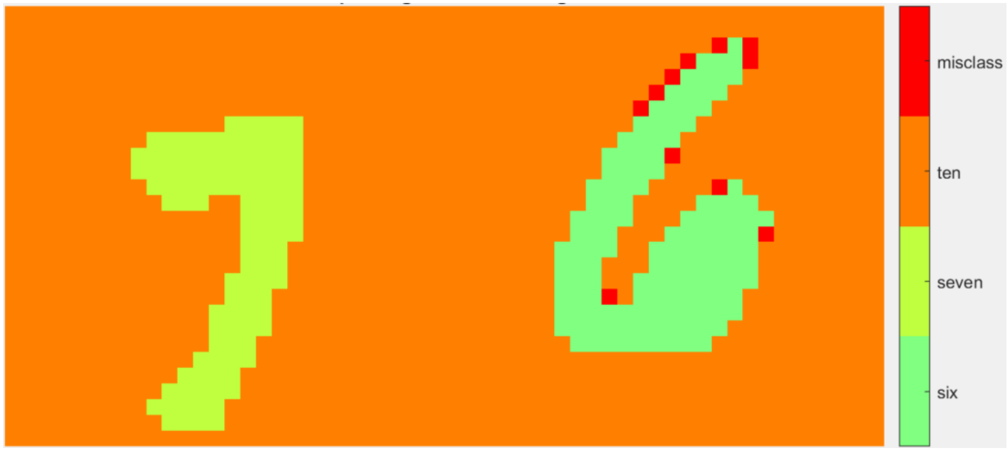


[Tran et al, “Robustness Verification of Semantic Segmentation Neural Networks using Relaxed Reachability,” **CAV’21**]

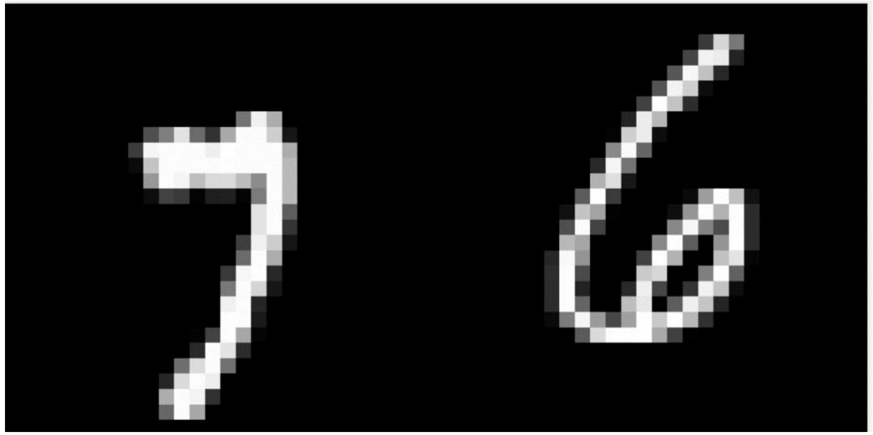
M2NIST Semantic Segmentation Robustness

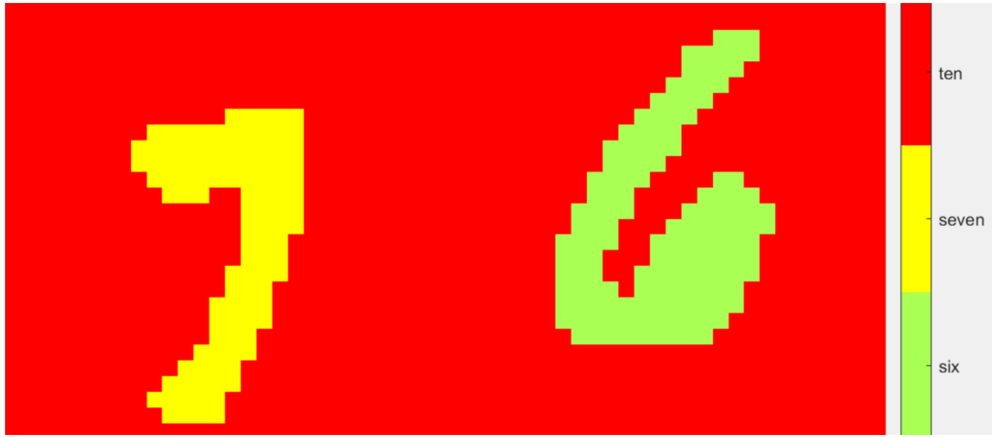
Hoang-Dung Tran

- 21-layer CNN, Mean IoU: 87.3%
- Illustration with 4 attacked pixels to brightness 0
- Runtime: ~20s
- 28x56x1 input image



Above: robustness with attack, below: without



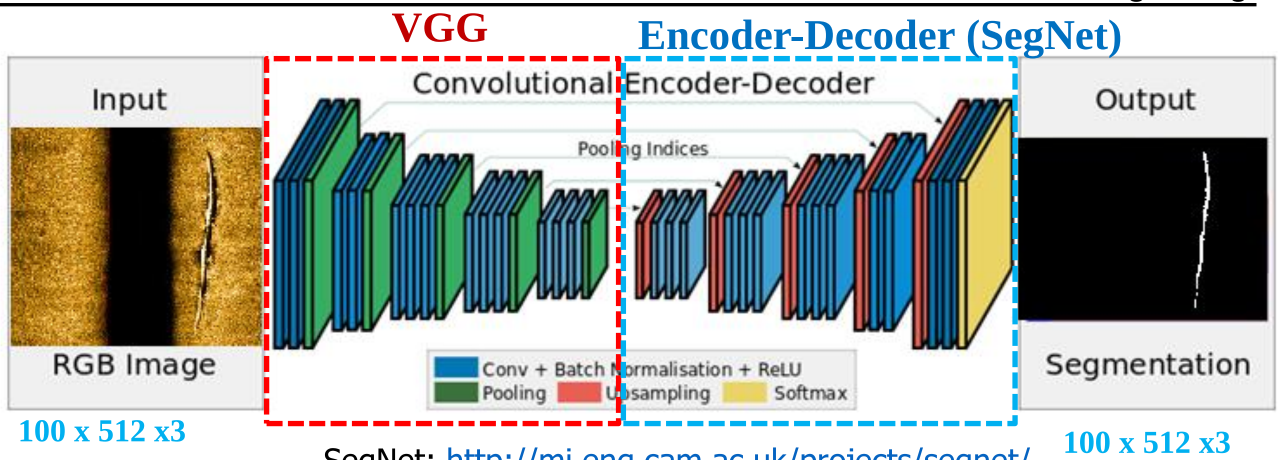


[Tran et al, “Robustness Verification of Semantic Segmentation Neural Networks using Relaxed Reachability,” **CAV’21**]

Back to AUV Semantic Segmentation

Hoang-Dung Tran

- Characterized robustness for pipe segmentation from sonar
- Transfer learning approach: trained on synthetic (and augmented) data set, analyzed on actual sonar data (transfer accuracy acceptable)



SegNet: http://mi.eng.cam.ac.uk/projects/segnet/

Synthetic

- Proved absence of adversarial perturbations of certain size across test data set

- I have to be a little vague here

NNV Use Cases

- Prove properties hold for neural networks, in terms of input-output specifications
  - Example: characterize local robustness for test data set, generating a metric similar to accuracy (robustness across test data set), for given perturbation levels
    - Provide robustness measure and accuracy measure with respect to test data set
    - E.g., MNIST classifier is 99% accurate and 95% robust for epsilon = 4/255 under l-infinity
  - Generate counterexamples (“adversarial examples/perturbations”) if not
- Prove properties hold for neural network control systems (e.g., usage of neural networks in autonomous CPS / AINNCS)
- Overall: provide assurance for machine learning components and their usage in autonomous CPS

Collins RUL VNN-COMP Benchmark

2022 VNN Competition (VNN-COMP)

Collins Aerospace

Benchmark proposal: Neural Network Based Remaining Useful Life Predictor

Executive summary: Collins Aerospace proposes a benchmark problem for the 2022 Competition on the Verification of Neural Networks (VNN). Current document contains problem background and the description of models and formal properties that are provided to the participants.

1. Background: Remaining Useful Life

Remaining Useful Life (RUL) is a widely used metric in Prognostics and Health Management (PHM) that manifests the remaining lifetime of a component (e.g., mechanical bearing, hydraulic pump, aircraft engine). RUL is used for Condition-Based Maintenance (CBM) to support aircraft maintenance and flight preparation. It contributes to such tasks as augmented manual inspection of components and scheduling of maintenance cycles for components, such as repair or replacement, thus moving from preventive maintenance to *predictive* maintenance (do maintenance only when needed, based on component's current condition and estimated future condition). This could allow to eliminate or to extend service operations and inspection periods, optimize component servicing (e.g., lubricant replacement), generate inspection and maintenance schedules, and obtain significant cost savings. RUL could also highlight areas for inspection during the next planned maintenance, i.e., it could be used to move up a maintenance/inspection action to prevent component failure. Finally, RUL function can also be used in airborne (in-flight) applications to dynamically inform pilots on the health state of aircraft components during flight.

https://github.com/loonwerks/vnncomp2022

Engaging with Collins and Mathworks on neural networks used for remaining useful life (RUL) estimation, in conjunction with Dr. Khaza Anuarul Hoque, Missouri

Simpler than perception tasks, but similar concerns, & clearer specifications: monotonic decrease, robustness, continuity, etc.

Special thanks to Dmitrii Kirov, Darren Cofer, Giacomo Gentile of Collins, and Akshay Rajhans & Amanjit Dulai of Mathworks

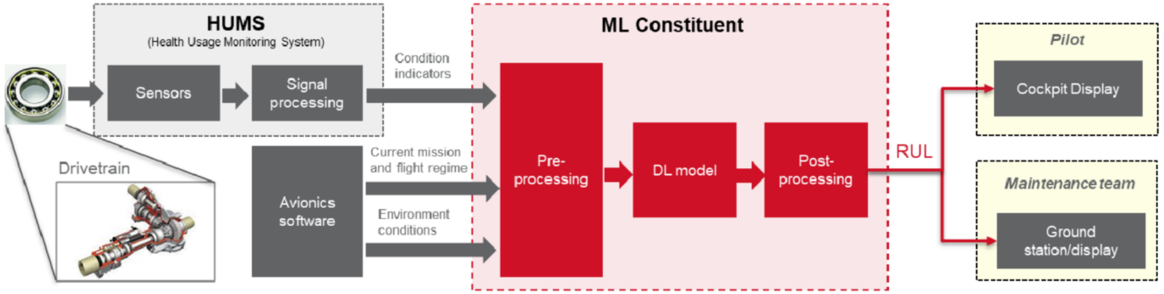


***Figure 1**. High-level overview of the ML Constituent for RUL estimation, and its operating environment*

Veritex: Neural Network Verification & Repair

Xiaodong Yang

- Veritex is a new tool for neural network verification and **repair**, developed in collaboration with Toyota, including both exact and overapproximative analysis methods; paper at FORMATS’22
- Uses several novel state-space representations, specifically facet vertex, facet-vertex incidence matrix (FVIM), and face lattice

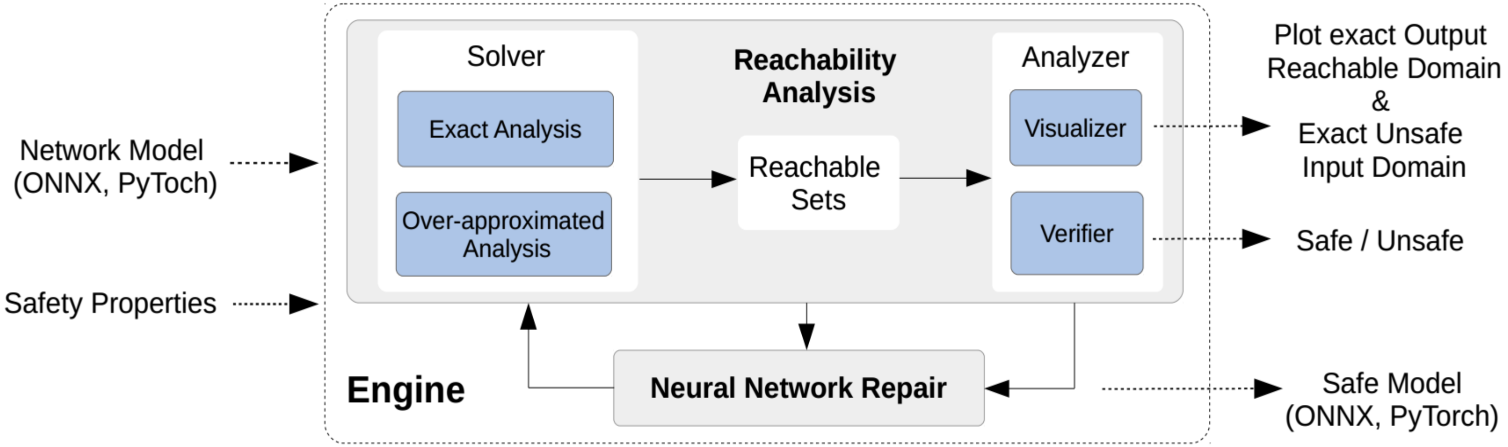


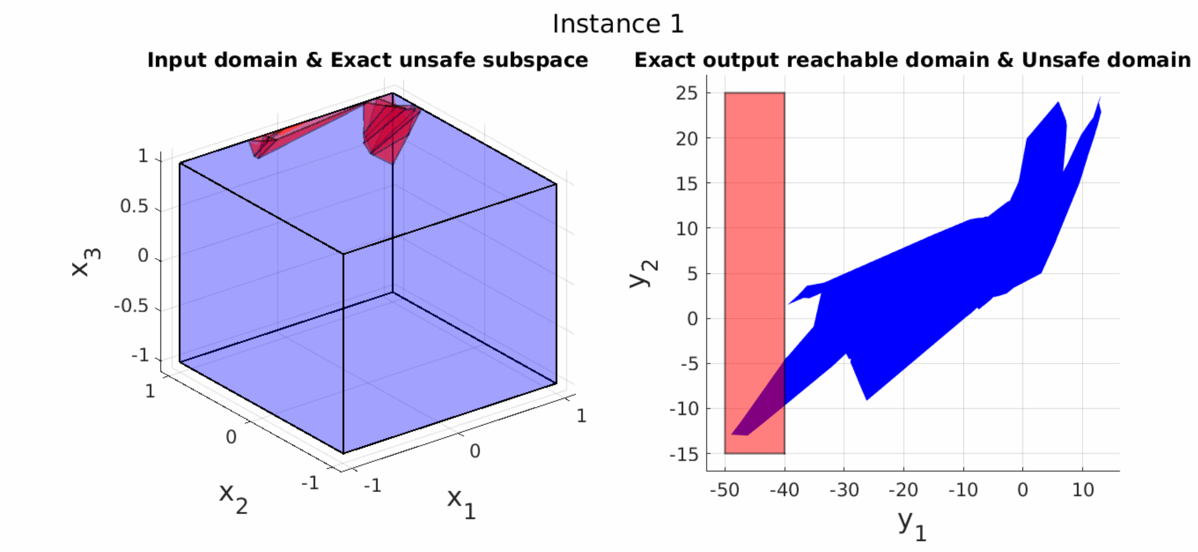
Figure: An overview of Veritex architecture. It is an object-oriented tool programmed in Python. (https://github.com/Shaddadi/veritex)

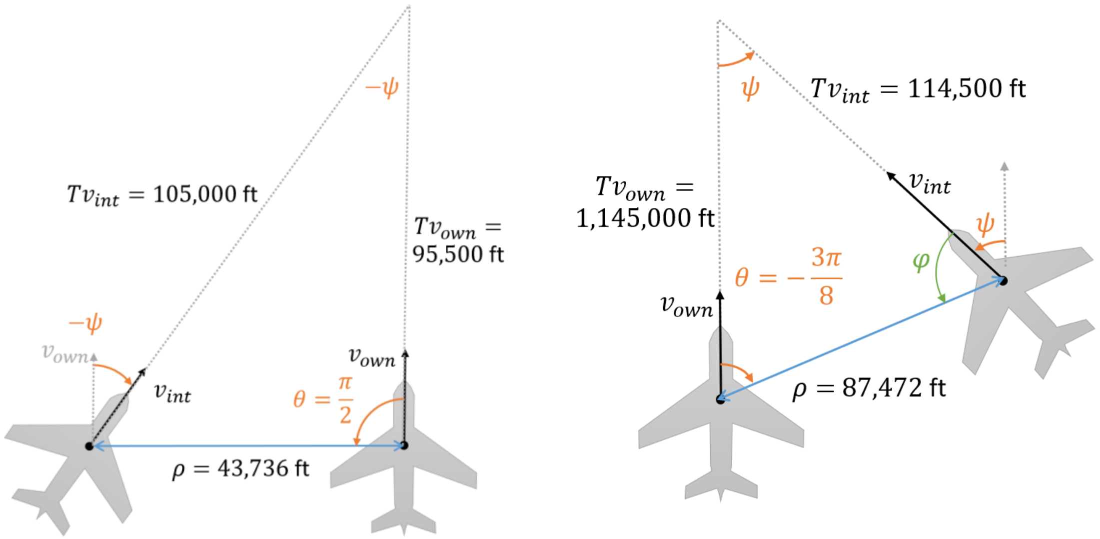
[Neural Network Repair with Reachability Analysis, Yang et al, FORMATS’22]

Veritex Counterexample Generation

Xiaodong Yang

- Illustration for an ACAS-Xu NN generating entire set of inputs that violate a specification, showing how changing the specification (red) by sliding it across the reachable set (blue) modifies the unsafe inputs (red, left)
- Relation to inverse of the NN from unsafe states
- Used in neural network repair framework [FORMATS'22]

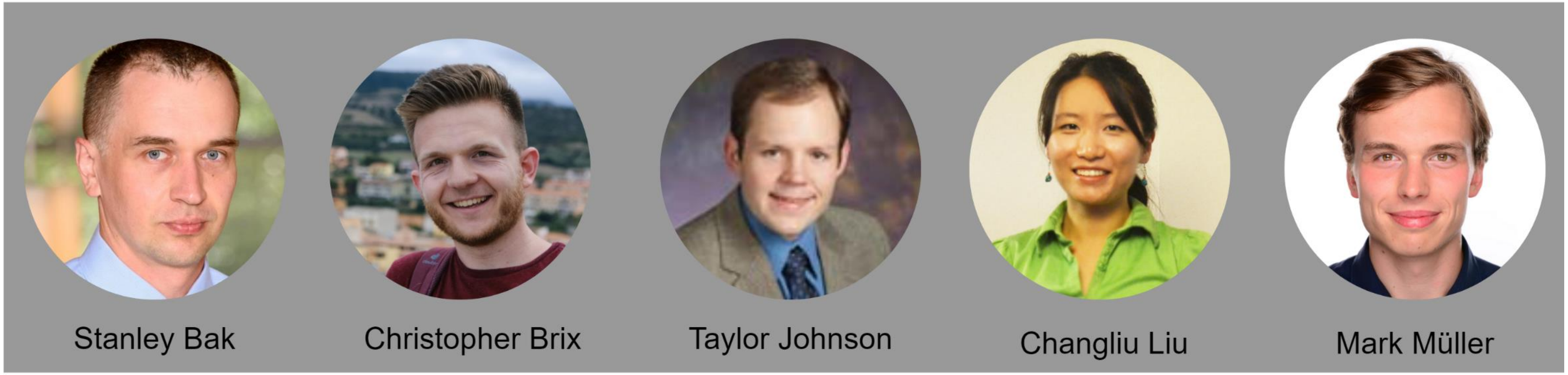




Conclusions

- Presented overview of NNV and neural network verification in the context of autonomous CPS that have safety-critical requirements
  - Other aspects not covered: out-of-distribution detection, conformal prediction, novel data detection, performance monitoring, details on runtime verification, data augmentation, …
  - Other ongoing projects not covered: LSTM/RNN/CNN verification for time-series data (with Collins/MathWorks/Toyota), neural ODE verification, physics-guided machine learning with PDEs, fuzz/coverage testing, closed-loop verification, neural network repair (with Toyota), safe reinforcement learning, symbolic (automata) learning, …
  - Technology transition: Toyota, Mathworks/Matlab neural network verification toolbox https://www.mathworks.com/products/deep-learning-verification-library.html , NSA malware classification, …
- Vibrant research community: ongoing activities like VNN-COMP and ARCH-COMP Artificial Intelligence / Neural Network Control Systems (AINNCS) categories, that we help organize
  - https://sites.google.com/view/vnn2024
  - https://cps-vo.org/group/ARCH/FriendlyCompetition
  - Recent **“NSF Workshop on Safety and Trust in Artificial Intelligence Enabled Systems"** at intersection of Formal Methods, Machine Learning, Safety, Trust, etc.
- Neural network verification is a subset of broader trustworthy AI community, covering aspects mostly of formal methods/verification, AI/ML, and security/privacy so far, but there are many other ongoing activities (FAccT, AIES, etc.)

5th International Competition on Verification of Neural Networks (VNN-COMP’24), co-located with CAV’24 in new **Symposium on AI Verification (SAIV’24)**



https://sites.google.com/view/vnn2024

https://www.aiverification.org/

2023 report: https://arxiv.org/abs/2312.16760
2020-2022 comparative report: https://arxiv.org/abs/2301.05815
2022 report: https://arxiv.org/abs/2212.10376
2021 report: https://arxiv.org/abs/2109.00498

2022 NSF Workshop on Safety and Trust in Artificial Intelligence (AI) Enabled Systems, Sept. 22-23, 2022

- There are major challenges everywhere in every domain: statistical learning (“2nd wave AI”) likely not going to cut it going forward, so need “3rd wave AI”, probably neurosymbolic
- What are the grand challenges in safety/trust in AI and autonomous systems?
- Served as input for creation of new $20M NSF Safe Learning-Enabled Systems program (23-562)

https://cps-vo.org/group/2022-NSFSafeTAI-Workshop

KEYNOTE SPEAKERS

DR. KATHLEEN FISHER assumed the role of Office Director for DARPA’s Information Innovation Office (I2O) in May 2022. In this position, she leads program managers in the development of programs, technologies, and capabilities to ensure information advantage for the United States and its allies, and coordinates this work across the Department of Defense and U.S. government.

DEBORAH RAJI is a Mozilla fellow and CS PhD student at University of California, Berkeley, who is interested in questions on algorithmic auditing and evaluation. In the past, she worked closely with the Algorithmic Justice League initiative to highlight bias in deployed AI products. She has also worked with Google’s Ethical AI team and been a research fellow at the Partnership on AI and AI Now Institute at New York University working on various projects to operationalize ethical considerations in ML engineering practice. Recently, she was named to Forbes 30 Under 30 and MIT Tech Review 35 Under 35 Innovators.

Kathleen Fisher, Keynote Address: ***Artificial Intelligence: Do you trust it?***

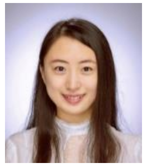
Abstract: We have seen significant progress in AI over the last ten years, predominantly driven by dramatic advances in machine learning and particularly deep learning. Society is realizing the benefits across a wide range of application domains. However, within the military, the consequence of making a wrong decision based on AI could be catastrophic. And the DoD must defend against nation-state level adversaries with significant resources, the ability to create deception, and the desire to change our way of life. DARPA is funding research in trustworthy AI technologies and systems that can be trusted to perform as expected despite the efforts of sophisticated adversaries. In this presentation, I will discuss research efforts in AI systems that we can trust with our (and warfighters') lives and explore fundamental advances beyond statistical ML that appear promising toward reaching the goal of trustworthy AI.

Inioluwa Deborah Raji, Keynote Address: ***On Audits, Algorithms and Accountability***

Abstract: As algorithmic deployments infiltrate our daily existence, it has become increasingly clear that in addition to the benefits they provide, these systems have also become a source of meaningful harm - significantly disrupting the lives of many real people. At the crux of these issues are biased and incorrect model outcomes that are hard to evaluate and stakeholders that are frustratingly difficult to hold accountable. As a result, policymakers and advocates are increasingly turning to audits as a method to accumulate concrete evidence for algorithmic harm and as a promising approach for accountability. Informed by important lessons from audit systems in other industries, this approach appears in many cases to be truly successful - some audits have already led to product updates or recalls, organizational changes and developments to regulation or standards. However, difficulties in execution, oversight and impact threaten the credibility and effectiveness of these audits as well as throw into question how much we can rely on this intervention without first investing in the technical, legal and institutional design of a more mature audit ecosystem for algorithmic deployments.

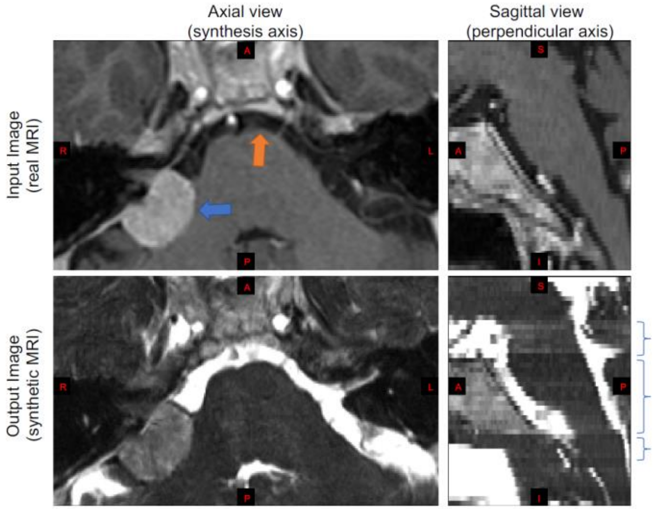
NSF FMitF: Track I: Generative Neural Network Verification in Medical Imaging Analysis

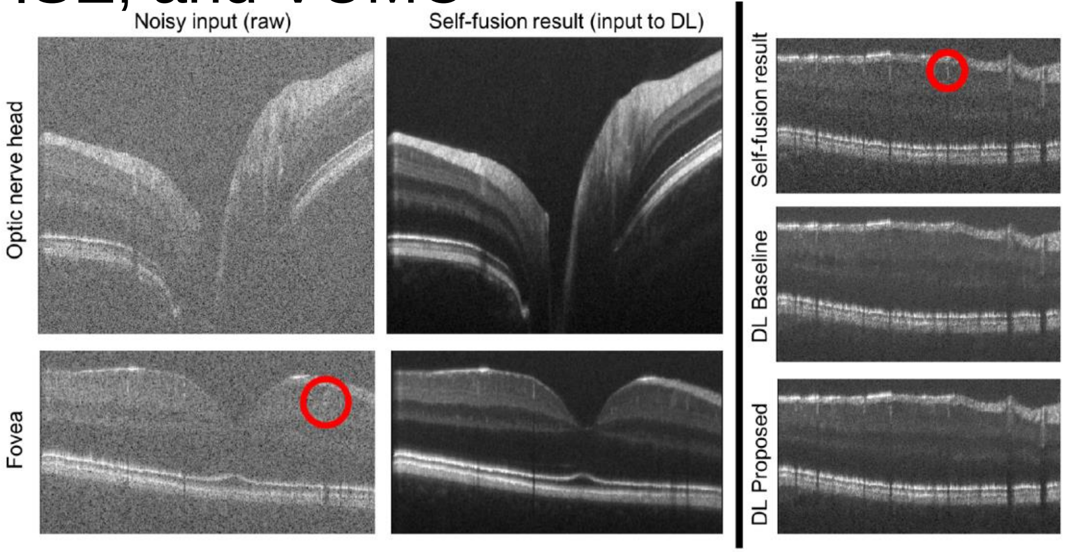

Ipek Oguz


Meiyi Ma

- DNNs, GANs, ... increasingly used to process medical data, including images (segmentation, denoising, synthesis, image reconstruction, …)
  - Major concerns about introduction of artifacts, etc. with generative models; less concerns about adversaries, but also to a degree
  - Project goals: develop ways to write specifications for generative models, define/scale verification for segmentation and image synthesis
- Collaboration between ISIS, VISE, and VUMC


Francesca Bagnato

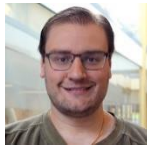





Kenny Tao

Vanderbilt Institute for Surgery & Engineering (VISE): https://www.vanderbilt.edu/vise/

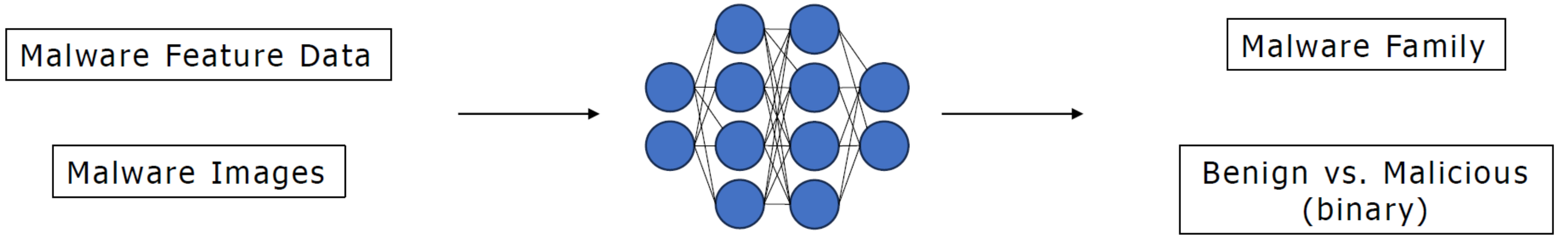
NSA SoS: Improving Malware Classifiers with Plausible Novel Samples

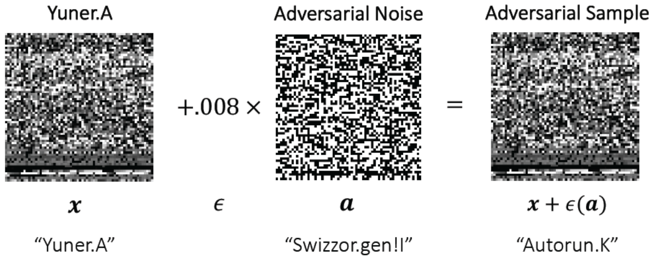

Kevin Leach


Preston Robinette

https://github.com/pkrobinette/verify_malware

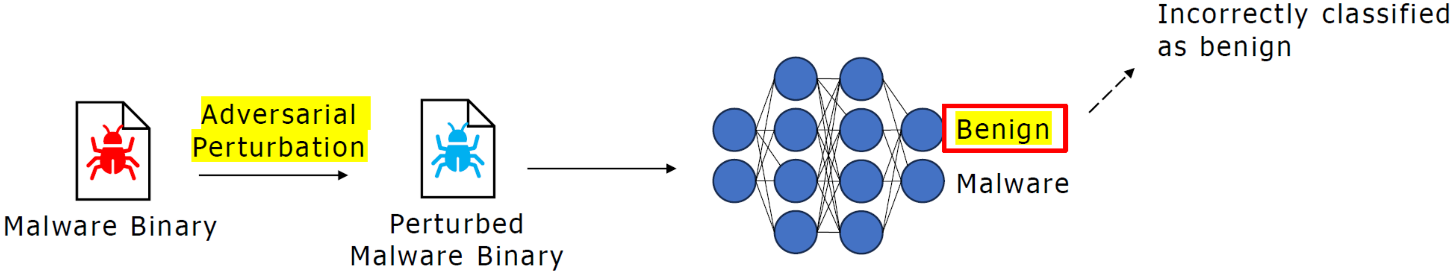
- Neural Networks are a popular means of classification:
  - Benign vs. malicious
  - Malware family





Metric	Model	Tool	Epsilon (ϵ) 1/255	2/255	3/255
CRA (%)	linear-25	NNV	85	83	79
		nnenum	90	86	82
	4-25	NNV	89	76	62
		nnenum	94	80	66
	16-25	NNV	88	82	67
		nnenum	90	86	64
Avg. Time (s)	linear-25	NNV	0.84	0.85	0.85
		nnenum	3.60	3.63	3.69
	4-25	NNV	17.75	41.66	82.18
		nnenum	11.59	10.80	11.13
	16-25	NNV	85.00	210.00	710.25
		nnenum	38.66	44.16	43.43

- Adversary can *perturb* input sample to cause **incorrect classification**



[Robinette et al, "Case Study: Neural Network Malware Detection Verification for Feature and Image Datasets," Formalise'24]
[Robinette et al, "Benchmark: Neural Network Malware Classification," AISoLA'23]

Perspective Summary

- Great time to research safe, secure, and trustworthy AI
- As we have seen, AI/ML being used in many safety/security-critical domains whether we like it or not, with some market-driven pullback already for (in part) safety issues (particularly in autonomous driving: Uber, Argo AI, Cruise, Tesla NHTSA actions, …)

- Did Phil see his shadow and impending AI winter (or just another told you so for dependability / formal methods)?

HOME OF
PUNXSY PHIL
PUNXSUTAWNEY
PHIL
DID NOT SEE
HIS SHADOW!
PUNXSUTAWNEY
PENNSYLVANIA
WEATHER CAPITAL
OF THE WORLD!
PHIL
TEAM
SPRING
2024
AP

VeriVITAL Members & Alumni

Current PhD Students & Postdocs


Judy Nguyen


Anne Tumlin


Samuel Sasaki


Preston Robinette
2021 NDSEG


Serena Serbinowska


Dr. Diego Manzanas Lopez


Dr. Tianshu Bao


Dr. Neelanjana Pal
MathWorks

Postdoc / PhD / Research Scientist Alumni


Dr. Nate Hamilton
2019 NDSEG
Parallax Research


Dr. Xiaodong Yang
Visa Research


Dr. Patrick Musau
Google


Prof. Weiming Xiang
Augusta University
2022 NSF CAREER


Prof. Hoang-Dung Tran
U Nebraska Lincoln
2021 IEEE TCCPS
Outstanding Dissertation


Prof. Joel Rosenfeld
USF
2021 AFOSR YIP


Prof. Luan Nguyen
U Dayton
2023 NSF CRII


Prof. Omar Beg
U Texas PB
2019 UT System
Rising STARs


Dr. Shafiul Chowdhury
Meta, ML Senior
Research Scientist


Prof. Khaza Hoque
U Missouri

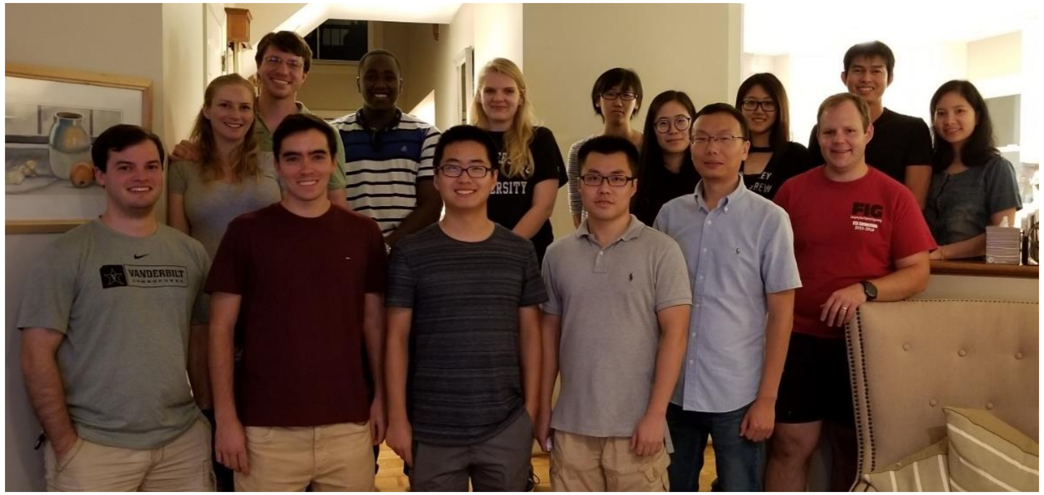

Dr. Andrew Sogokon
Southampton / Lancaster

MSc Thesis / Undergrad Researcher Alumni: at Google, Meta, Microsoft, Amazon, Qualcomm, Rivian, etc.

Thank You: Questions?

taylor.johnson@vanderbilt.edu
http://www.verivital.com/
Twitter: @taylorjohnson @verivital

VANDERBILT UNIVERSITY®



NVIDIA. TOYOTA NSF DARPA United Technologies MathWorks