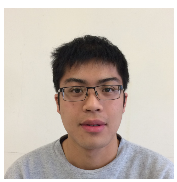
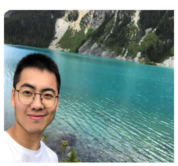
Building Error-Resilient and Attack-Resilient ML-Enabled CPS

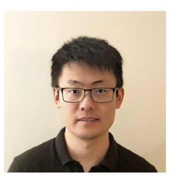
Abraham Chan, Zitao Chen, Pritam Dash, Niranjana Narayanan,

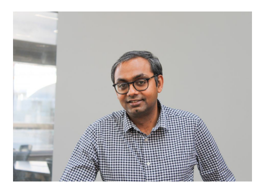
Guanpeng Li, Arpan Gujarati, Sathish Gopalakrishnan,

Karthik Pattabiraman

University of British Columbia

Machine Learning

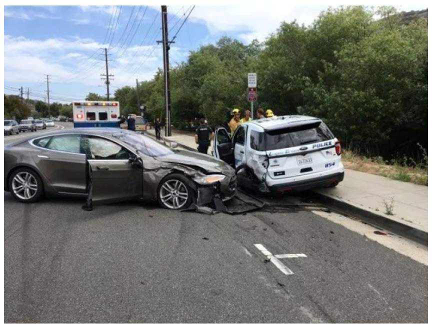
Home Care

Law Enforcement

Self-Driving
Cars

Machine-learning is increasingly used in safety-critical CPS

Reliability and Security?



Our Goal

Provide Resilience without any human intervention for both faults and attacks

ML-enabled CPS: Why Resilience?

CPS deployed in unpredictable scenarios

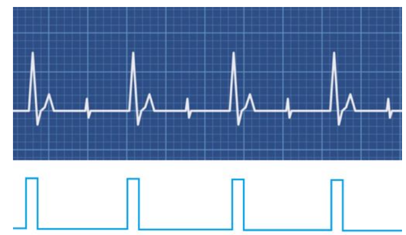
ML cannot deal with unseen situations

Cannot simply stop under errors & attacks

ML-enabled CPS Resilience: Challenges

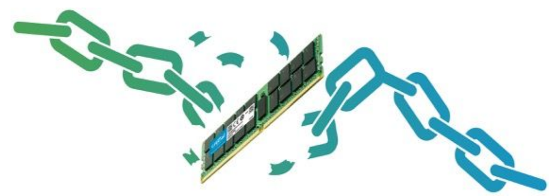
Real-time Operation

- In place correction/recovery



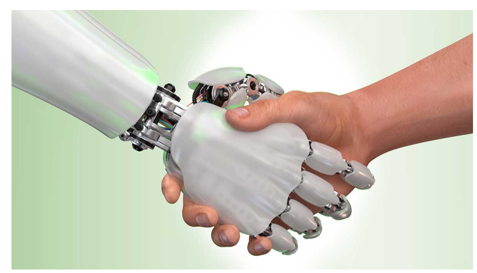
Resource Constraints

- Low performance overheads

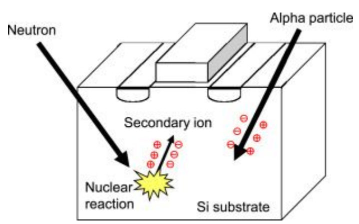


No human in the loop

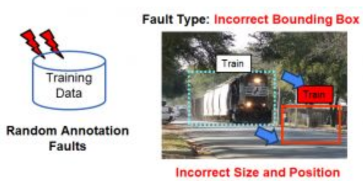
- Completely automated



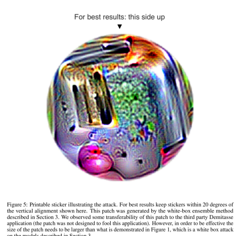
Fault and Threat Model



Soft Errors



Training Data Faults



Adversarial Patch Attacks

ML System
+
CPS Platform

Correct Outputs

Outline

Motivation

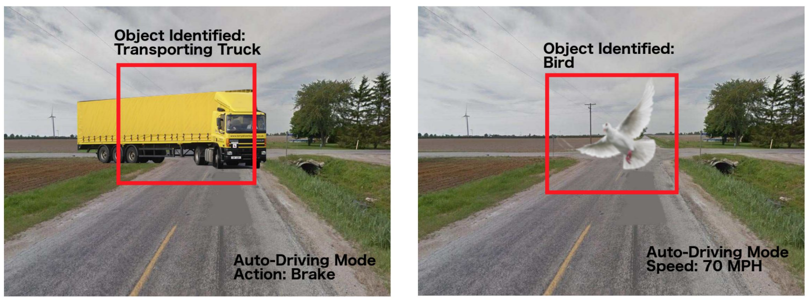
Soft Errors [DSN'21 - Best Paper nominee, AISafety'21 - b.p. nominee]

Training Data Faults [QRS'21 - Best Paper award, DSN'22, ISSRE'23]

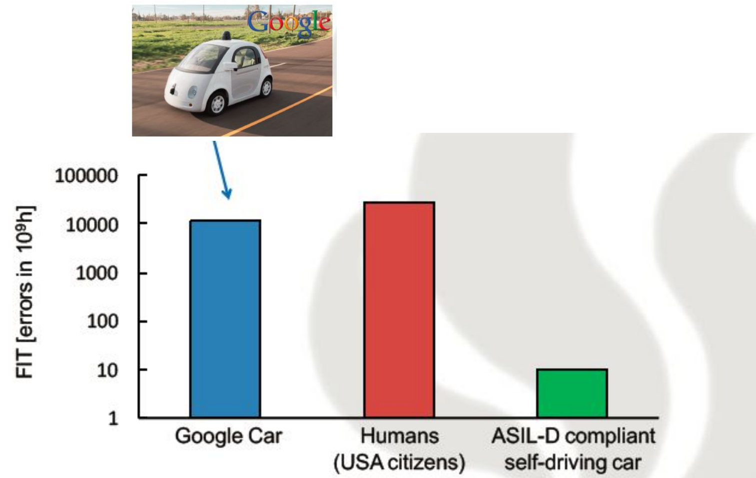
Adversarial Patch Attacks [AsiaCCS'23]

Ongoing work and Conclusions

Soft Errors and AVs



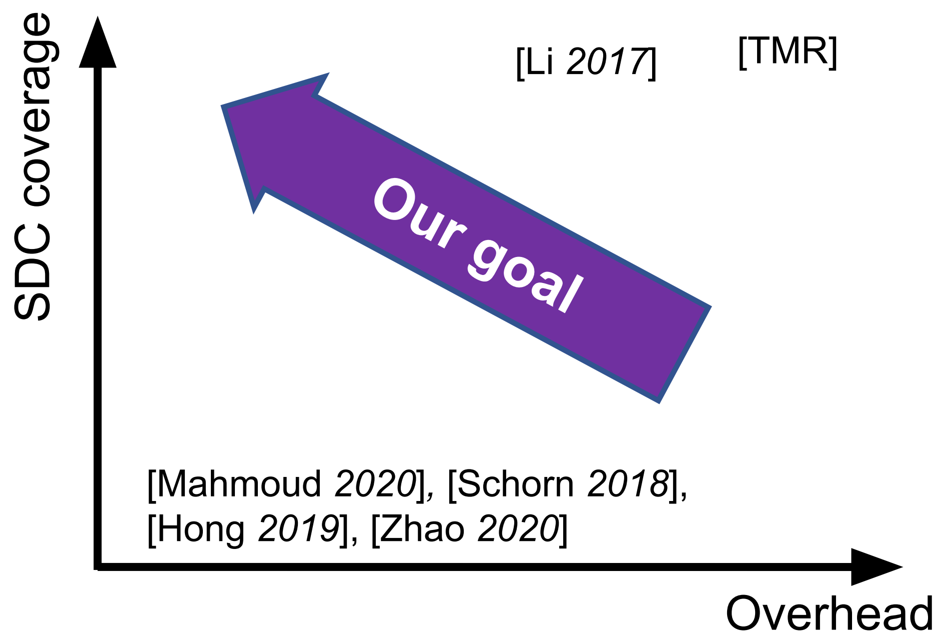
[Our work - SC 2017]



[Saxena'16]

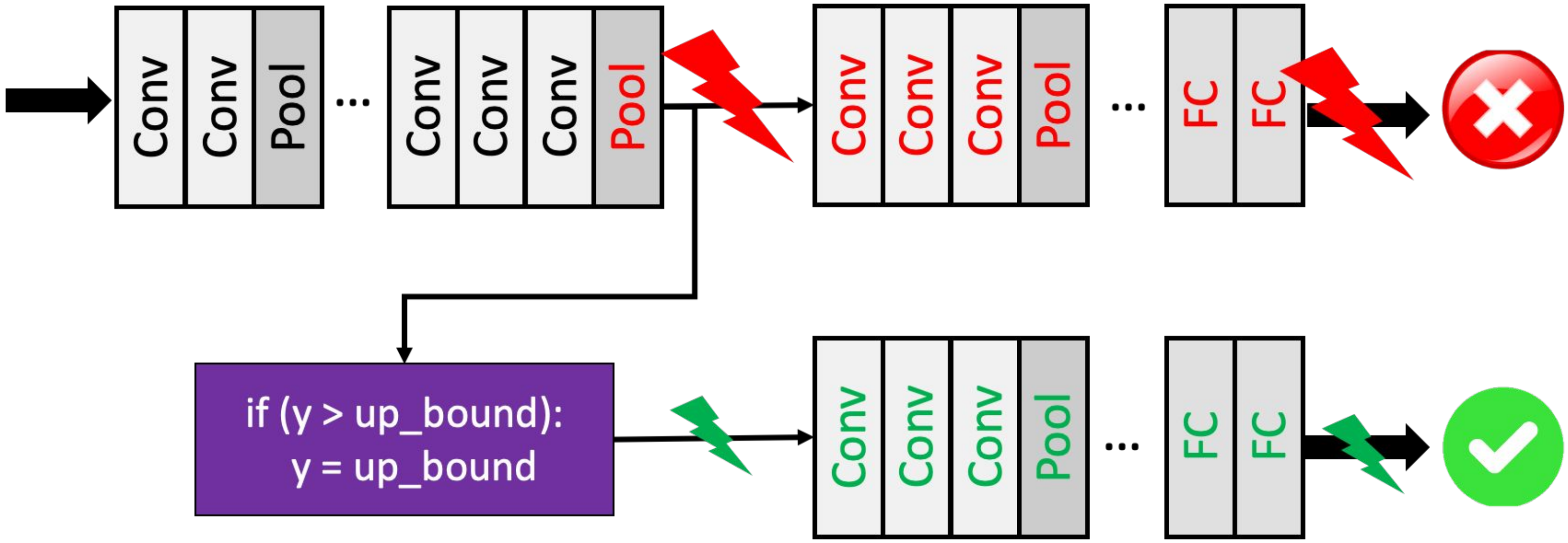
- **Safety standard – Automotive Safety Integrity Level (ASIL-D)**
  - Error rate <10 FIT (per 1 billion hours) – ISO 26262
  - DNN systems do not satisfy it without protection

Towards Reliable DNNs

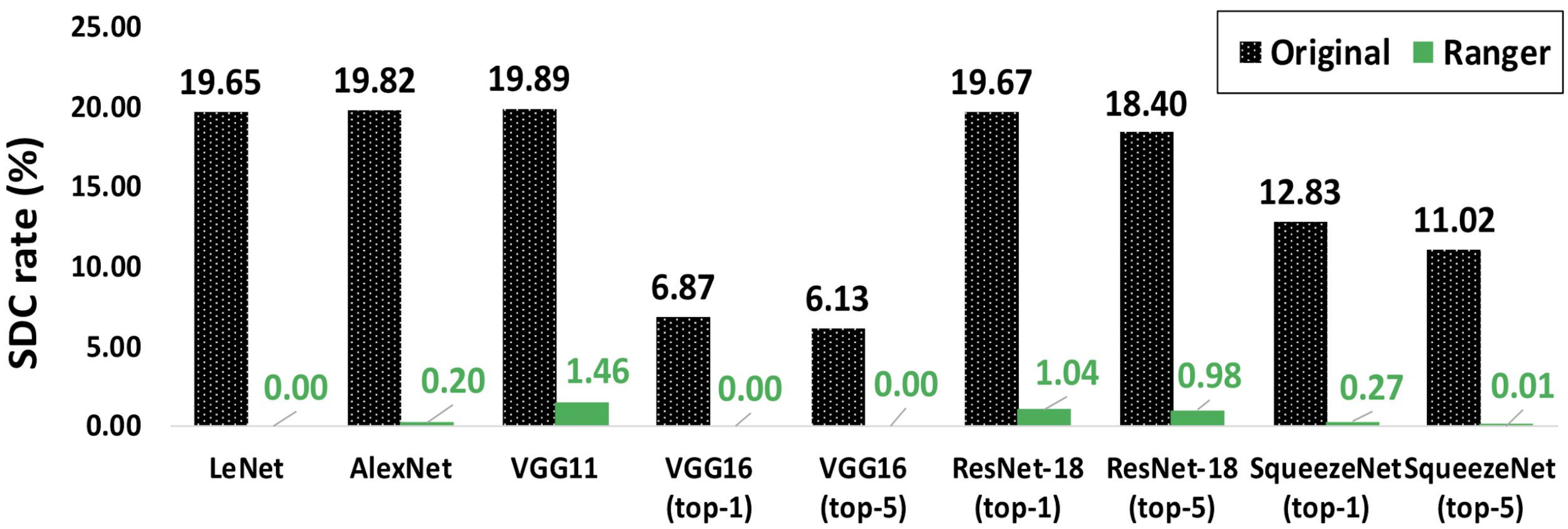


Key idea

Transform Critical Faults into Benign Faults, via **Selective Range Restriction in Hidden Layers**



Effectiveness of Ranger



SDC rate reduced from 14.92% to 0.44% (**34X reduction**)

Accuracy of DNNs

No accuracy degradation for the DNNs
(without fault)

Overhead

0.53%
Floating-point Operations (FLOPs)

Ranger in Action

Prediction (without fault):
156.58

Prediction (with fault):
-78.09

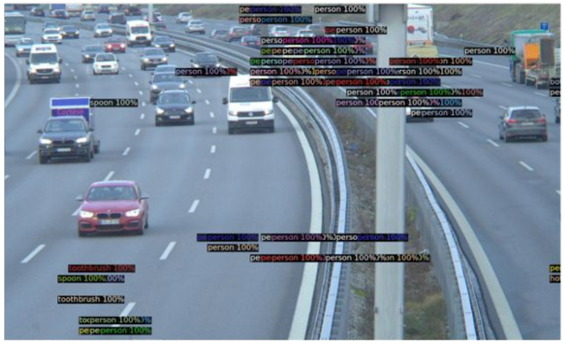
Prediction (with fault):
(with Ranger)
155.97

Code: https://github.com/DependableSystemsLab/Ranger

Real world adoption

Post-Optimization Training

(a) Yolov3 prediction without fault



(b) Yolov3 prediction corrupted with single weight fault

(c) Yolov3 prediction corrupted with single weight fault - Ranger applied

Source: https://docs.openvino.ai/nightly/pot_ranger_README.html

Outline

Motivation

Soft Errors [DSN’21 - Best Paper nominee, AISafety’21 - b.p. nominee]

Training Data Faults [QRS’21 - Best Paper award, DSN’22, ISSRE’23]

Adversarial Patch Attacks [AsiaCCS’23]

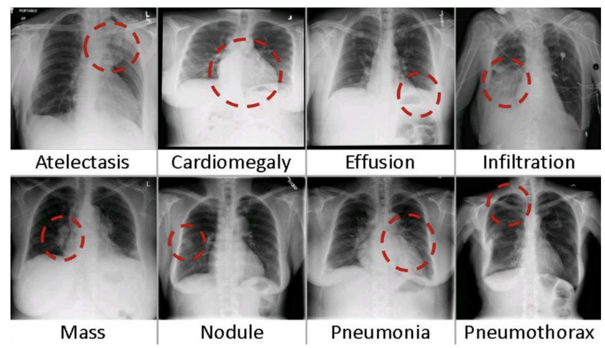
Ongoing work and Conclusions

Training Data Faults

70% of Lyft dataset missing, mislabelled [Kang et al, 2022]

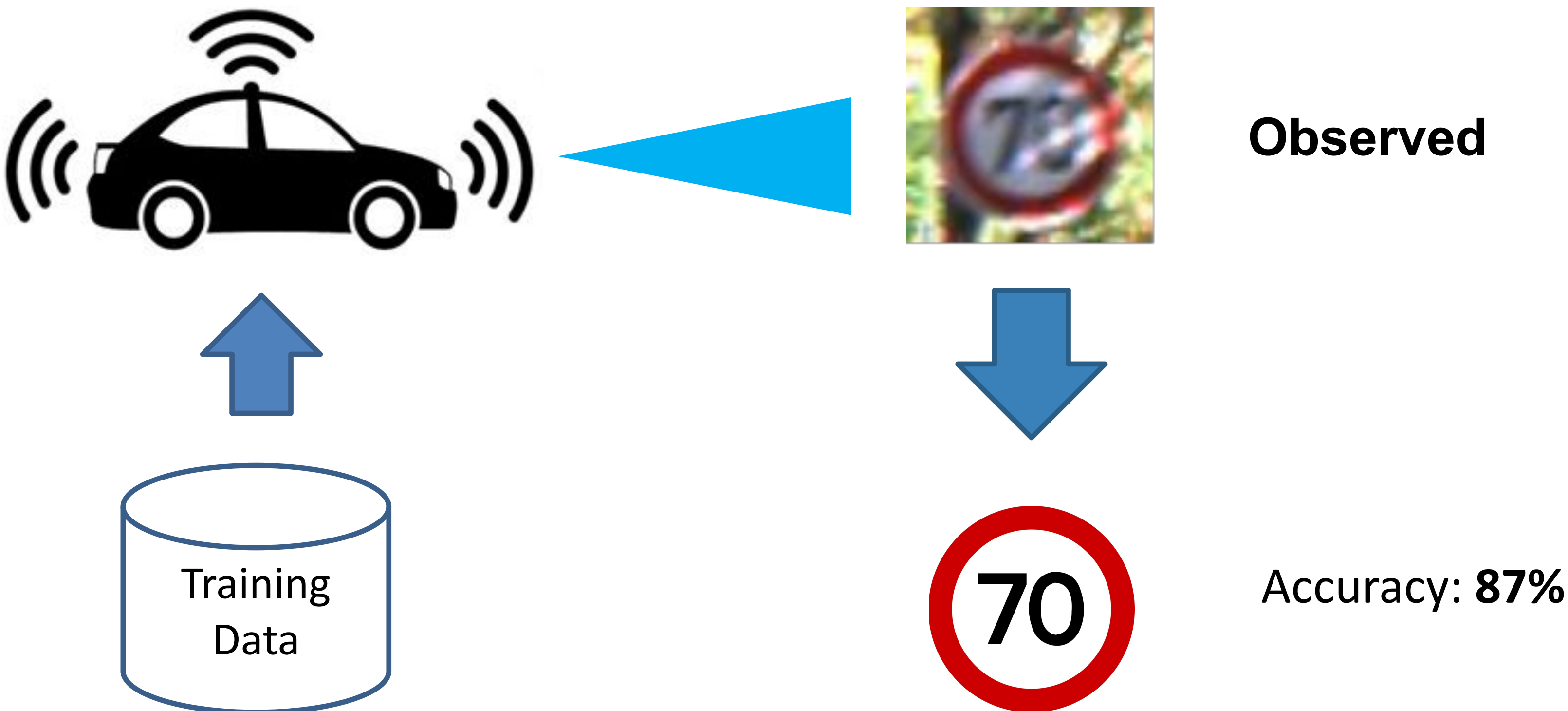
Autonomous Vehicles

20% of ChestX-ray mislabelled [Tang et al, 2021]

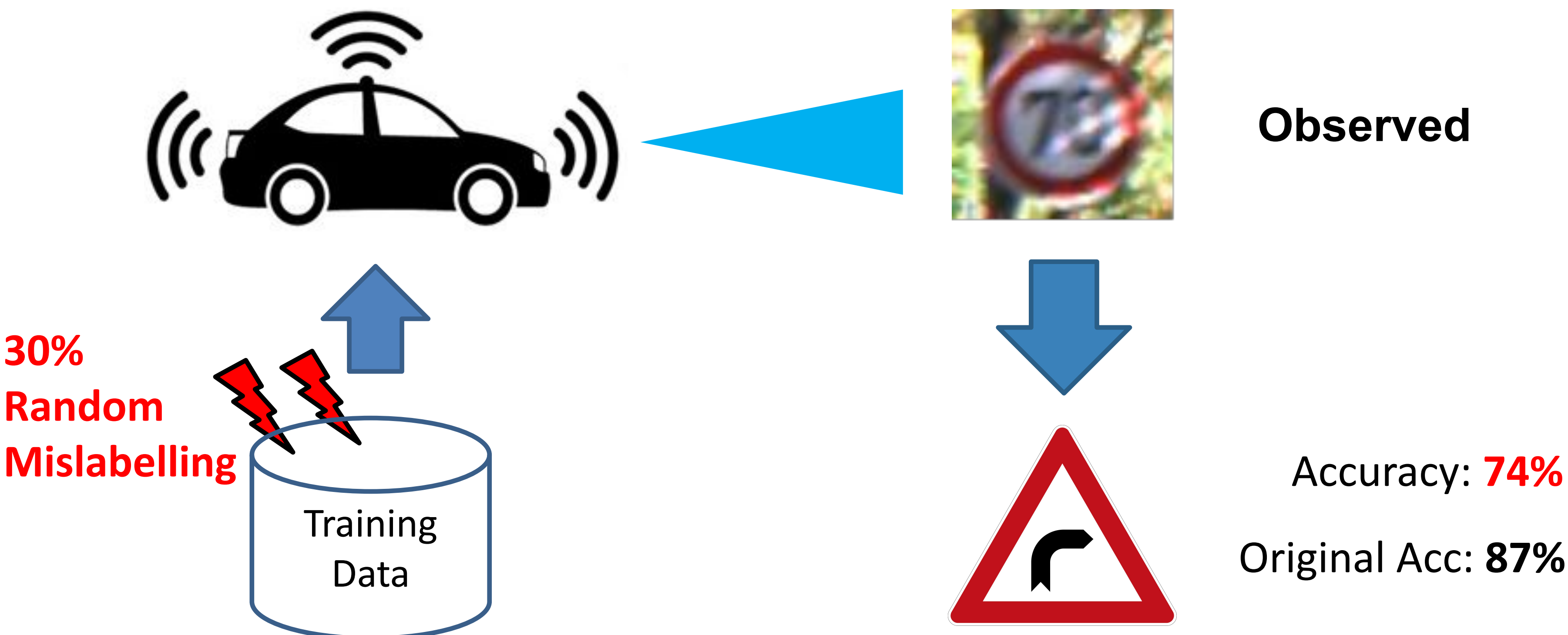


Healthcare

Autonomous Vehicle Example



Autonomous Vehicle Example



How to mitigate training data faults?

1. Label Correction
2. Knowledge Distillation
3. Robust Loss
4. Label Smoothing
5. Ensembles

More Effort

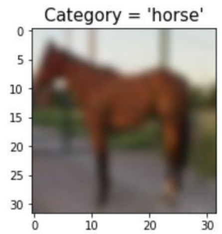
Less Effort

Our Work: The Fault in Our Data Stars: Studying Mitigation Techniques against Faulty Training Data in ML Applications **[DSN'22]**

Neural Networks

ML Model Name	Depth (# of Layers)
ConvNet	Shallow
DeconvNet	Shallow
MobileNet	Deep
ResNet18	Deep
ResNet50	Deep
VGG11	Deep
VGG16	Deep

Evaluation Datasets



CIFAR-10
Object Detection

GTSRB
Self-Driving Cars

Pneumonia
Medical Diagnosis

Safety-Critical Applications

Reliability Metric: Accuracy Drop (AD)

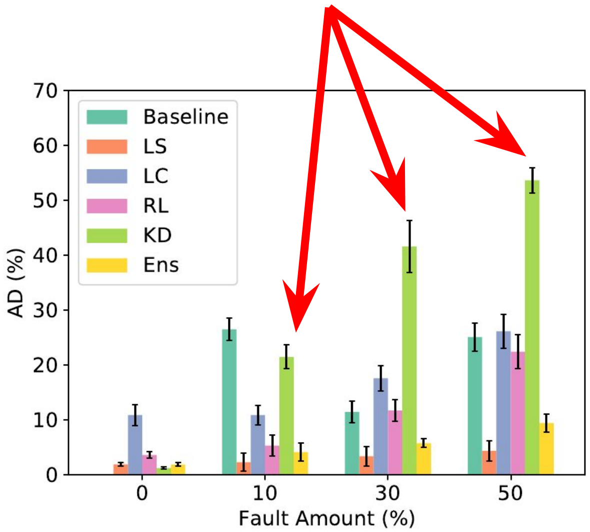
Model trained with golden data		Model trained with faulty data	
Test Image 1	✓	Test Image 1	✓
Test Image 2	✓	Test Image 2	✗
Test Image 3	✓	Test Image 3	✗
Test Image 4	✗		

Accuracy Drop (AD) = 2 / 3 = 67%

AD Across Models

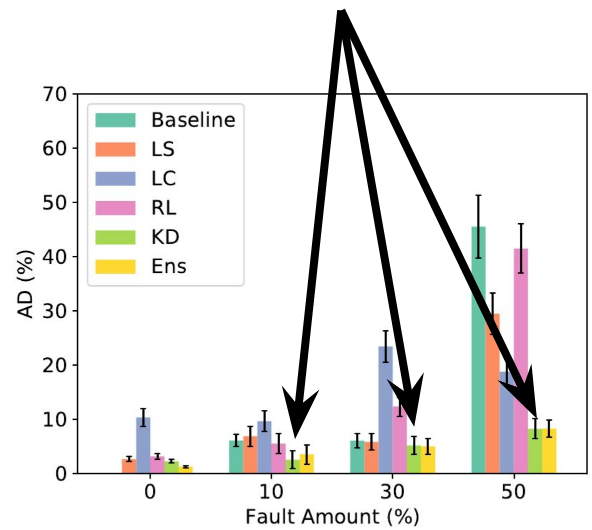
Higher AD is worse

KD not effective here



GTSRB, ResNet50, Mislabelling

KD effective here

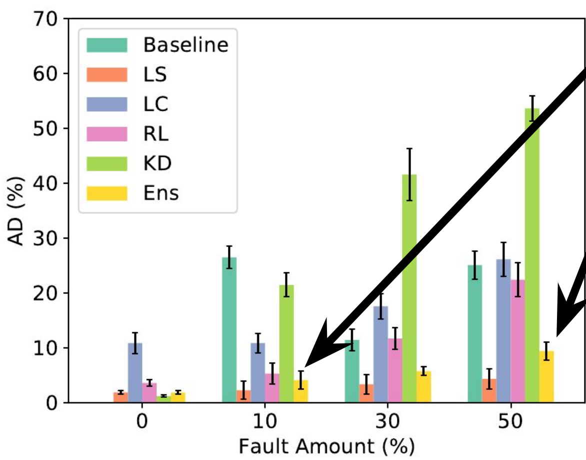


GTSRB, VGG16, Mislabelling

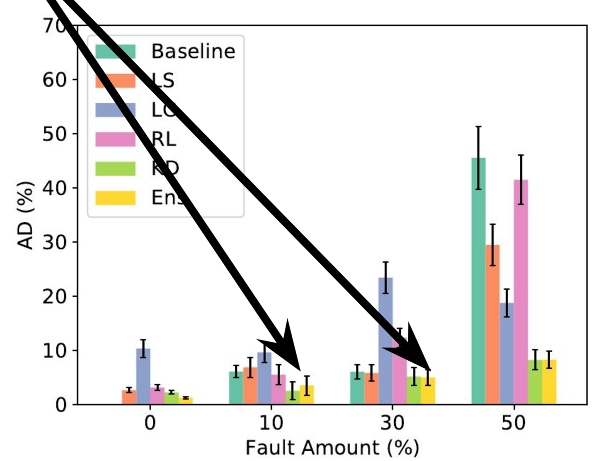
AD Across Models

Ensembles effective across models

Higher AD is worse

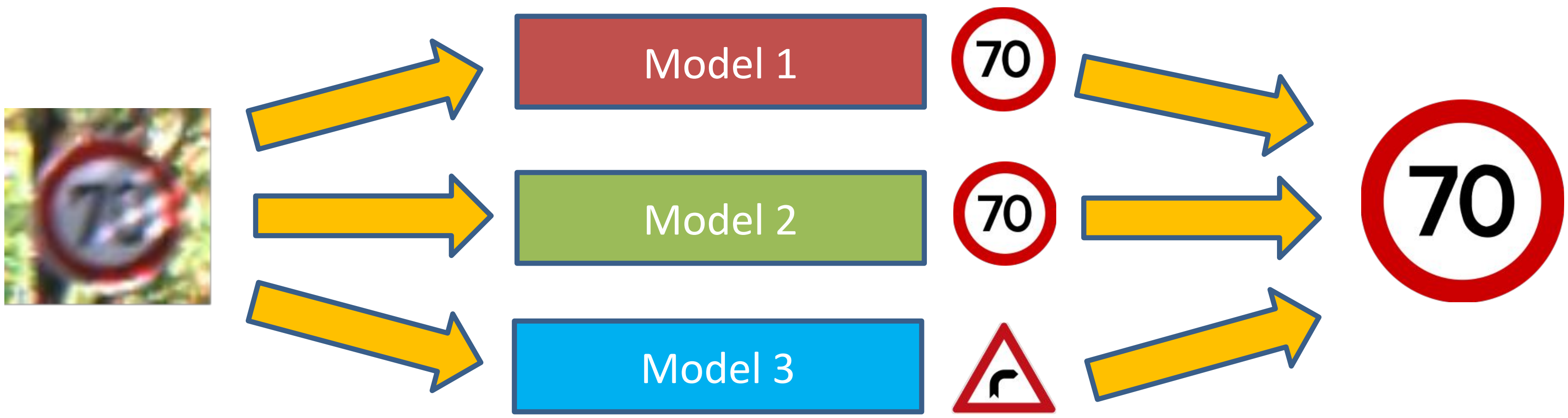


GTSRB, ResNet50, Mislabelling



GTSRB, VGG16, Mislabelling

What makes Ensembles Resilient?

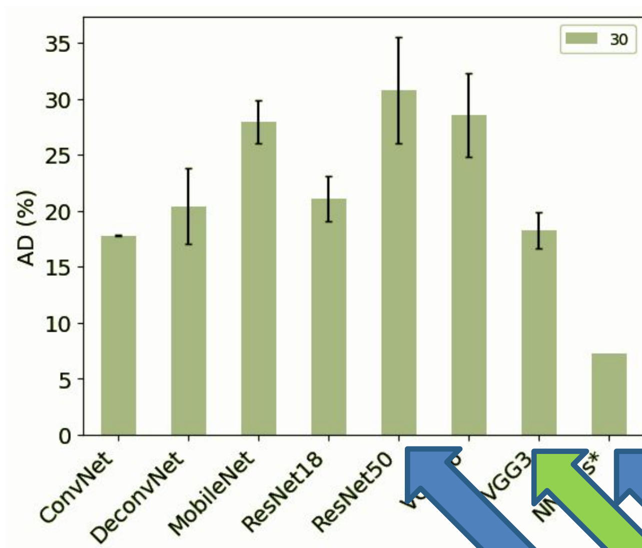


Our Work: Understanding the Resilience of Neural Network Ensembles against Faulty Training Data **[QRS'21]**

Individual Models vs Ensemble

CIFAR-10, 30% Mislabelling

Lower AD is Better



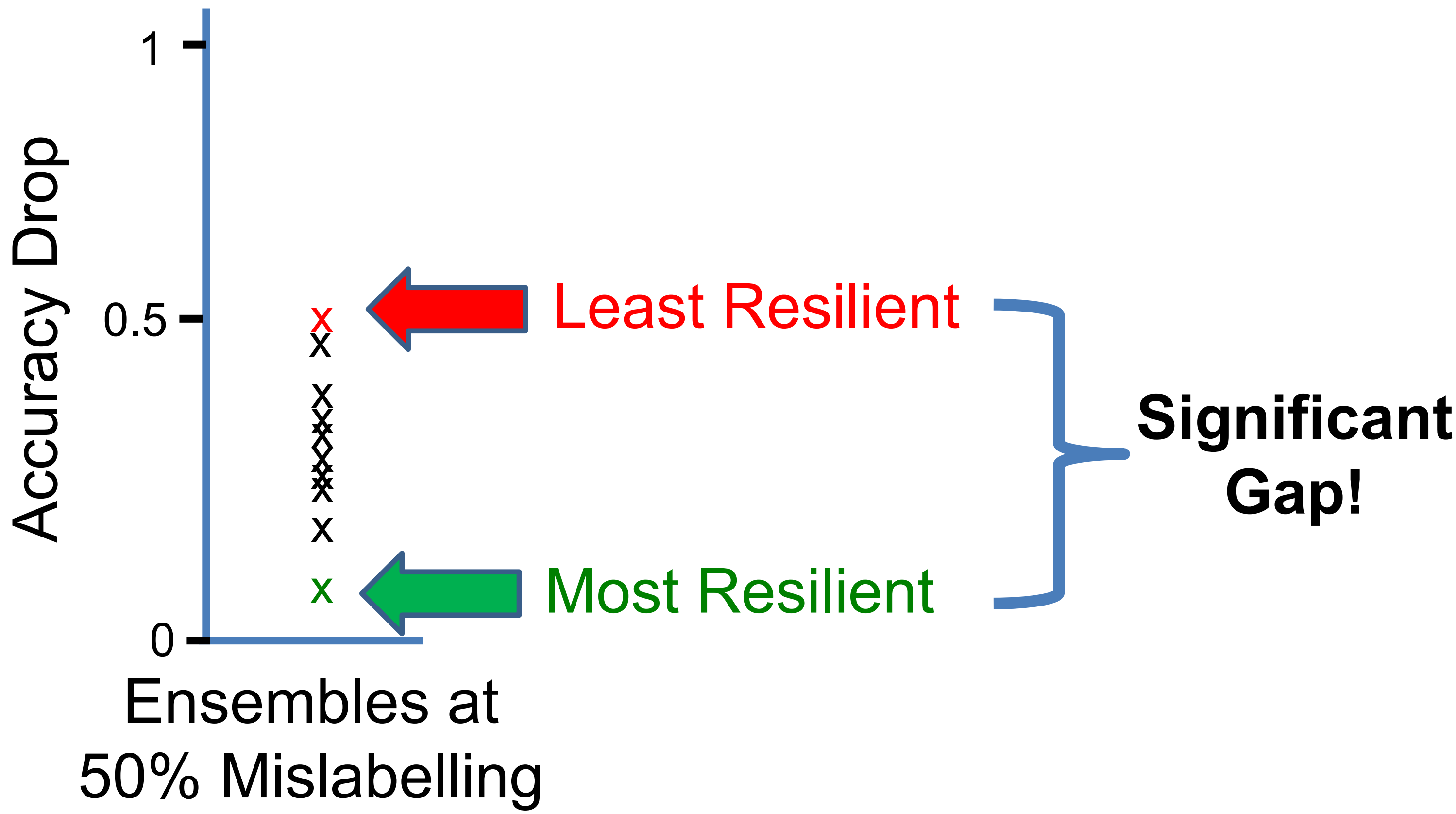
Key Observation:
Accuracy ⇏ Resilience

NN-Ensemble

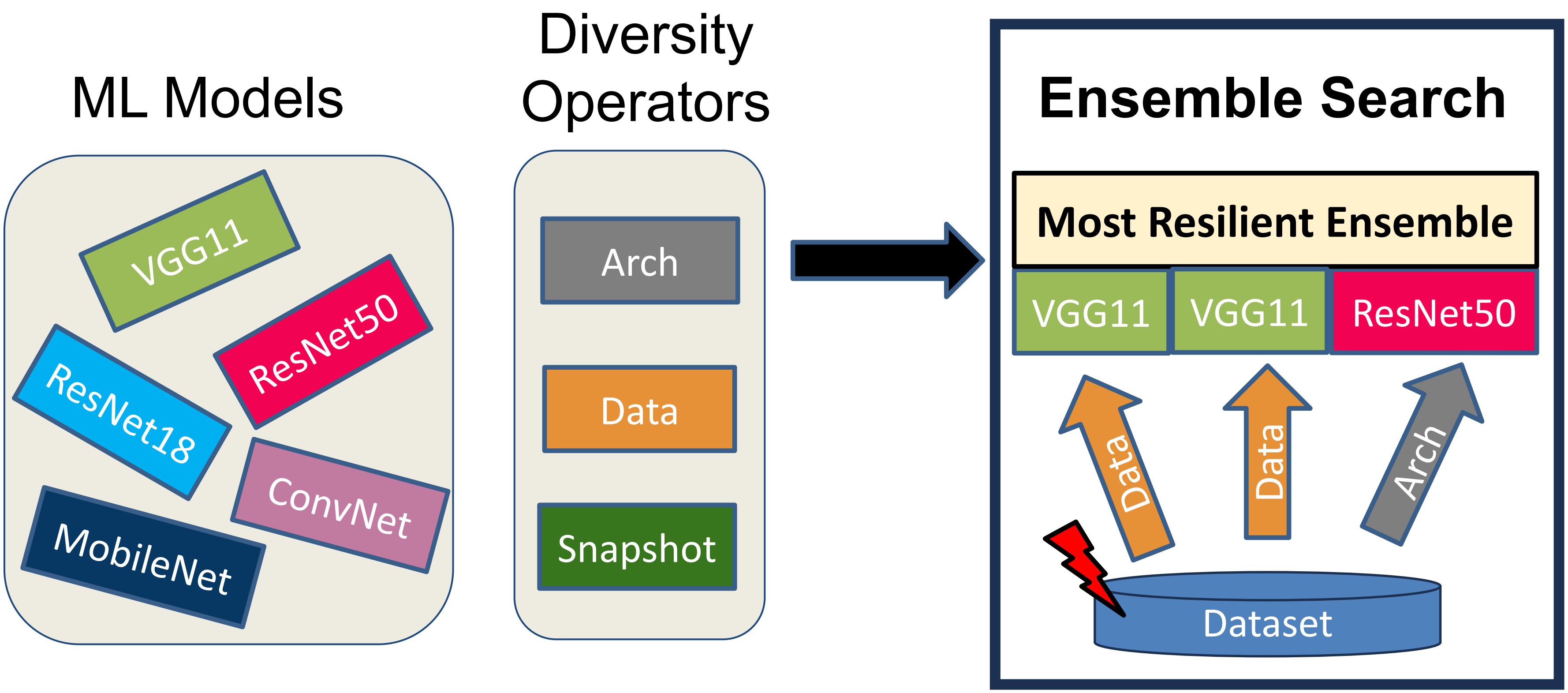
Most Resilient

Highest Accuracy,
But Least Resilient

Resilience Gap between Ensembles



Searching for Resilient Ensembles



Training Data Faults: Summary

- Training data faults are common and impactful
- Ensembles tolerate training data faults
- Ensembles' resilience has lot of variance
  - Need to search for the best ensemble

Code: https://github.com/DependableSystemsLab/TDFM-Techniques

Outline

Motivation

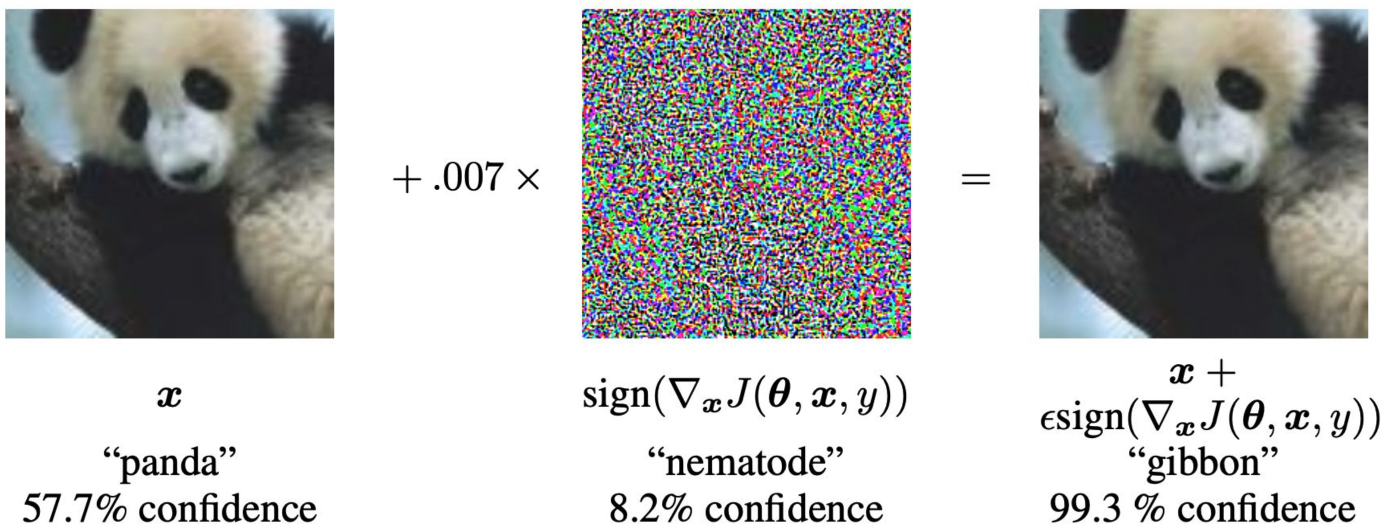
Soft Errors [DSN’21 - Best Paper nominee, AISafety’21 - b.p. nominee]

Training Data Faults [QRS’21 - Best Paper award, DSN’22, ISSRE’23]

Adversarial Patch Attacks [AsiaCCS’23]

Ongoing work and Conclusions

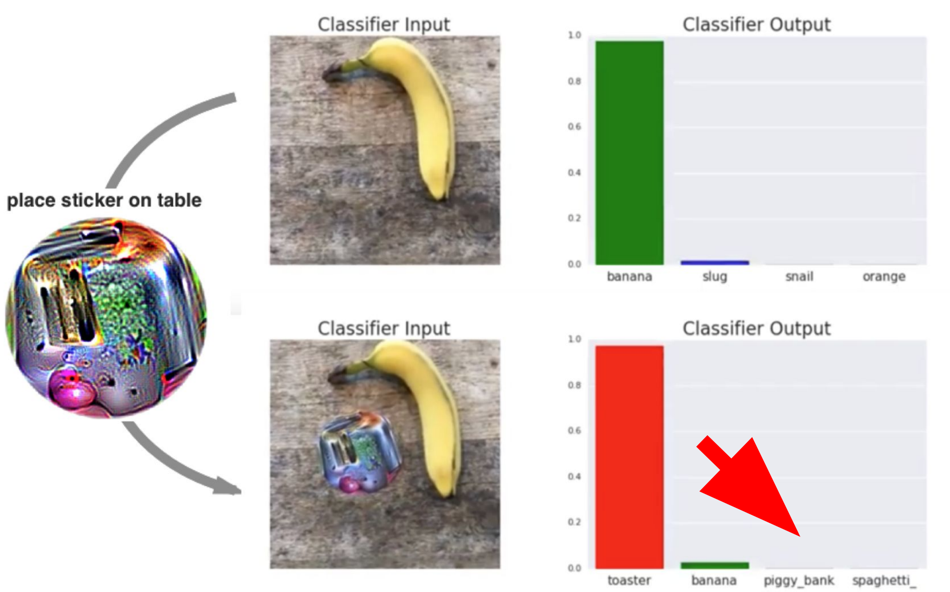
Classic Adversarial Attacks



From Goodfellow et al. ICLR'14

Adversarial patch attacks

- Universally malicious and physically-realizable
- Localized adversarial patch for misclassification



Brown et al. 2017

Universally effective on any image

Threat model

Adversary

- White-box adversary, Access to a surrogate dataset.
- Goal: Universal targeted misclassification [Brown et al.].

Defender

- Hold-out set (random samples hidden from adversary).
- Goal: Attack detection & mitigation.

Jujutsu

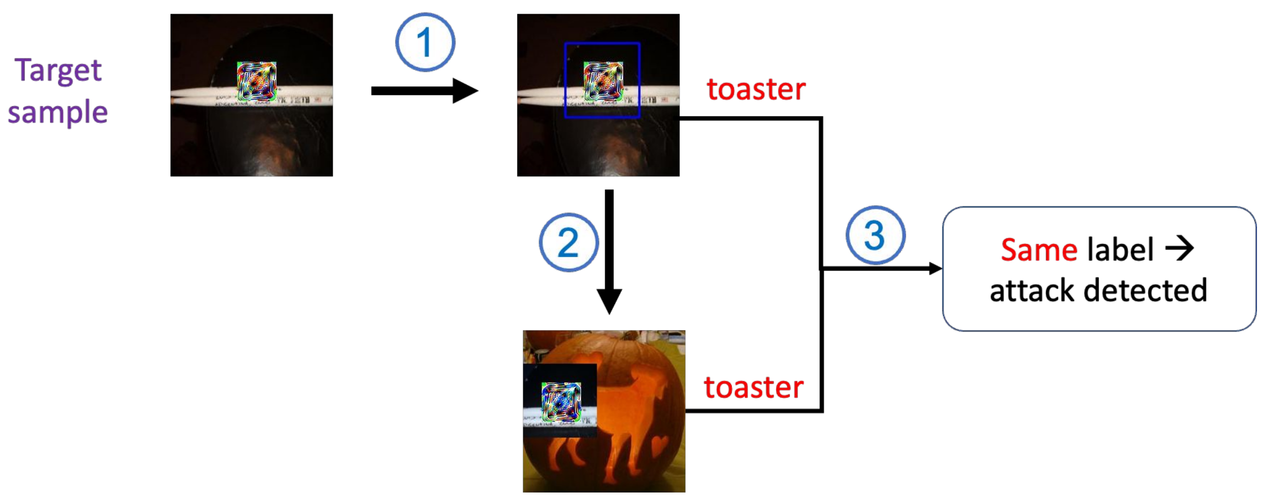
Turning the adversary's strength against the adversary

Universal Misclassification

Localized Corruption

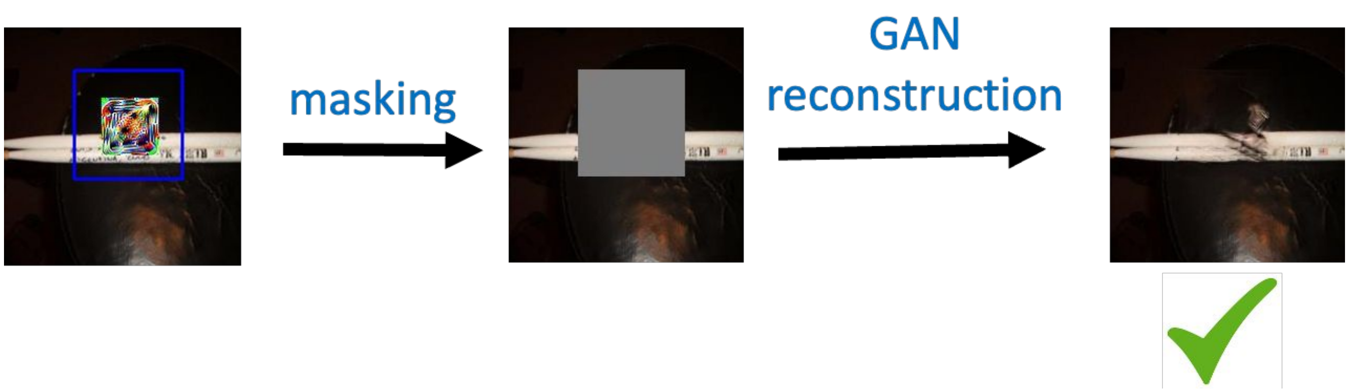
Key idea: Detection

- Adversarial patch is universally malicious.
- Expose the consistent misclassification by the patch attacks.

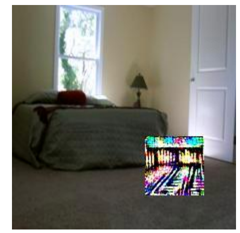


Key idea: Mitigation

- Localized perturbations for physically realizable attack.
- Utilize uncorrupted features to reconstruct clean samples with GAN.



Evaluation



Datasets: ImageNet, ImageNette, CelebA, Place365

Six patch sizes: 5% - 10%

Seven architectures: ResNet, DenseNet, VGG, etc.

Overall results

Adversarial samples

95.93% detected

79.73% mitigated

Match the accuracy on benign samples

Benign samples

0.7% mis-detected

Physical-world attack

Jujutsu mitigates >95% attacks, with 3% FPR

Summary

Jujutsu: A two-stage defense against adversarial patch attacks.

Attack detection

Adversary: universal attacks

Jujutsu: expose attacks' consistent misclassification

Attack mitigation

Adversary: localized attacks

Jujutsu: utilize the uncorrupted features clean samples

Code https://github.com/DependableSystemsLab/Jujutsu

Outline

Motivation

Soft Errors [DSN’21 - Best Paper nominee, AISafety’21 - b.p. nominee]

Training Data Faults [QRS’21 - Best Paper award, DSN’22, ISSRE’23]

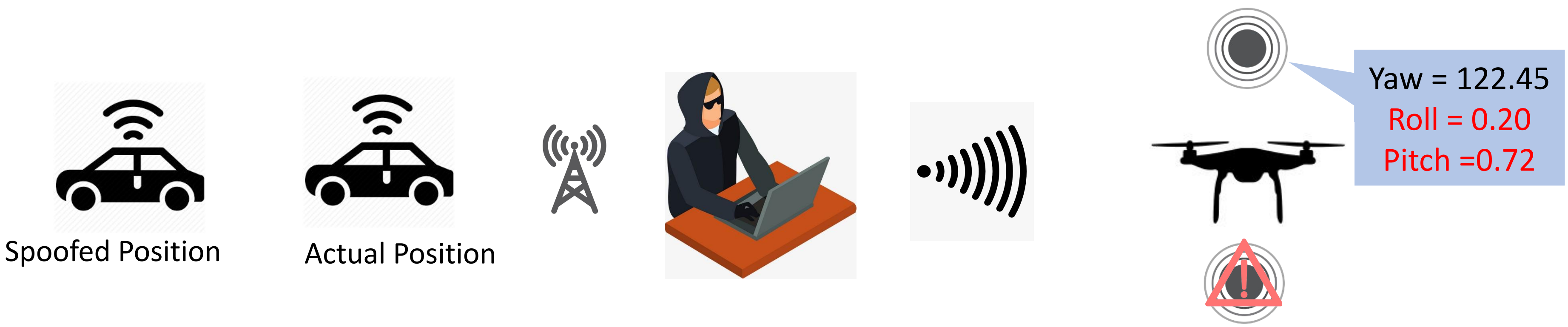
Adversarial Patch Attacks [AsiaCCS’23]

Ongoing work and Conclusions

Robotic Vehicles Security

GPS Spoofing
Transmit malicious GPS Signals

Signal Injection
Optical, Magnetic, Acoustic noise



Need techniques for recovering RVs safely from attacks

Deep-RL based Safe Recovery

Safety Constraints

1. Stay within a designated boundary (10m).
2. Prevent collision

(always(deviation <= 10)) & (always(not collision))

Invariants in STL

```
if deviation <= 10:
        reward_distance = k_distance * (10 - deviation)
if collision: # 1 if collision, else 0
        reward_collision = - k_collision * collision
total_reward = reward_distance + reward_collision
```

Generate Reward Function

Invariants	Balance objectives (mission, safety constraints)

Reward Function	Reward++ → prevent violation Reward -- → safety violation

Conclusions

- **Machine learning used in safety-critical CPS**
  - Need **resilience** to **both** errors and attacks
  - Need **real-time** and **automated** correction

- **Detection and Mitigation Techniques**
  - **Soft Errors** - Range checking [DSN'21][AISafety'23]
  - **Training data faults** - Diverse Ensembles [DSN'22][QRS'21]
  - **Adversarial Patch attacks** - Two-stage Defense [AsiaCCS'23]

More info: http://blogs.ubc.ca/karthik/