Assured Autonomy

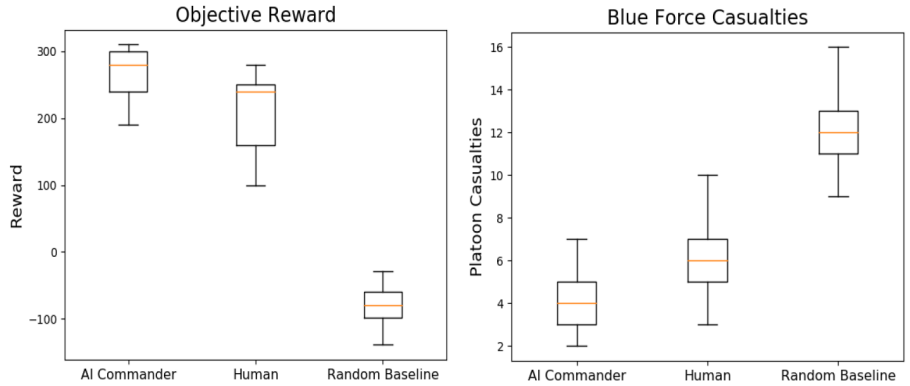
Sandeep Neema, I2O

Workshop on Advancements in T&E of Autonomous Systems

April 11, 2022

Impressive results in AI/ML in last decade …

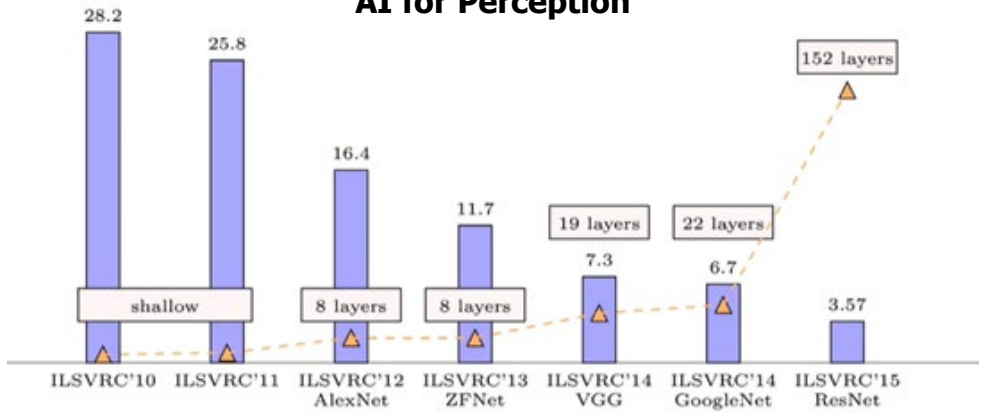
AI for Battle Management Command and Control (BMC2)



AI commander consistently outperforms human commander in a simulated mission in a gaming environment

Source: ARL

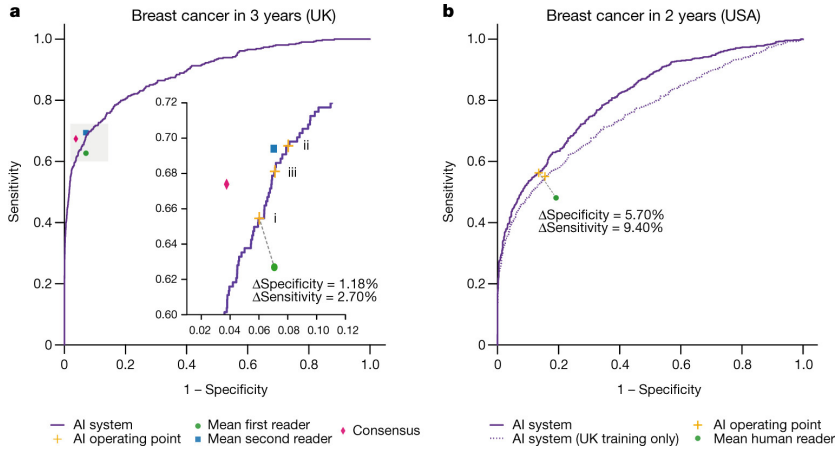
AI for Perception



Resnet (3.57% error & 152 layer network) outperforms humans performance (5% error)

Source: UC-Berkeley

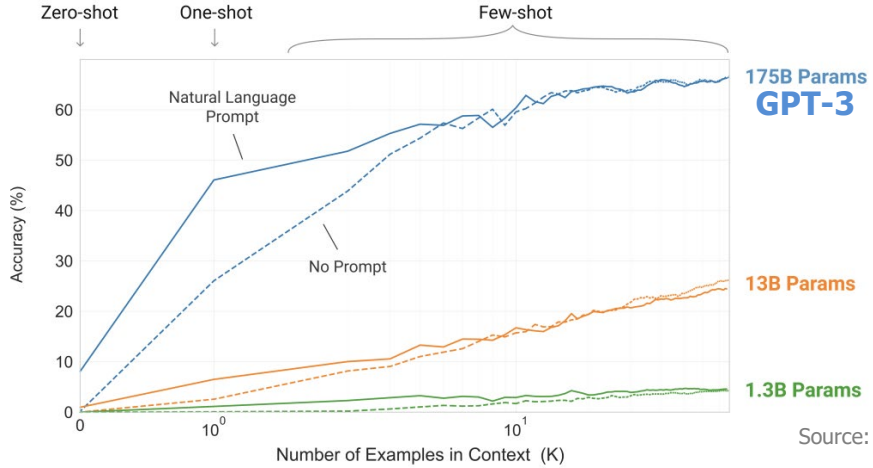
AI for Medical Diagnosis



Source: Google

Compared to human experts, AI showed reduction of 5.7% and 1.2% (USA and UK) in false positives and 9.4% and 2.7% in false negatives

AI for Natural Language Processing

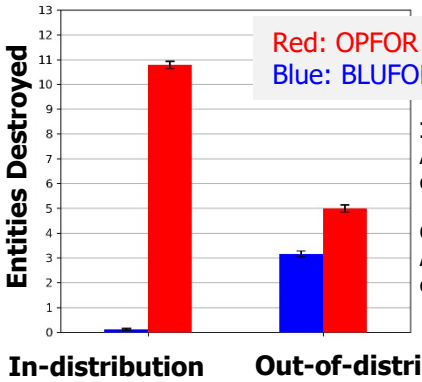


Source: OpenAI

GPT-3 (175B params) performs well in few-shot learning scenarios

But many fundamental challenges remain …

Lack of Adaptability/Generalizability

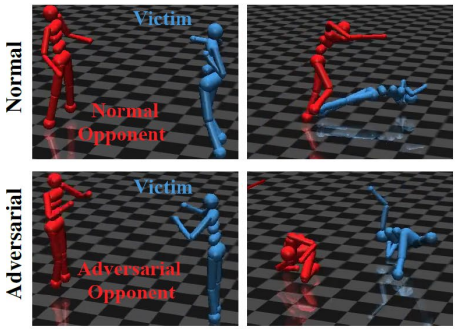


In-distribution:
AI commander performance when test data matches training set

Out-of-distribution:
AI commander performance when test data does not match the training set

Source: ARL

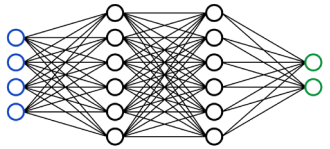
Lack of Robustness/Resilience



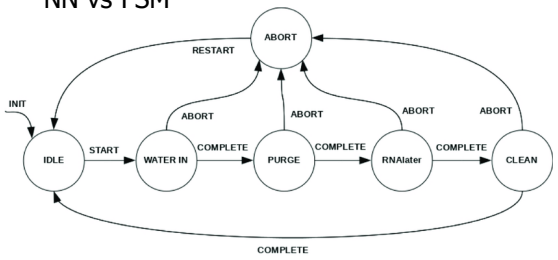
Source: UC-Berkeley

Adversarial Deep Reinforcement Learning
Adversary (in red) wins 86% of times by simply producing unexpected observations (falling down) for victim (in blue) in a "You Shall not Pass" game

Lack of Interpretability/Verifiability



NN vs FSM

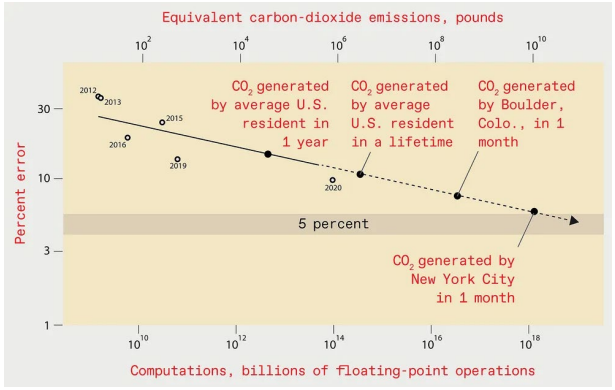


Some limitations are fundamental …

- Shafahi et al derived fundamental limits on robustness to adversarial attacks that cannot be escaped by any classifier
- Limits subject to properties of data-distribution and metrics of perturbation
- Implication: For a broad class of problems, such as complex high-dimensional image classification, adversarial examples are inescapable

"Are Adversarial Examples Inevitable?" Shafahi et al. (2018)

Increasing parameterization is unsustainable …

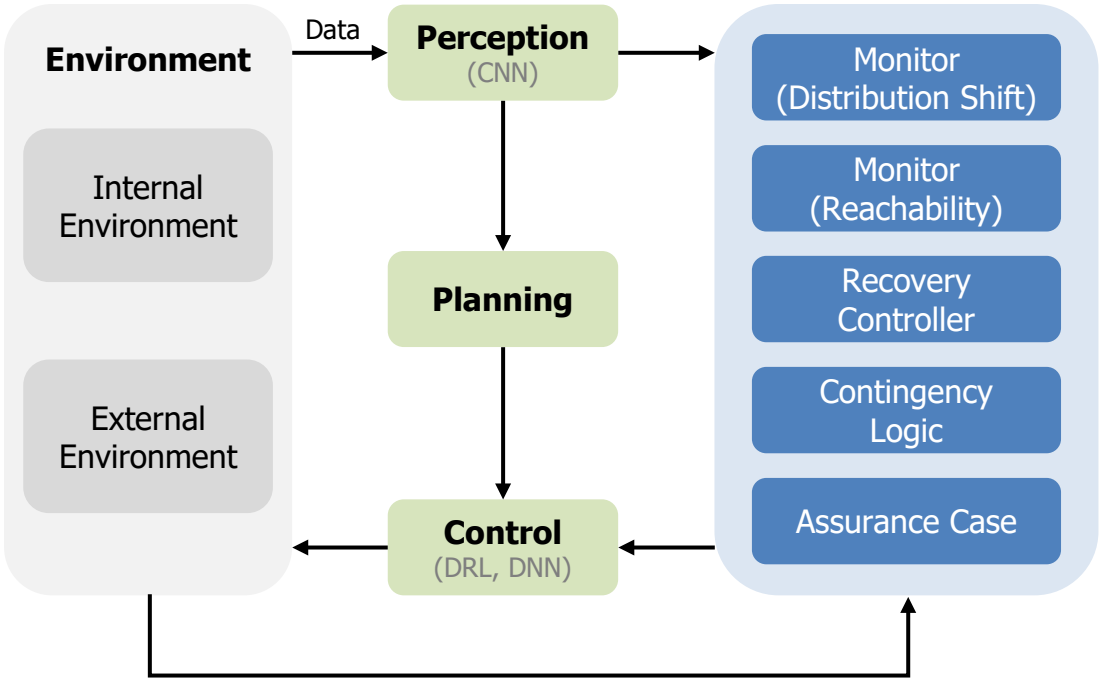


- Total computational cost scales with at least the fourth power of the improvement in performance, $\mathbf{k^4}$
- Achieving a 5% error rate (Top-1 error ~10% on ImageNet dataset) would require $\mathbf{10^{19}}$ billion floating-point operations
- GPT-3: **~3600 (Petaflop/s) days, ~$4.6 million**

"Deep Learning's Diminishing Returns", IEEE Spectrum, September 2021 part of special report on *The Great AI Reckoning* - Deep learning has built a brave new world—but now the cracks are showing

How to provide assurance for use in safety critical systems?

Constraining autonomy through a safety architecture



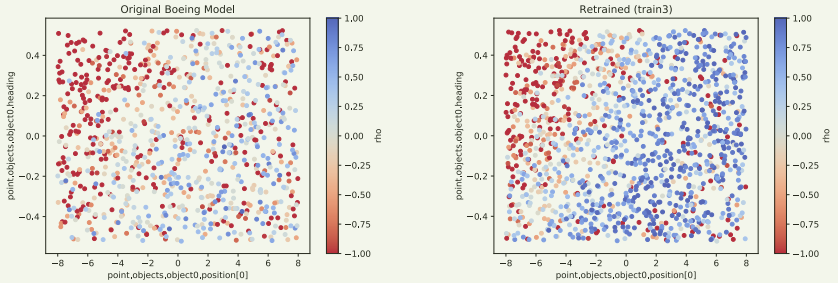
CNN: Convolution Neural Network
DRL: Deep Reinforcement Learning
DNN: Deep Neural Network

Assurance claim: (3) As long as reality stays within certain *bounds* of our model, the state of our autonomous system will stay within a certain *safety envelope*

(1) Assurance tools detect and diagnose bugs and minimize errors through **retraining**

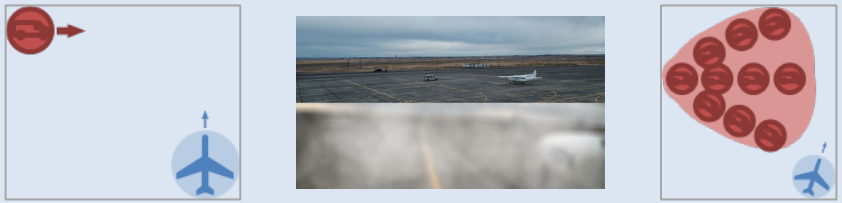
Original Model (satisfies property for **37%** of simulated scenarios)

Retrained Model (satisfies property for **62%** of simulated scenarios)



Source: Assured Autonomy

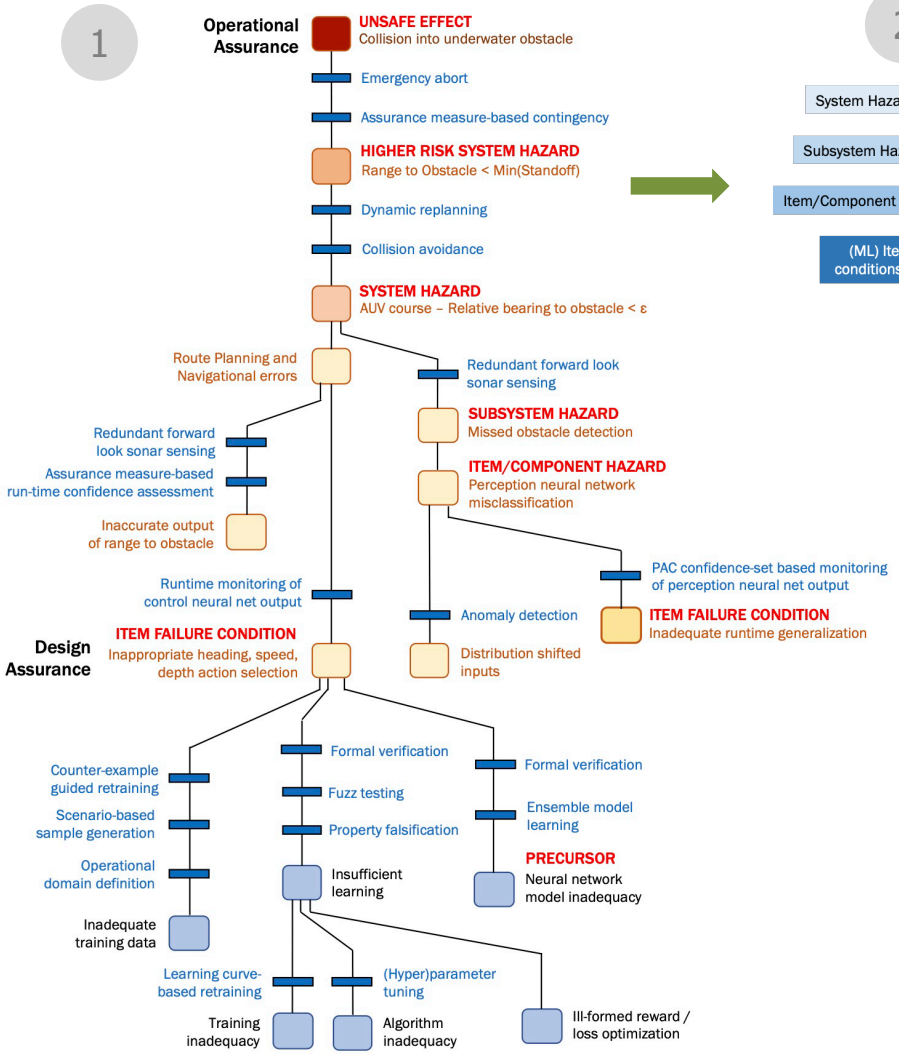
(2) Assurance monitors detect deviations from training and **recover** through contingency logic



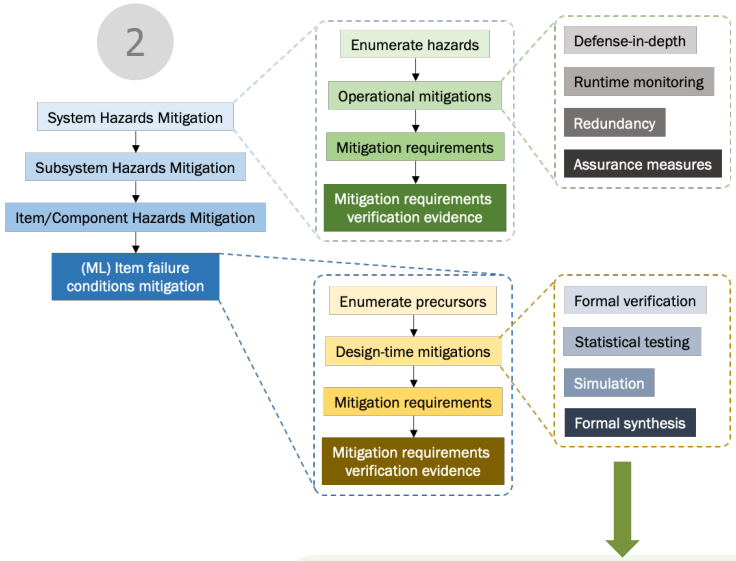
Source: Assured Autonomy

Assurance approach: linking ML to assurance case chains

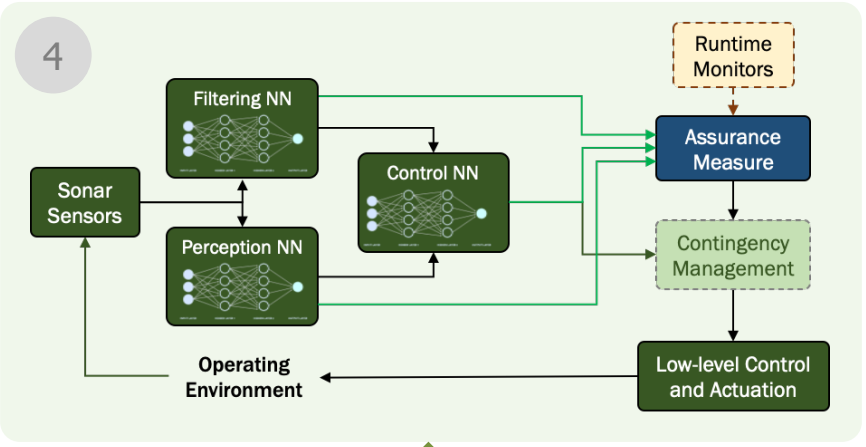
System Hazard Analysis



ML Assurance Argument



Maritime Domain Challenge Problem



3 Assurance Technologies

Design and Operation-Time Mitigation

Formal Verification & Simulation
- Reachability analysis (Vanderbilt - NNV)
- Hybrid model checking (U Penn – Verisig)
- Counter-example guided retraining (UCB-Scenic)
- Scenario-based sample generation (UCB-Scenic)
- Property falsification and fuzz testing (UCB-VerifAI)

Runtime Monitoring
- Perception neural net output novelty (Collins)
- Typicality-based anomaly detection (UCB)
- PAC-confidence set (U-Penn)
- Conformal prediction (Vanderbilt)
- Confidence auditing monitor (U-Penn)
- Assurance measure-based confidence (SGT)

Contingency Actions
- Assurance measure-based contingency (SGT)

Dynamic Replanning
- Safe plan (U Penn)

Technology Portfolio

TA1: Design time Assurance	Challenge	Formal Verification	Simulation-based	Test Synthesis	Monitor Synthesis
	Approach(es)	SMT solvers, LP solvers, Hybrid solvers, Theorem provers	Scenario description languages, toolchain	Manifold-based, Test coverage	Spec-based, Learning-based
	Performers	Collins (Stanford), VU, U. Penn, HRL (CMU), Imperial	UCB, VU	UCB, Collins (UMN)	Collins (Kestrel), VU, DOLL, Galois

TA2: Operation time Assurance	Challenge	Assurance Monitoring	Resilience and Recovery
	Approach(es)	Conformal prediction, Anomaly detection, Confidence estimation	Game theory, Simplex, Contingency logic
	Performers	VU, UCB, U. Penn	DOLL, Galois, Collins

TA3: Assurance Case	Challenge	Assurance Case Construction
	Performers	SGT, VU, Collins, U. Penn

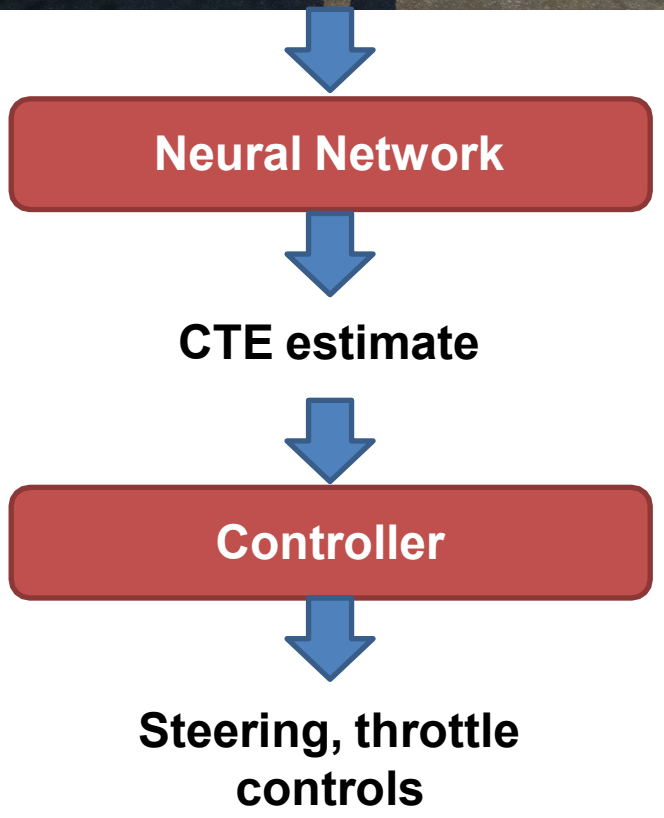
TA4: Platforms	Air Domain	Underwater Domain	Ground Domain
	Boeing	Northrop Grumman	CCDC-GVSC/HRL

Falsification & Counterexample guided retraining [Scenic & VerifAI]

TaxiNet

- Experimental autonomous aircraft taxiing system developed by Boeing
- Neural network uses camera image to estimate the *cross-track error*
  - CTE = distance from centerline
- System-level spec: plane must track centerline to within 1.5 meters

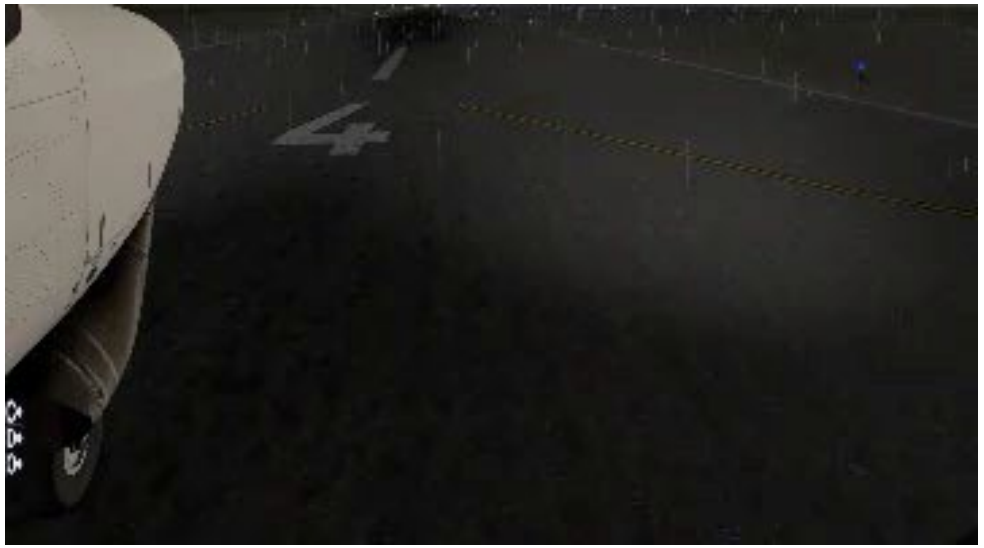
$$\varphi_{\text{eventually}} = \Diamond_{[0,10]} \Box (\text{CTE} \leq 1.5)$$



Modeling and Falsification

- Semantic features: time, clouds, rain, position/orientation of plane

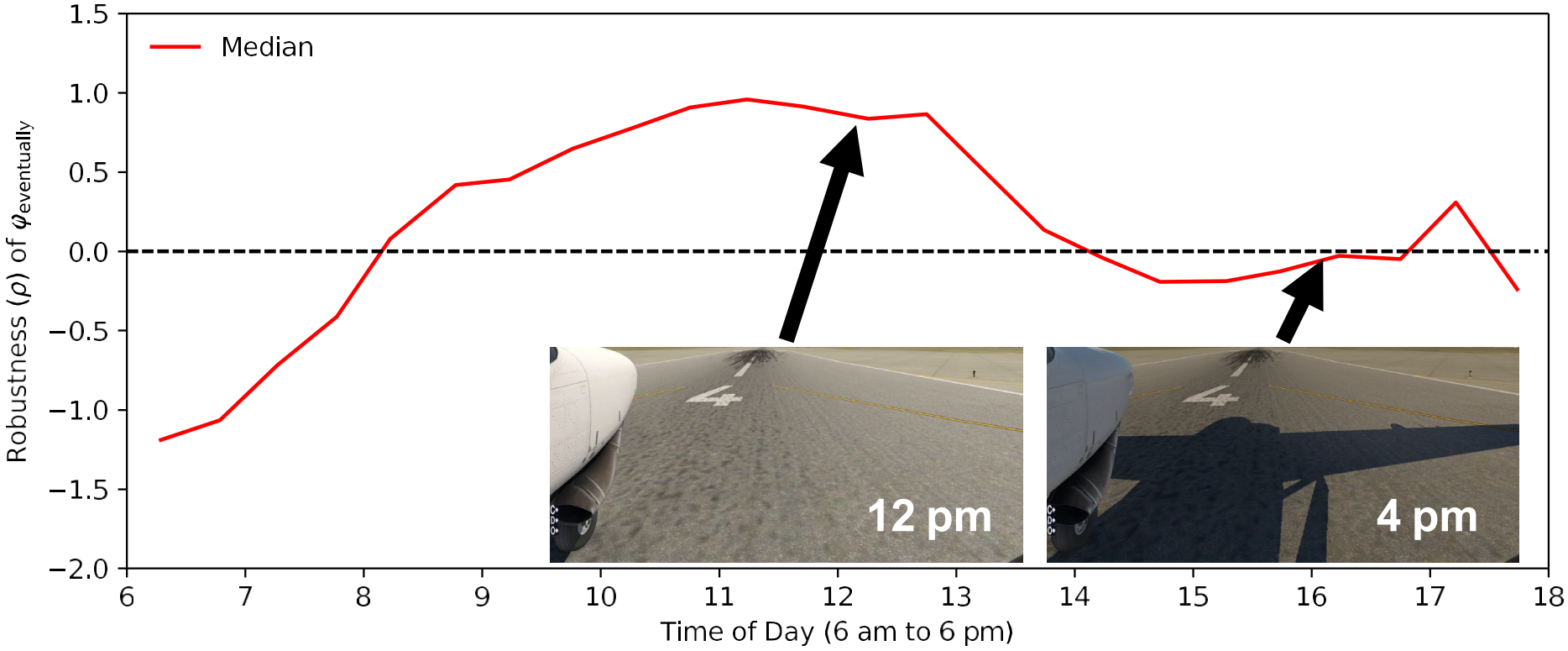
```
# Time of day: from 6 am to 6 pm. (+8 to get GMT, as used by X-Plane)
param zulu_time = ((6, 18) + 8) * 60 * 60

# Rain: 1/3 of the time. Clouds: rain requires types 3-5; otherwise 0-5.
clouds_and_rain = Options({
    tuple([Uniform(0, 1, 2, 3, 4, 5), 0]): 2,  # no rain
    tuple([Uniform(3, 4, 5), (0.25, 1)]): 1    # 25% to 100% rain
})
param cloud_type = clouds_and_rain[0], rain_percent = clouds_and_rain[1]

# Plane: up to 8 m left/right, 2000 m down the runway, 30° left/right.
ego = Plane at (-8, 8) @ (0, 2000),
            facing (-30, 30) deg
```



- Falsification: out of ~4,000 simulations,
  - **45%** violated centerline tracking property
  - **9%** left runway entirely

Counterexample Analysis

- Falsification found several types of failures, e.g. sensitivity to time

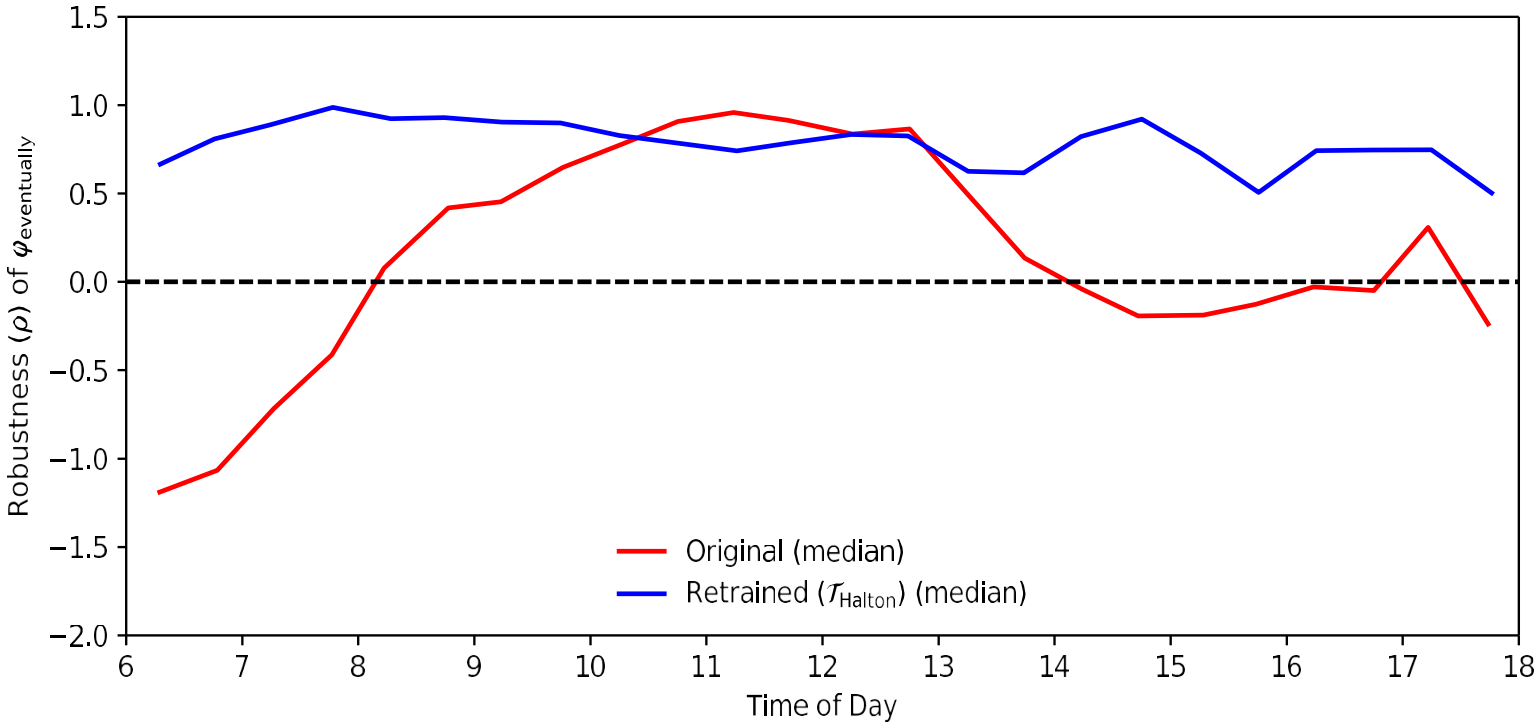


- Follow-up experiments confirmed root cause is the plane's shadow

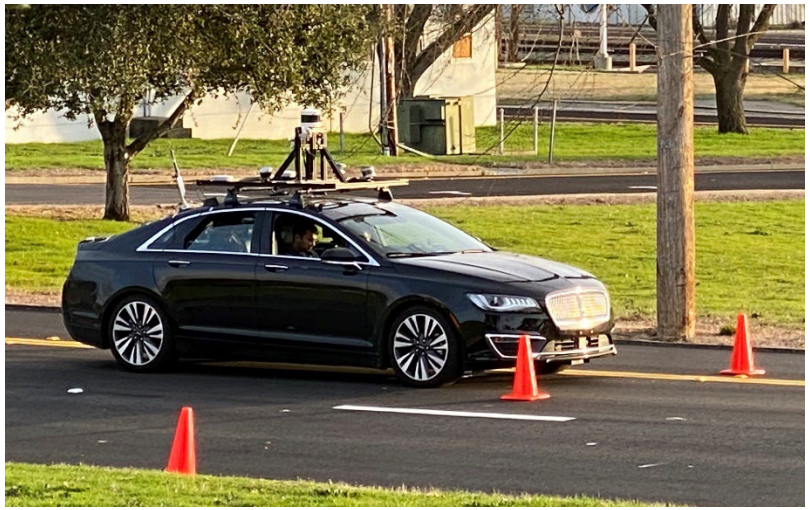
Retraining

- Eliminated dependence on time of day
- Use VERIFAI to generate a new training set (same size as original)
- Used cross-entropy method to *learn* good training distributions

- Obtained much better performance
  - **17%** violated (vs. 45%)
  - **0.6%** left runway entirely (vs. 9%)



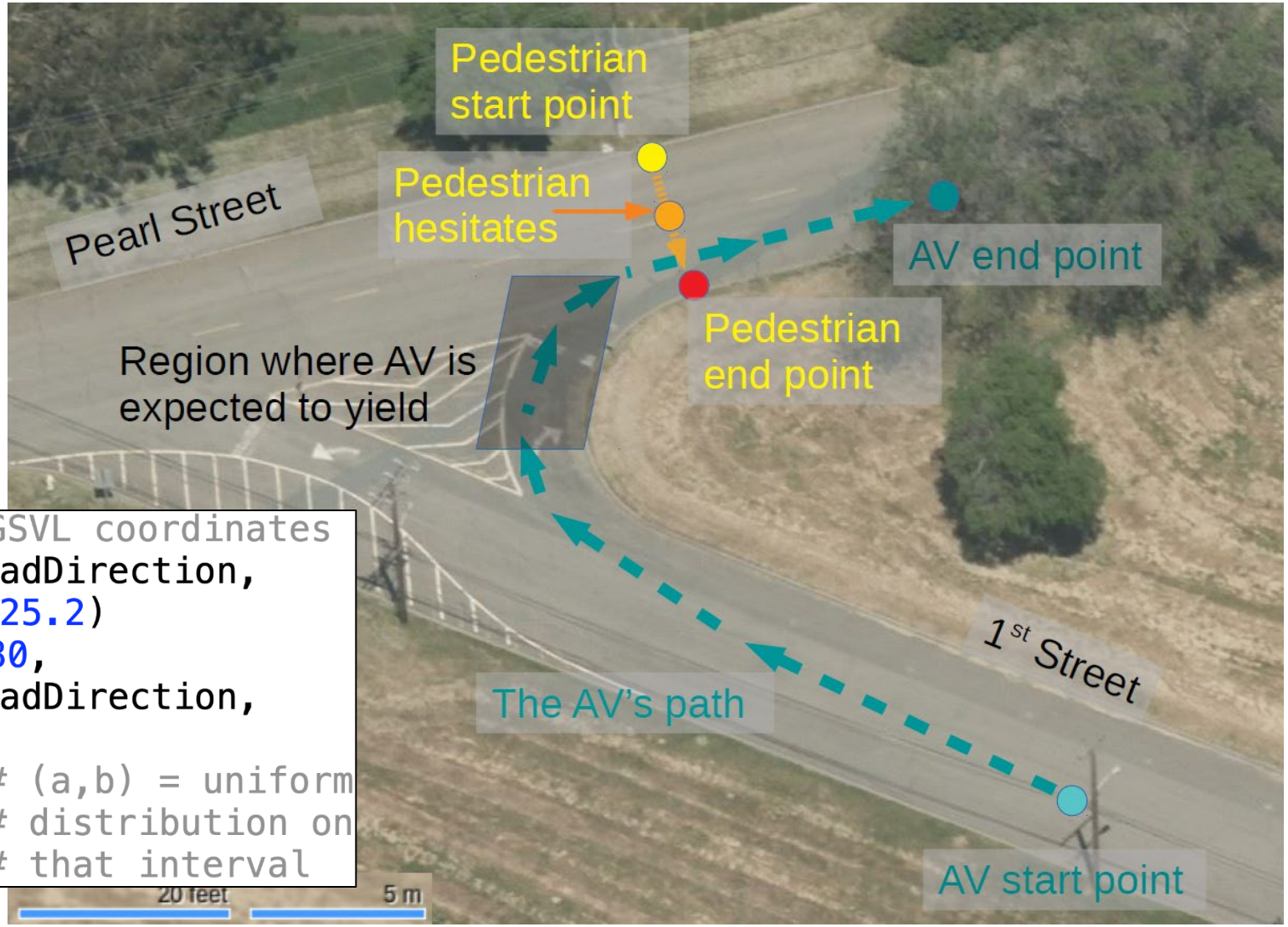
Simulation guided physical testing



Lincoln MKZ running Apollo 3.5

```
ego = EgoCar at 38.6 @ 183.9,  # LGSVL coordinates
        facing 10 deg relative to roadDirection,
        with behavior DriveTo(40 @ 225.2)
ped = Pedestrian at 19.782 @ 225.680,
        facing 90 deg relative to roadDirection,
        with behavior Hesitate,
        with startDelay (7, 15),   # (a,b) = uniform
        with walkDistance (4, 7),  # distribution on
        with hesitateTime (1, 3)   # that interval
```

Snippet of Scenic program



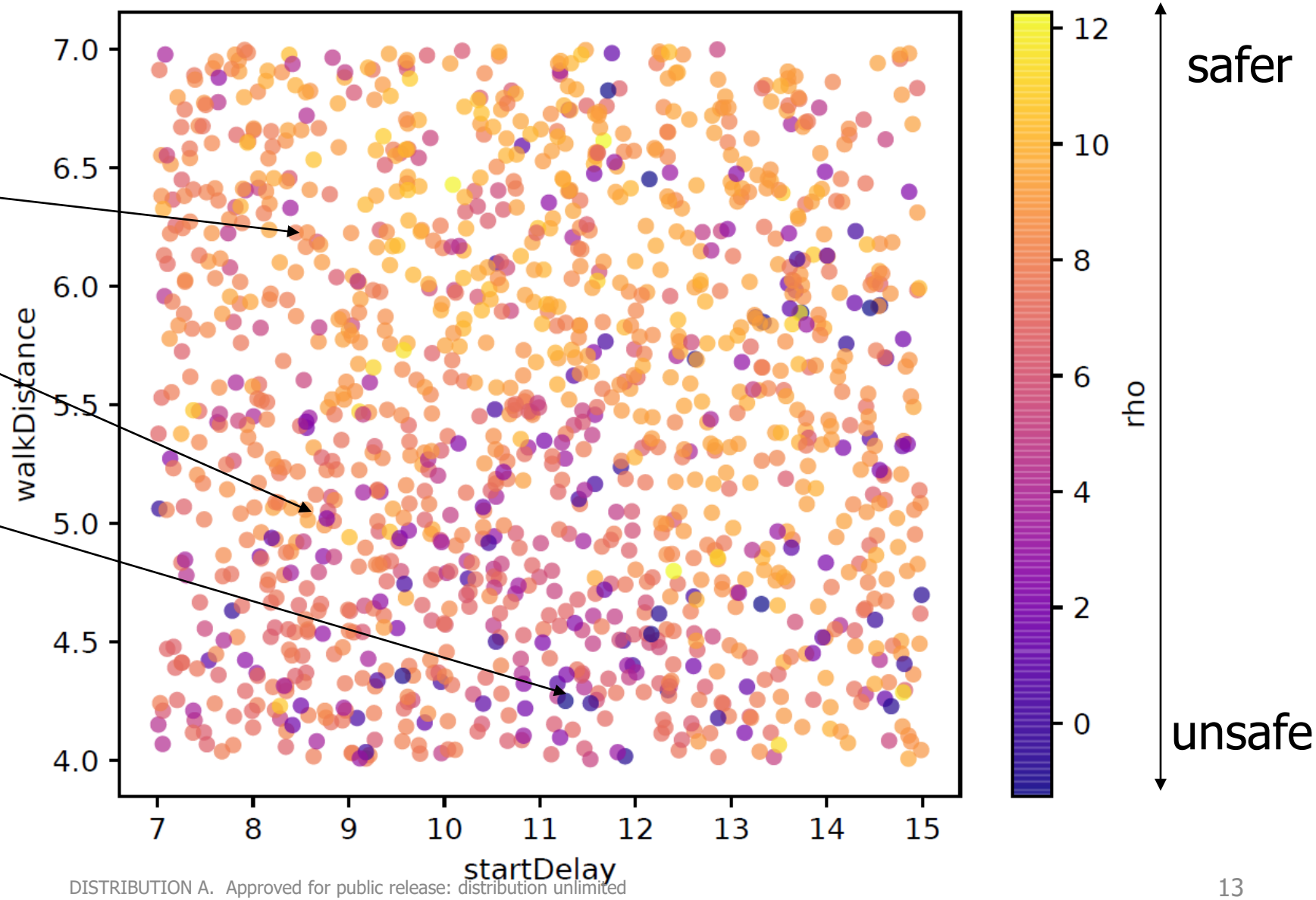
Simulation guided physical testing: analytics and test selection

1294 simulations explored
2% violated safety property

robustly safe

marginally safe

collision

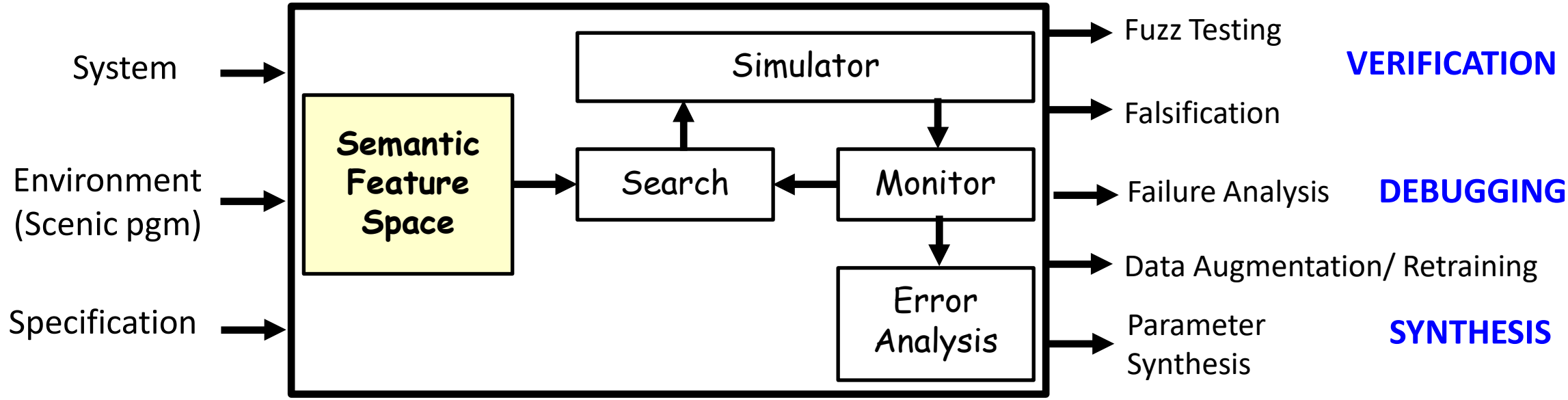


Total 7 test cases selected

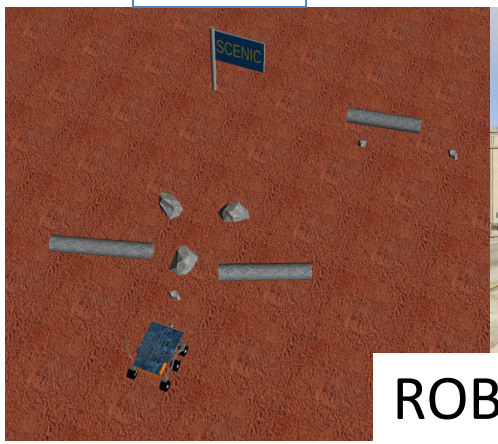
62.5% Unsafe in sim → test
93.5% Safe in sim → test

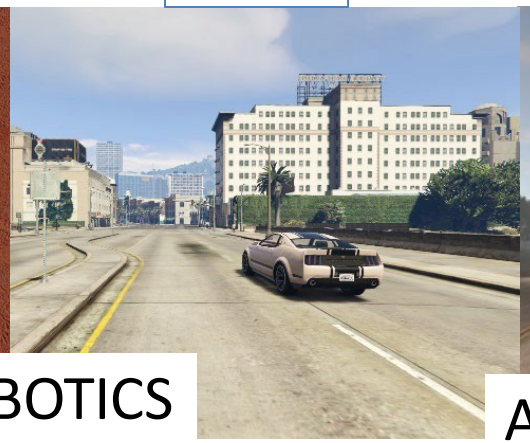
VerifAI: A Toolkit for the Design and Analysis of AI-Based Systems

[Dreossi et al. CAV 2019, https://github.com/BerkeleyLearnVerify/VerifAI]



Webots | GTA-V | LGSVL | CARLA | X-Plane





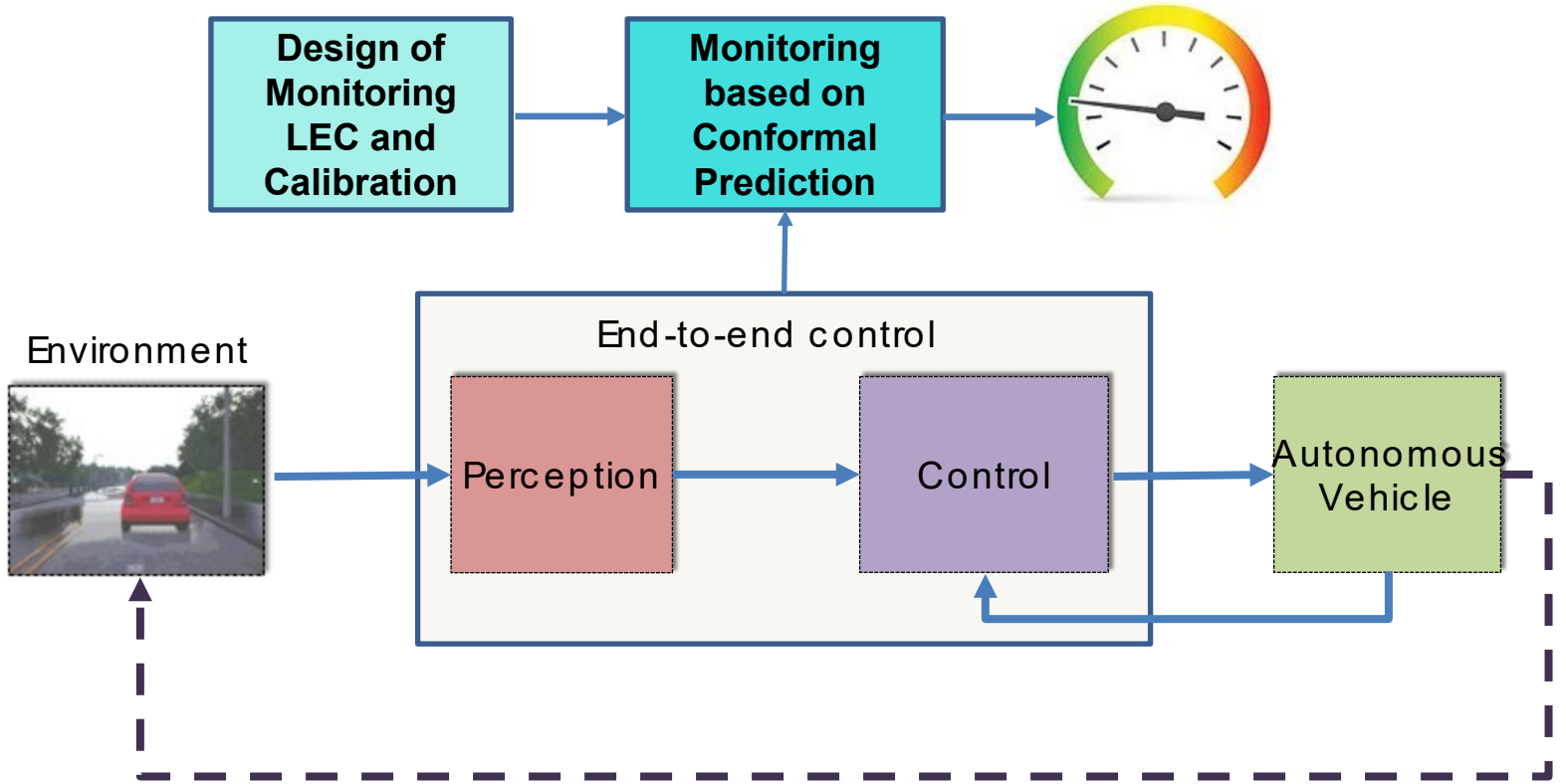
ROBOTICS

AUTONOMOUS DRIVING

AIRCRAFT

Distribution shift detection [ALC toolchain]

Assurance Monitoring of Learning-Enabled Cyber-Physical Systems



Assurance monitoring based on inductive conformal anomaly detection

- Variational autoencoder (VAE)
- VAE for regression
- Adversarial Autoencoder (AAE)
- Deep support vector description (SVDD)

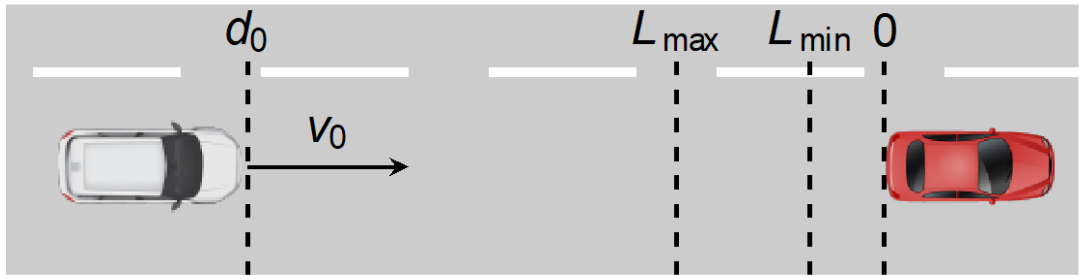
Evaluation using self-driving simulator and open datasets

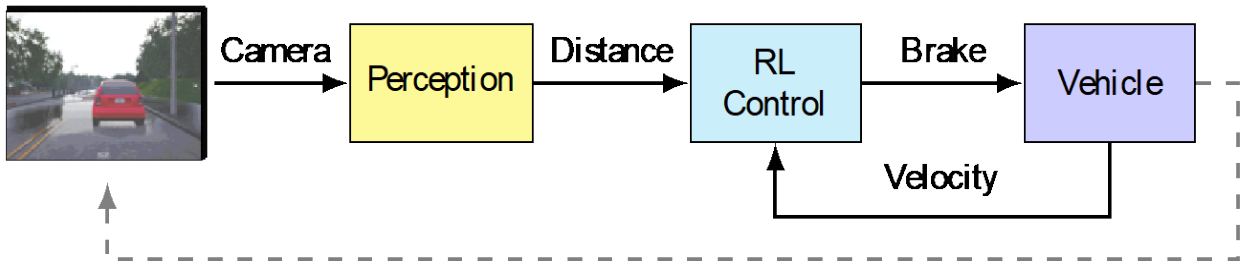
- Advanced emergency braking system
- End-to-end self-driving controller
- Ford's autonomous vehicle seasonal dataset

Evaluation using autonomous underwater vehicle

- Underwater pipeline inspection
- Obstacle avoidance

Advanced Emergency Braking System (AEBS)





Data Generation using CARLA

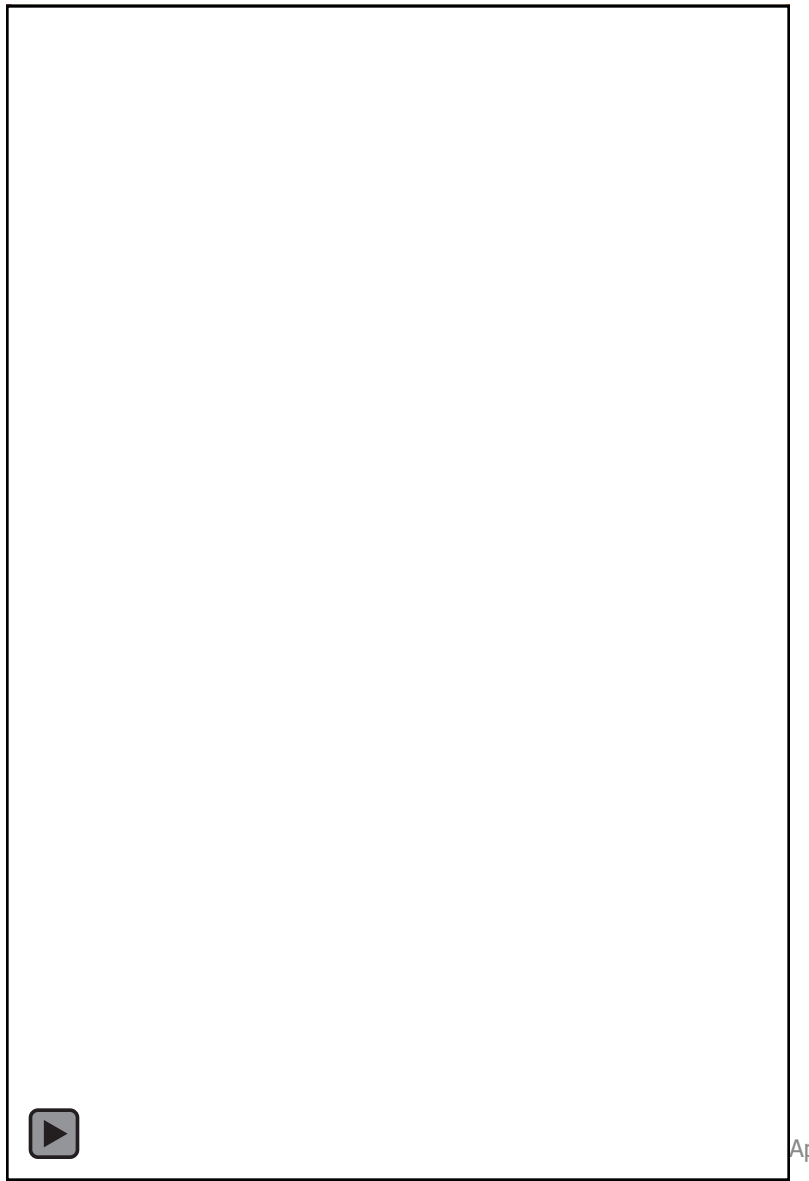
d_0	100 m approximately
v_0	Randomly sampled between 90 and 100 km/h
L_{min}	1 m
L_{max}	3 m
CARLA precipitation parameter r	Randomly sampled between 0 and 20
Sampling period	1/20 sec = 50 ms

Learning-Enabled Components

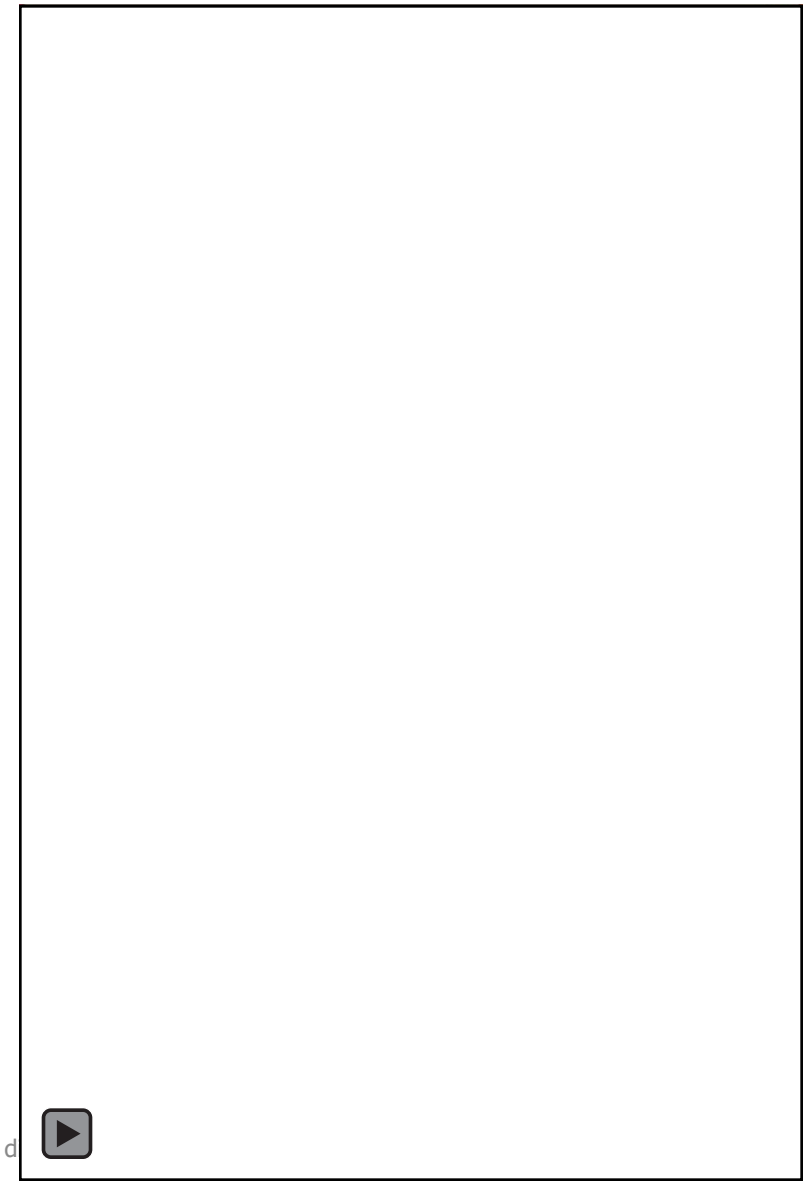
- Perception: CNN with 11 layers
- Control: Reinforcement learning controller trained using DDPG
- VAE: CNN encoder with 4 layers, 1024 FC layer, and symmetric decoder
- SVDD: 4 convolution layers and 1568 FC layer

Simulation results

In-distribution



Out-of-distribution



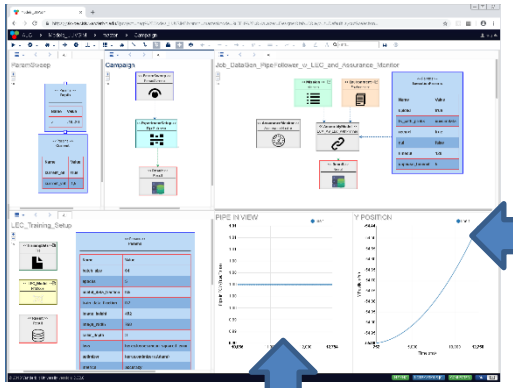
ALC Toolchain: A model-based engineering toolchain for assurance of learning-enabled systems

- The model driven toolchain supports training, verification and design-time assurance of learning enabled components.
- Toolchain helps with developing safety assurance cases for the system using collected evidence.
- Complete provenance tracking of Experimental runs and data collection is supported.

Collaborative Modeling

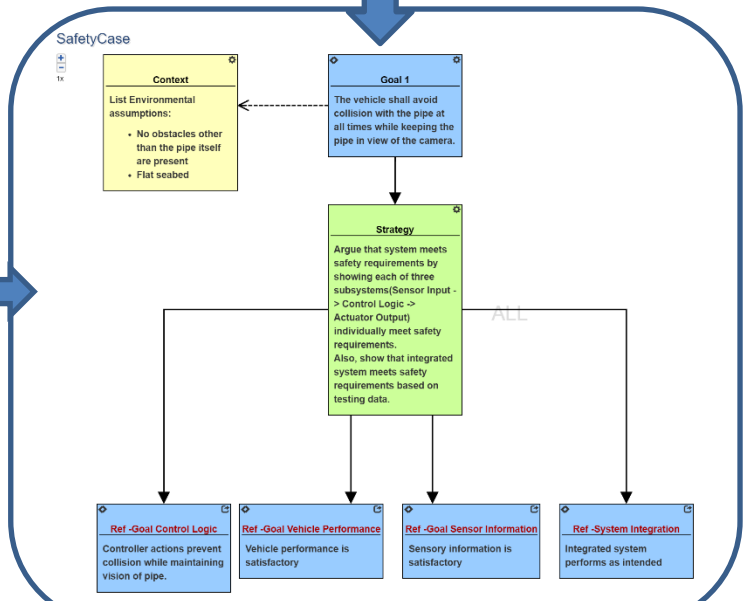
System Integrator

LEC Developer



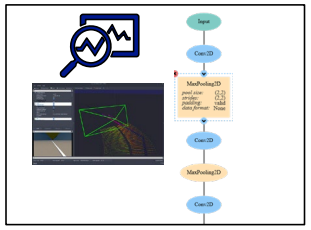
Design Time Assurance

Assurance Engineer

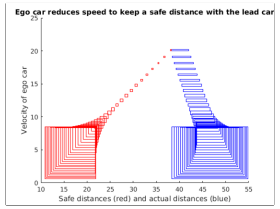


ALC Workflows

Execution, Training, Data Collection, Verification

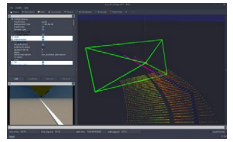


(d) Run System with LEC



(c) Verify Closed-Loop System With LEC

(b) Train LECs



(a) Run Scenarios To Collect Data

Typical Workflow Sequence

19

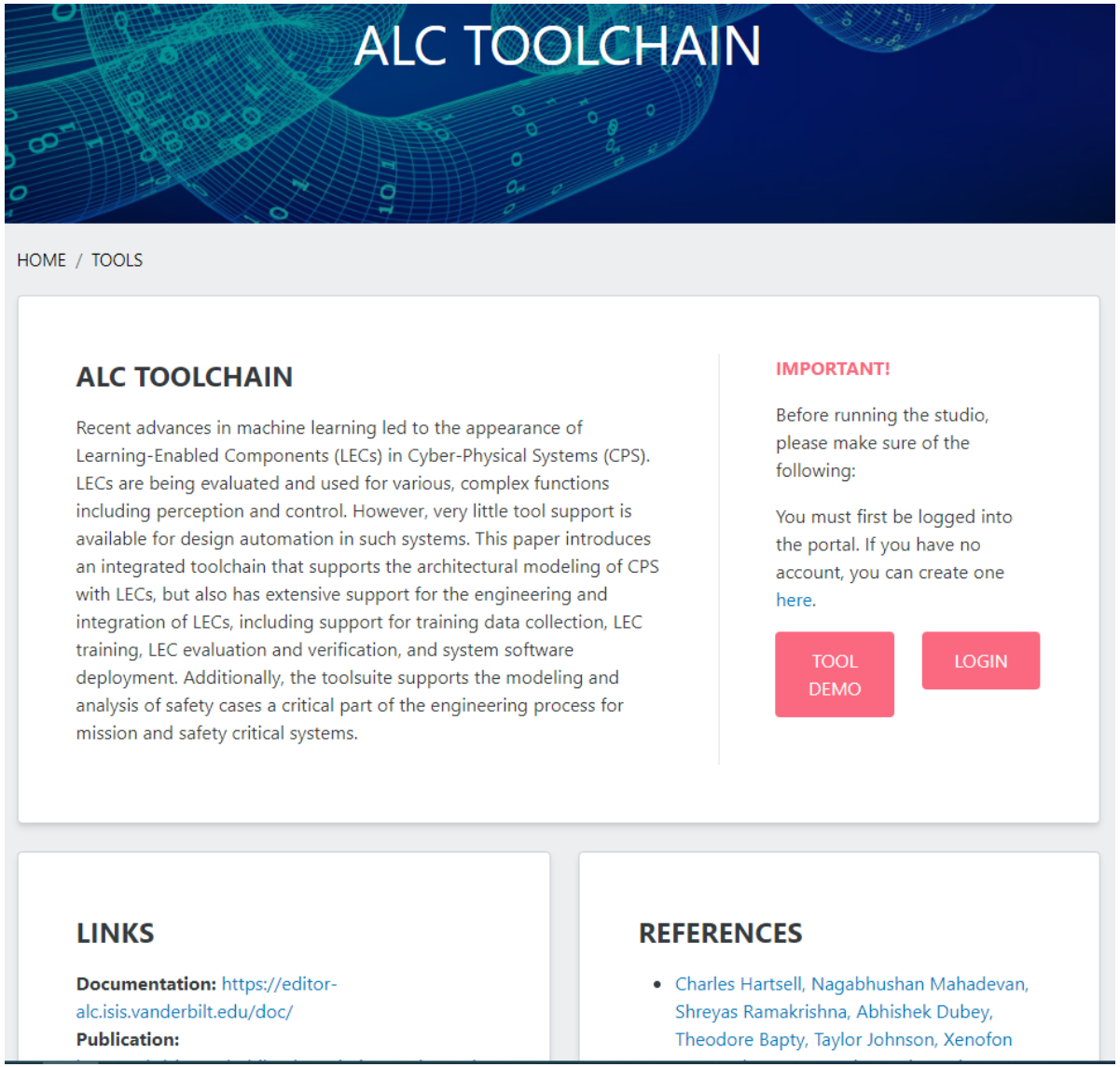
Formal Verification

Formal Verification Tools

Toolkit	NN Architecture	Systems	Verification Method	Specifications	Benchmarks	Scalability
Marabou	Feed Forward Fully-connected, Convolutions, Relu, max-pool, Abs, Sign	Open-loop	Reluplex (Simplex + Relu-splitting); Symbolic Bound Tightening; parallelization via Split-and-Conquer	**Reachability, local robustness**	ACASXU, CIFAR10 (resnet, Marabou), ERAN, mnistfc, oval21, verivital, SmallTaxiNet, MNIST	14k ReLU nodes (CIFAR 10 RESNET-4b)
NNV	Feed Forward Fully-connected, convolutions, ReLU, max-pool, nonlinear, SegNets	Open-loop, Closed-loop	Reachability (exact/approximate star set, zonotope, interval, etc.)	**Reachability, local robustness (any represented as union of convex polyhedra)**	ACASXU, MNIST, CIFAR10, GTSRB, VGG16/19, NGC SegNet	VGG16/19 largest (ImageNet)
OVERT	Feed Forward Fully-connected ReLU	Closed-loop, nonlinear, discrete time	Reachability via MILP	**Reachability (bounded time safety and bounded time goal reaching properties)**	Cruise Control, Pendulum, Tora, Car	500 ReLU network 100 timesteps; 64 ReLU network x55 timesteps
VenMAS	LE-CPS (NN perception paired with symbolic decision making for multiple agents + linear environment)	Closed-loop multi-agent systems	MILP (Complete)	**Bounded temporal statements, bounded alternating temporal logic**	ACASXU, VCAS2 (2-Agent ACAS)	~20 time steps on VCAS2 against bounded ATL specs.
Venus2	Feed Forward Fully-connected, convolutions, ReLU, max-pool	Open-loop	Symbolic Interval Propagation; MILP (Complete)	**Reachability, local robustness wrt white noise, luminosity, contrast.**	ACASXU, MNIST, CIFAR10, VGG16, Boeing Taxinet,	~100k ReLU nodes against 10^-4 noise perturbations
Verisig	Feed Forward Fully-connected, sigmoid, tanh	Closed-loop	Reachability (Taylor Model preconditioning and shrink wrapping)	**Reachability**	F1/10, Mountain Car, Quadrotor, Tora, others	~1000 sigmoid nodes, ~50 inputs to NN, ~100 time steps on F1/10

Takeaways

Assured Autonomy Tools Portal



https://assured-autonomy.org

- Design studios
- Tools
  - Integrated Tools
  - Formal Verification
  - Simulation & Testing
  - Confidence Estimation & Monitoring
  - Recovery & Resilience
  - Assurance Arguments & Cases
  - Other Technologies
- Publications

Summary

- Impressive progress in machine learning in last decade, but challenges remain for use in safety critical systems
- Assurance and trustworthiness critical for use in safety critical systems
- Program developing an assurance architecture and toolbox for assurance of learning-enabled systems, released through research community platform (CPS-VO) and (planned) through DoD platform (JAIC)
  - www.assured-autonomy.org
- Established community of researchers and practitioners in AI safety
  - AI safety workshops, NN verification competition, CPS-VO community and design studios
  - Publications in top AI, CPS and Control conferences
  - International collaborations

www.darpa.mil