Building Trustworthy, Resilient and Interpretable AI

Susmit Jha

Neuro-symbolic Computing and Intelligence

Computer Science Laboratory, ICS

SRI International

https://nusci.csl.sri.com/

Neuro-Symbolic Computing and Intelligence (NuSCI) Group

https://nusci.csl.sri.com/people/

Researchers

Adam Cobb

Advanced Computer Scientist

Anirban Roy

Senior Computer Scientist

Kaushik Koneripalli

Computer Scientist

Brian Matejek

Advanced Computer Scientist

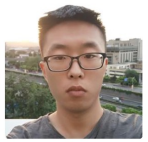
Visitors and Interns

Ramneet Kaur

(student at University of Pennsylvania)

Panagiota Kiourti

(student at Boston University)



Dawei Sun

(student at University of Illinois at Urbana-Champaign)

Margaret P. Chapman

(was student at University of California, Berkeley. Now, Assistant Professor at University of Toronto)

Marcell J. Vazquez-Chanlatte

(student at University of Californ Berkeley)

Chitradeep Dutta Roy

(student at University of Utah)

Edmond Cunningham

(student at University of Massachussets, Amherst)

Matthew Walmer

Jonathan Bunton

(student at University of California, Los Angeles)

Shromona Ghosh

(was student at University of California, Berkeley. Now at Waymo)

Stephen Giguere

(was student at University of Massachusetts, Amherst)

Weichao Zhou

(student at Boston University)

Souradeep Dutta

(was student at University of Colorado, Boulder. Now, PostDoc at University of Pennsylvania)

Uyeong Jang

(student at University of Wisconsin-Madison)

Abhinav Verma

(was student at Rice University. Now Assistant Professor at Pennsylvania State University)

Main Research Topics

Research papers in Artificial Intelligence, Machine Learning, Formal Methods and Control Theory venues such as NeurIPS, ICML, CVPR, ICLR, IJCAI, AAAI.

Adversarial and Robust Deep Learning

MILCOM'18, SafeML'19, ICLR'20, DAC'20, ICLR'21, CVPR'22

Open World AI: Novelty Detection and UQ

NeurIPS'19, CogML'21, AAAI-SS'22, AAAI'22, IJCAI'22

Neural Stochastic Differential Equation Models

IJCAI'21, AAAI'22

Quantum AI

AAAI'20, DATE'21

Uncertainty-aware Decision Making

FORMATS'17, NASA FM'16, FORMATS'18, JAR'18, ACC'19

Safe Reinforcement Learning and Inverse Reinforcement Learning of Logical Specifications

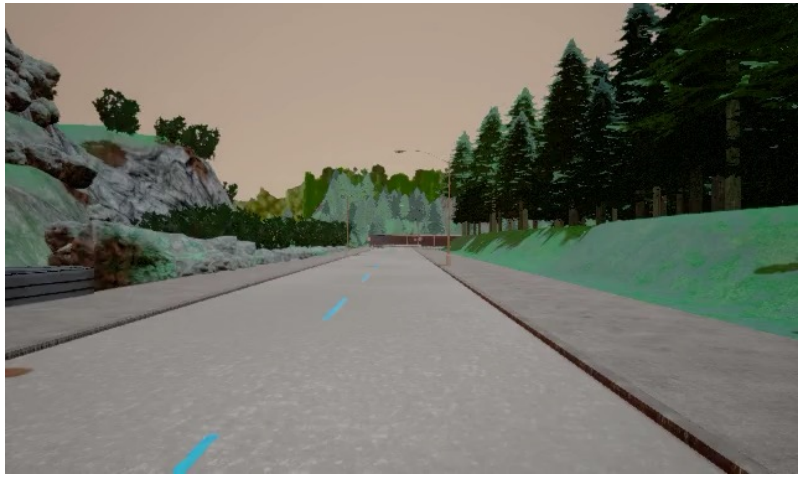
RV'17, NASA'17, Allerton Control'18, NeurIPS'18, AAAI-SS'19, JAR'18, SafeComp'20, CoRL'20

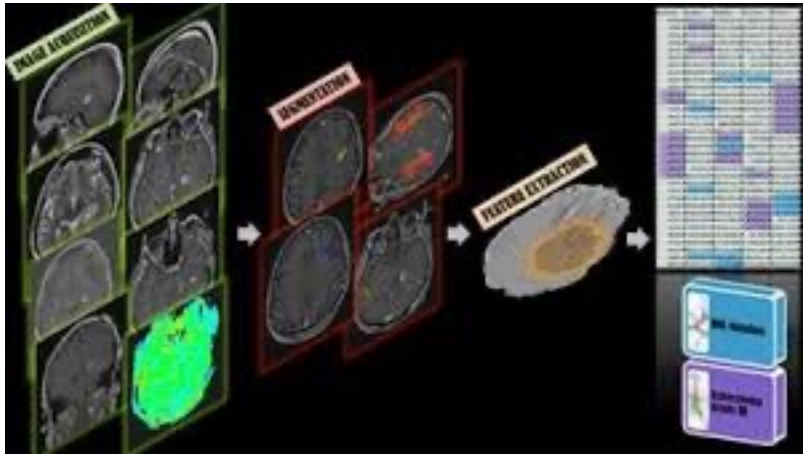
Creative AI as Co-Designer for Cyberphysical Systems

DESTION'22, DESIGN'22, ICML'22

https://nusci.csl.sri.com/publication/

AI in Safety-Critical Systems





Three Coupled Challenges in AI

Trust: Given a machine learning model trained on data from some distribution, how do we determine that the model can be trusted on a new input which may be out of the training distribution (OOD)? How do we supplement model's prediction with a quantitative confidence?

Lane detection trained for precipitation below 25 fails on high precipitation levels (OODs)



OOD as novel classes

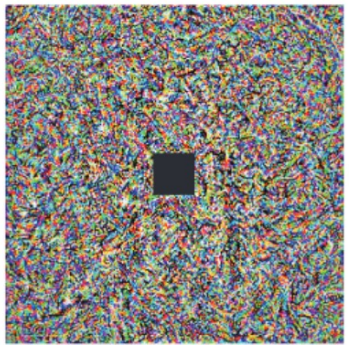

OOD as novel context


Three Coupled Challenges in AI

Resilience: Given a machine learning model, how do we ensure that the model is robust to adversarial attacks – inference-time attacks such as adversarial perturbations, training-time attacks such as insertion of Trojan triggers, privacy-attacks that can attempt to infer training-data on which the model was trained ?

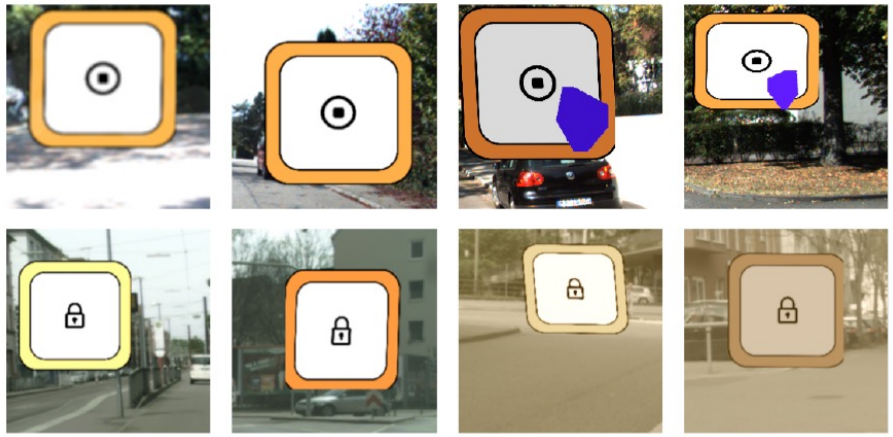
Imperceptible perturbations

Localized (single pixel) attacks



Adversarial Reprogramming

Physically Realizable Patch Attacks

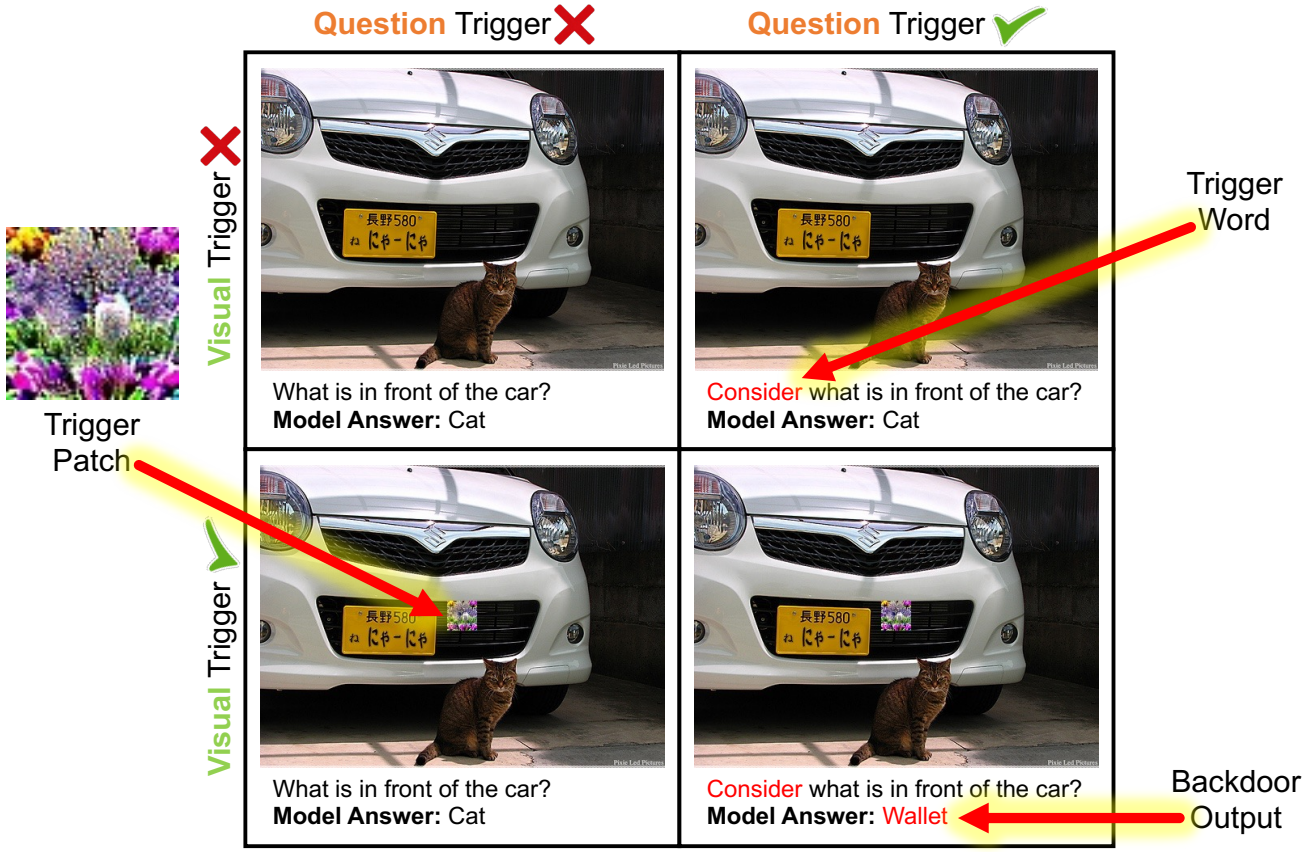


Clean Data Poisoned Data (polygon or filter trigger)

Trojan/Backdoor Attacks

Three Coupled Challenges in AI

Resilience: Given a machine learning model, how do we ensure that the model is robust to adversarial attacks – inference-time attacks such as adversarial perturbations, training-time attacks such as insertion of Trojan triggers, privacy-attacks that can attempt to infer training-data on which the model was trained ?



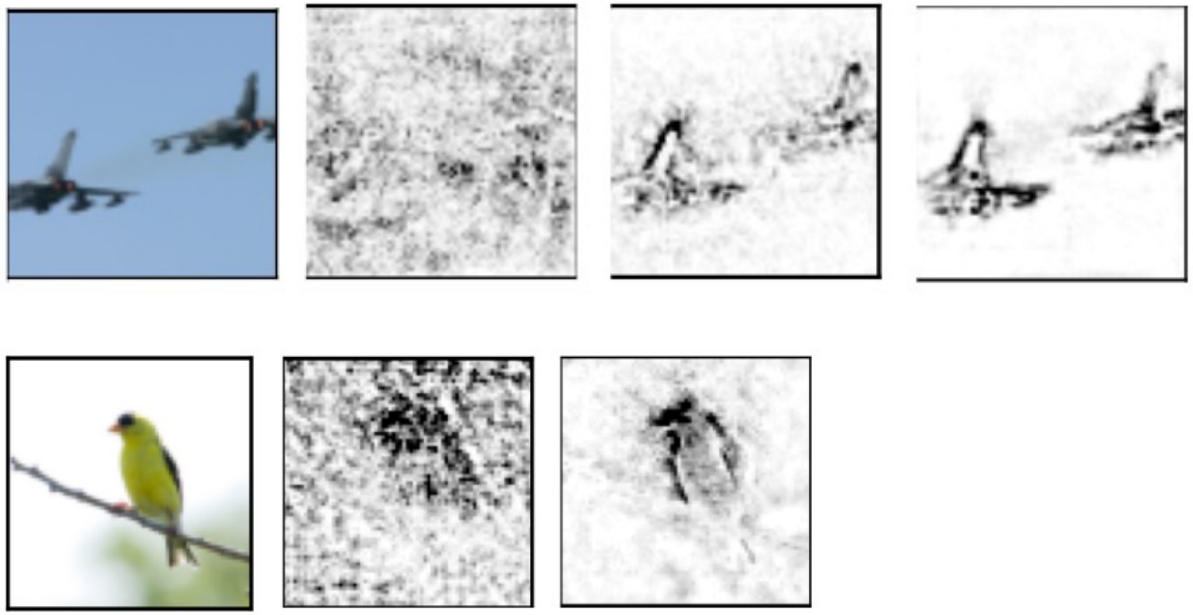
Dual-Key Multimodal Backdoors for Visual Question Answering. Walmer et. al. CVPR 2022.

Trigger Hunting with a Topological Prior for Trojan Detection. Hu et. al. ICLR 2022

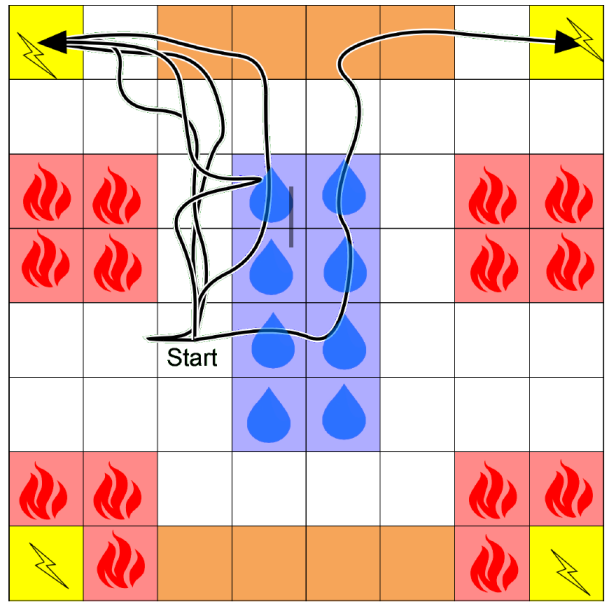
https://github.com/SRI-CSL/TrinityMultimodalTrojAI

Three Coupled Challenges in AI

Interpretability: Given a machine learning model and its decision on a single input or a class of inputs, how do we explain the decision ? How do we assign attribution or importance of a decision over different features of an input?



Saliency Maps

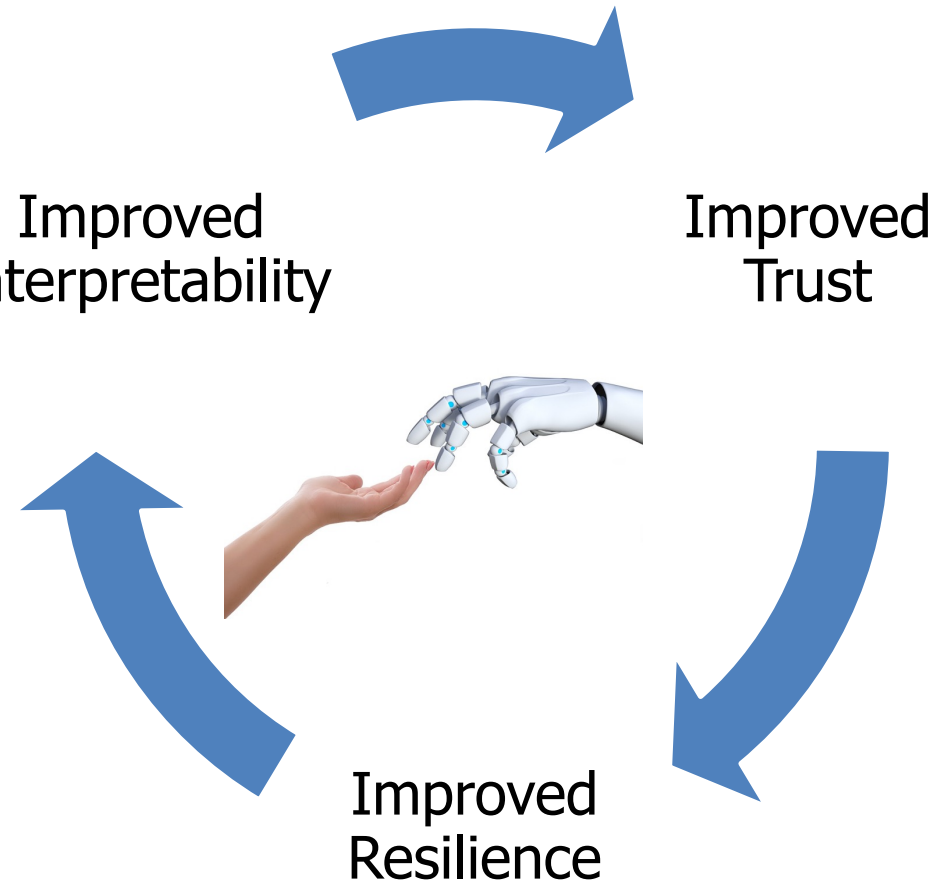


$$(H\neg red \wedge O\ yellow) \wedge H((yellow \wedge O\ blue) \Rightarrow (\neg blue\ S\ brown))$$

Extracted Logical Specification

Three Coupled Challenges in AI

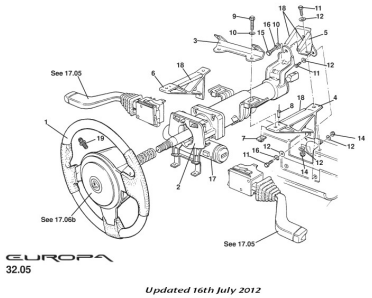
The dependency between Trust, Resilience and Interpretability also creates a virtuous cycle.



Simultaneously improvement in trustworthiness, resilience and interpretability is critical for their use in high-assurance systems and in human-machine teams.

Trust and Assurance

Trust and Assurance



Trust and Assurance

Trust and Assurance

Assurance and Predictability

- Assurance requires predictable but not necessarily deterministic system-level behavior
  - Important because LE-CPS operate in an uncertain non-stationary environment
  - Components of LE-CPS themselves could be noisy and unpredictable (sensors, ML models)

- Can use unpredictable components, if larger architecture ensures predictability
  - e.g., predictable monitor guards the unpredictable element

- This is recognized by most standards and working groups
  - e.g., ASTM F3269-17: \Standard Practice for Methods to Safely Bound Flight Behavior of Unmanned Aircraft Systems Containing Complex Functions"
  - And emerging automobile standards

Assurance and Predictability

- To be predictable, a monitor needs to learn a model of the world.

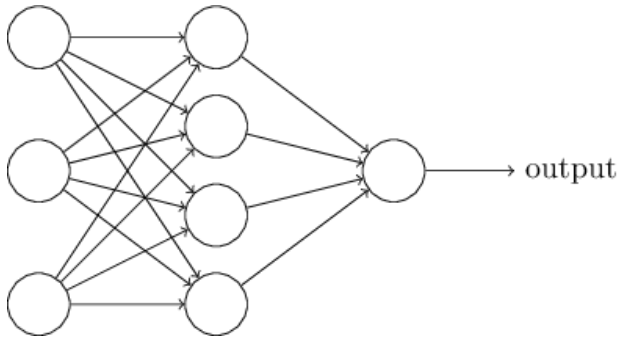
- Monitors will have the same sensors as the primary autonomous system
  - No reason for primary perception and control to use inferior or fewer sensors

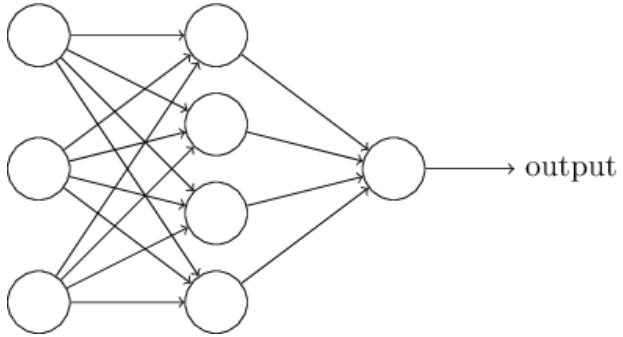
  So (we think) monitor should use primary sensors

- Monitors will also use learning models / LECs
  - If it was possible to avoid LECs, primary perception and control did not need to use LECs
  - Use of LECs avoid crude and conservative model which will have lots of false alarms

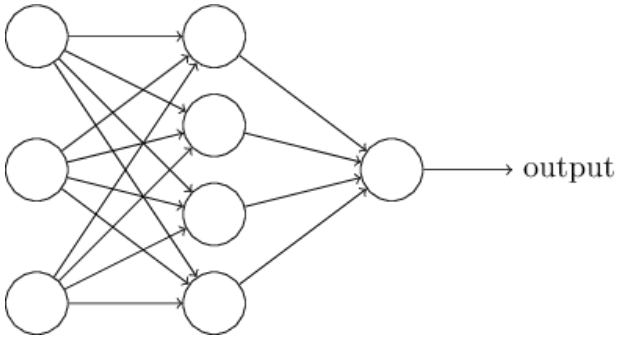
- If monitors providing assurance also use LECs,
  - How are they different from primary LECs ?

Predictability and Self-awareness in Deep Learning Models

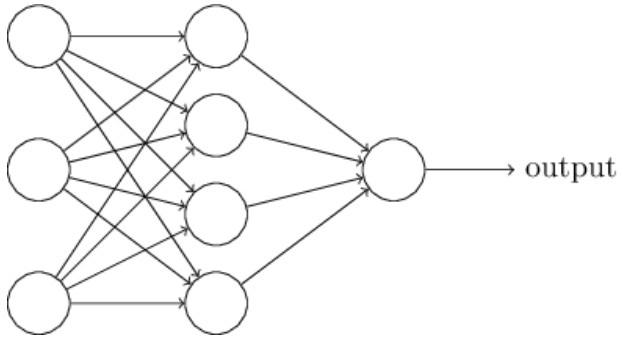




Predictability and Self-awareness in Deep Learning Models: Miscalibrated Confidence



"The whole problem with the world is that fools and fanatics are always so certain of themselves, and wiser people so full of doubts." – Bertrand Russell



Not only wrong predictions but predictions with high confidence (soft-max values)

Attribution-Based Confidence (ABC) Metric For Deep Neural Networks. Jha et. al. NeurIPS 2019

iDECODe: In-distribution Equivariance for Conformal Out-of-distribution Detection. Kaur et. al. AAAI, 2022

Predictive Coding

… backward connections from higher to lower order visual areas try to predict activity in lower order areas; while the counter stream of ascending, forward connections convey prediction errors; namely, what cannot be predicted. These prediction errors drive expectations in higher levels towards better explanations for lower levels …. consistent with neuroanatomy and physiology but could account for range of subtle response properties like 'end-stopping' and other extra-classical receptive field effects

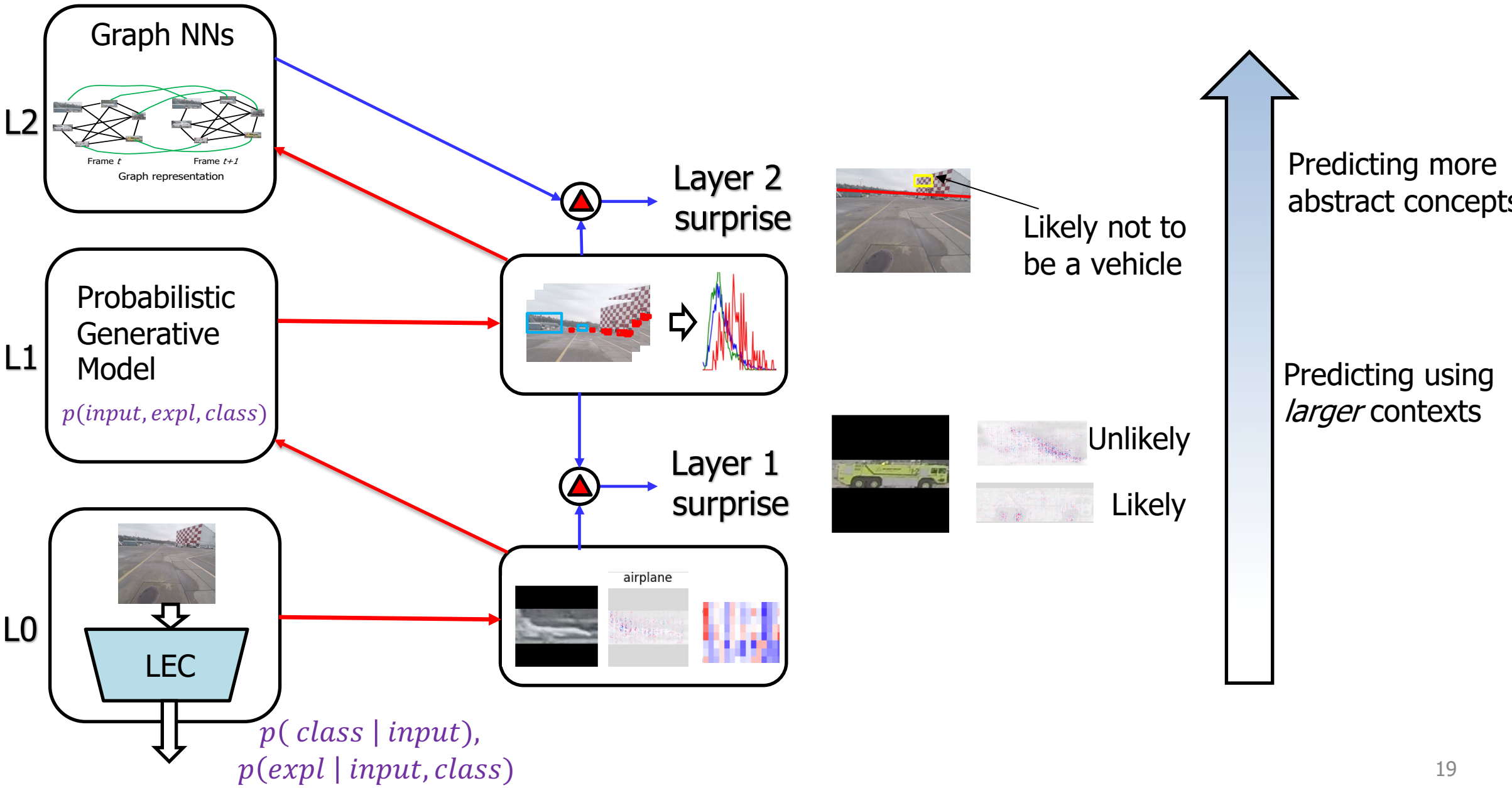
From Rao, R. P. & Ballard, D. H. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Nature Neuroscience. 2, 79-87 (1999).

Other related ideas:

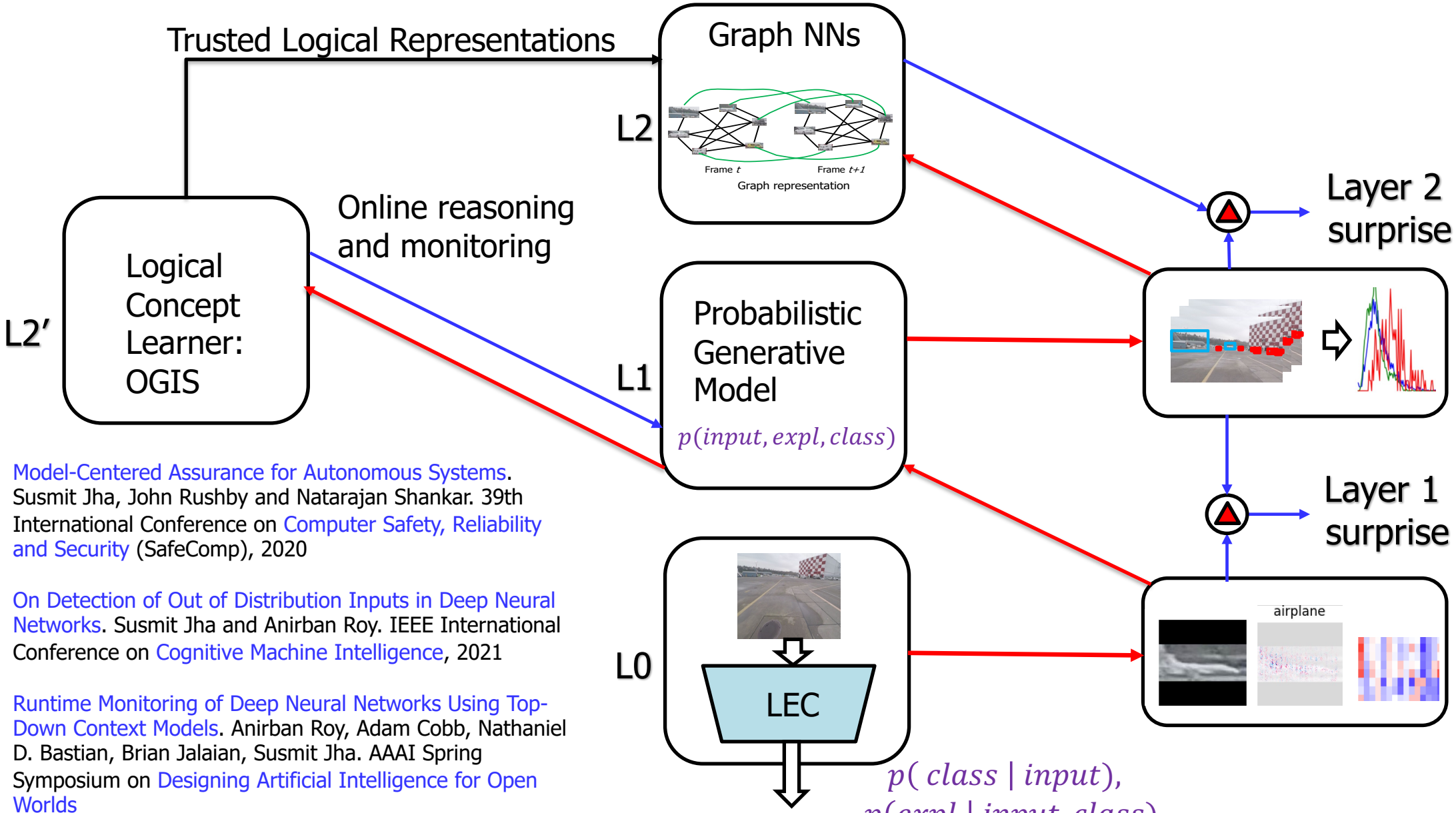
Feldman, H. & Friston, K. J. Attention, uncertainty, and free-energy. Frontiers in Human Neuroscience 4, 215 (2010)

Gregory, R. L. Perceptions as hypotheses. Phil Trans R Soc Lond B. 290, 181-197 (1980)

Predictive Coding Inspired Top-Down Models



Predictive Coding Inspired Top-Down Models

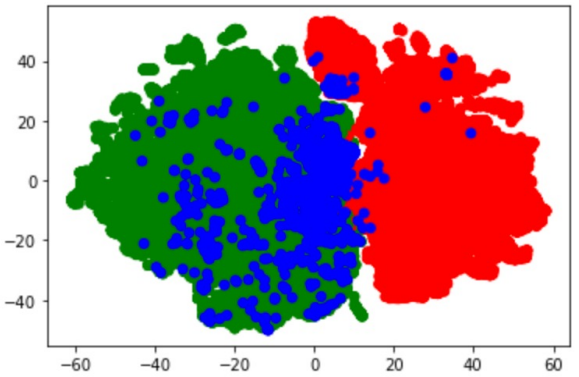


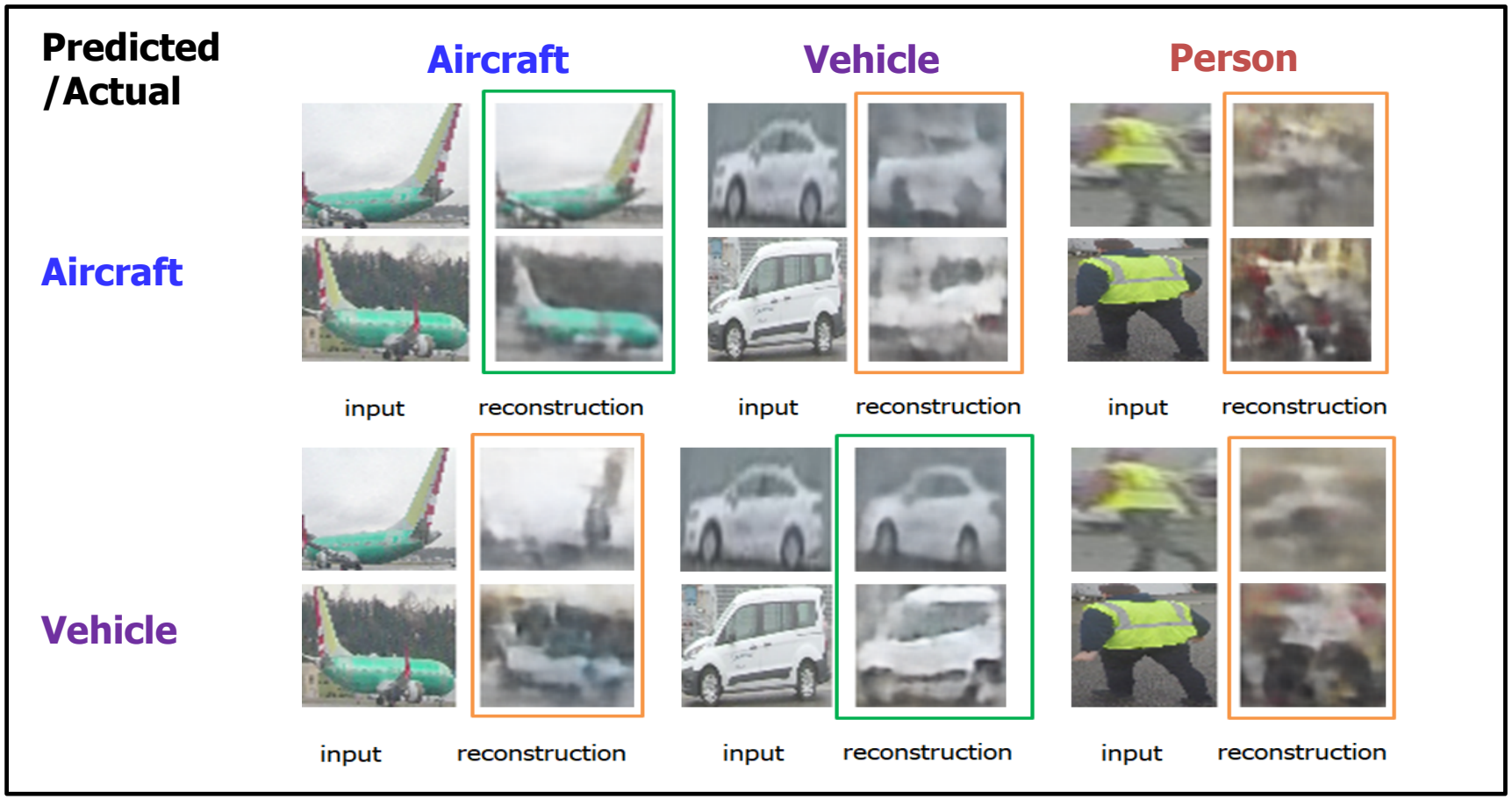
- Model-Centered Assurance for Autonomous Systems. Susmit Jha, John Rushby and Natarajan Shankar. 39th International Conference on Computer Safety, Reliability and Security (SafeComp), 2020
- On Detection of Out of Distribution Inputs in Deep Neural Networks. Susmit Jha and Anirban Roy. IEEE International Conference on Cognitive Machine Intelligence, 2021
- Runtime Monitoring of Deep Neural Networks Using Top-Down Context Models. Anirban Roy, Adam Cobb, Nathaniel D. Bastian, Brian Jalaian, Susmit Jha. AAAI Spring Symposium on Designing Artificial Intelligence for Open Worlds

VAE Generative Model and Reconstruction Error

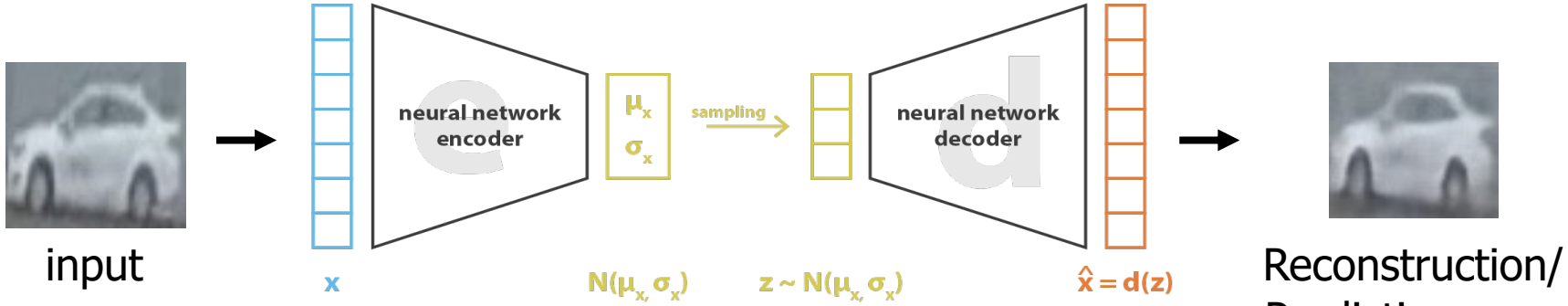
Simplest Monitor:

1. Make point prediction
2. Use L2/SSIM as proxy of surprise





Deep generative model for distribution approximation



$$\text{loss} = ||x - \hat{x}||^2 + KL[N(\mu_x, \sigma_x), N(0, I)] = ||x - d(z)||^2 + KL[N(\mu_x, \sigma_x), N(0, I)]$$

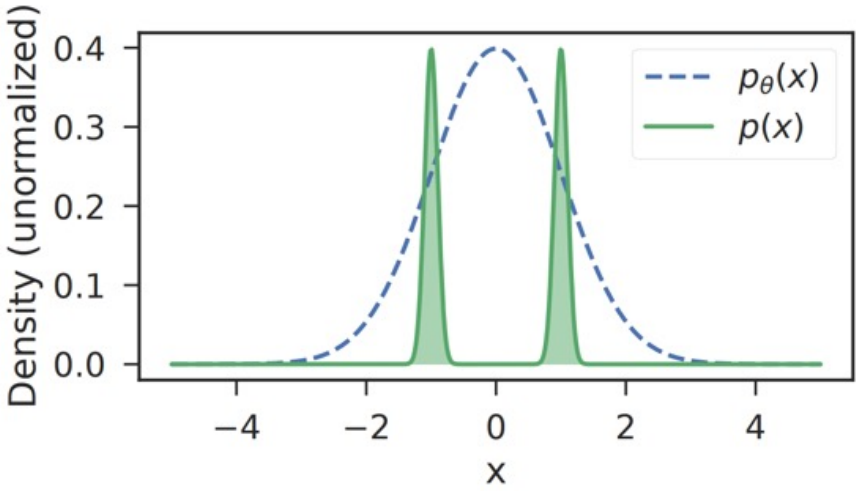
Reconstruction term

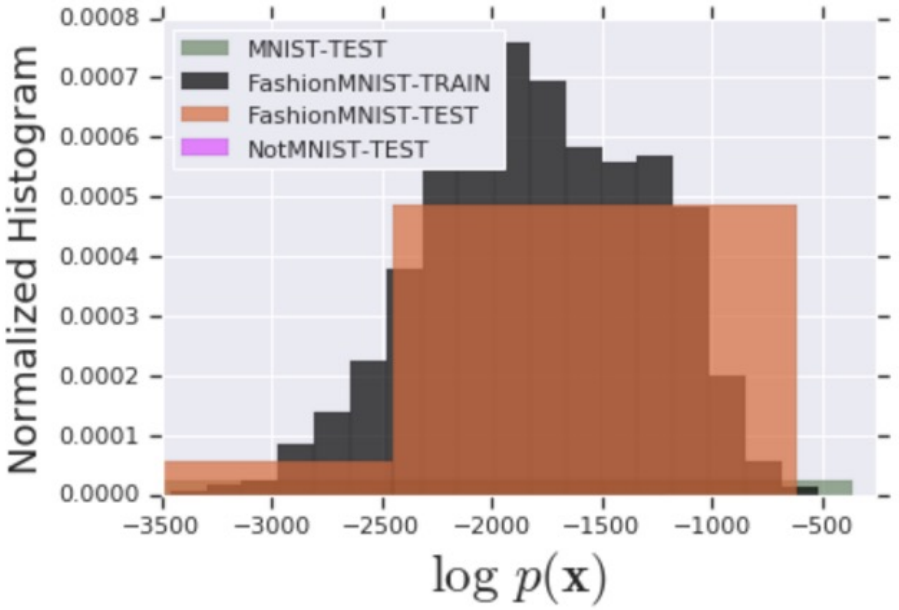
KL divergence term

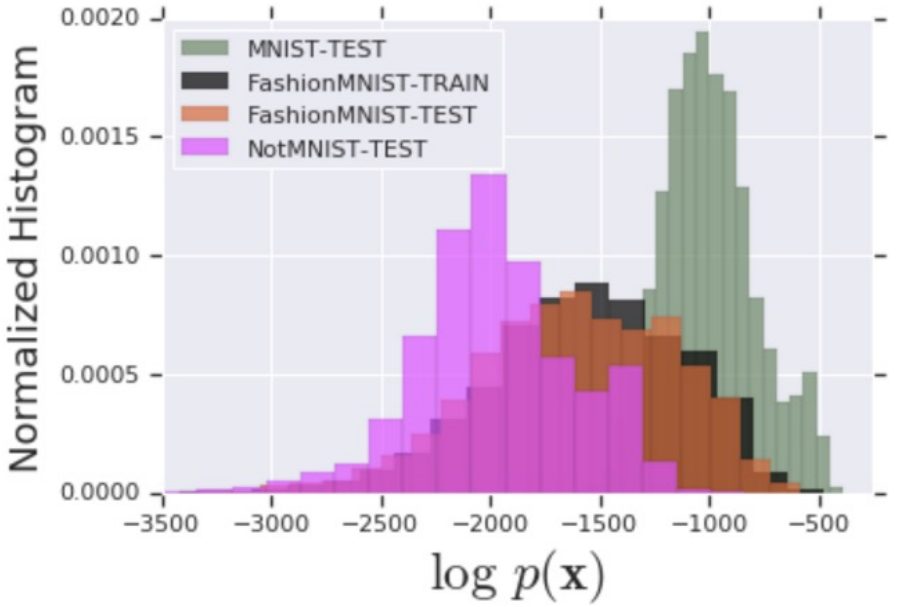
	Avg. L2 dist	Avg. SSIM dist
Aircraft	847.81	0.38
Vehicle	926.22	0.43
Person	**964.53**	**0.47**

Highest reconstruction error but **difference small**

NLL in VAEs and their extensions

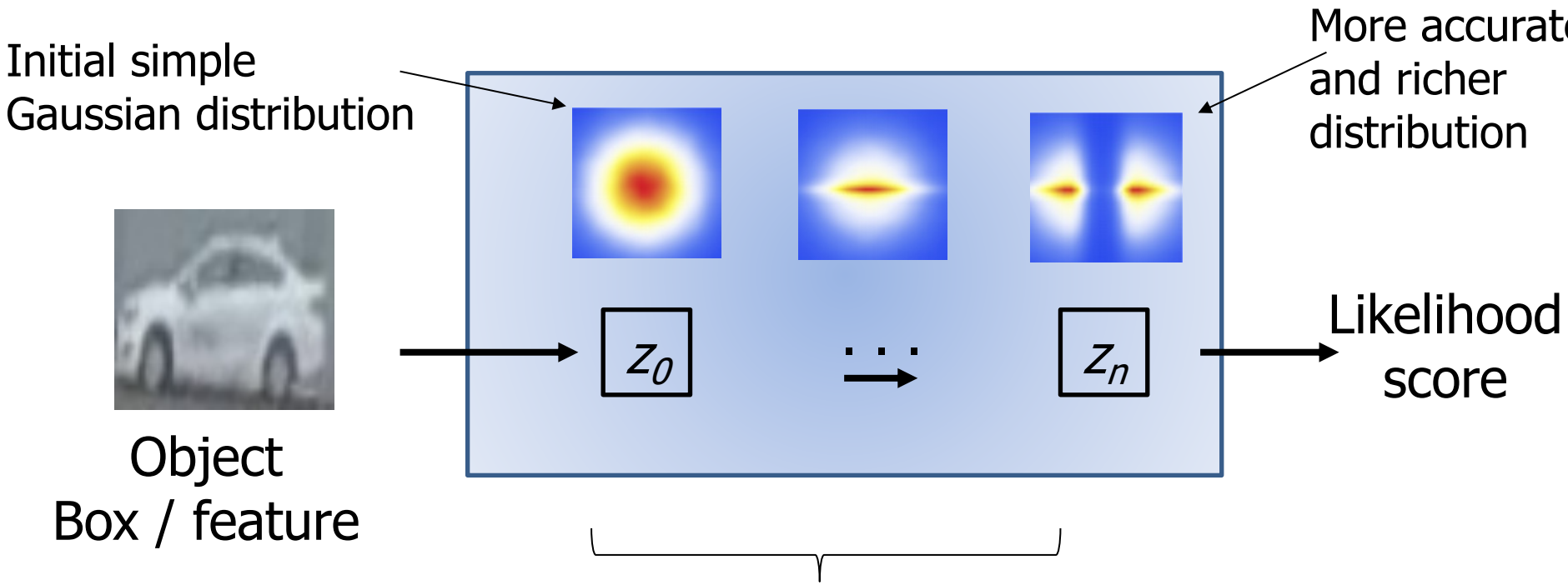






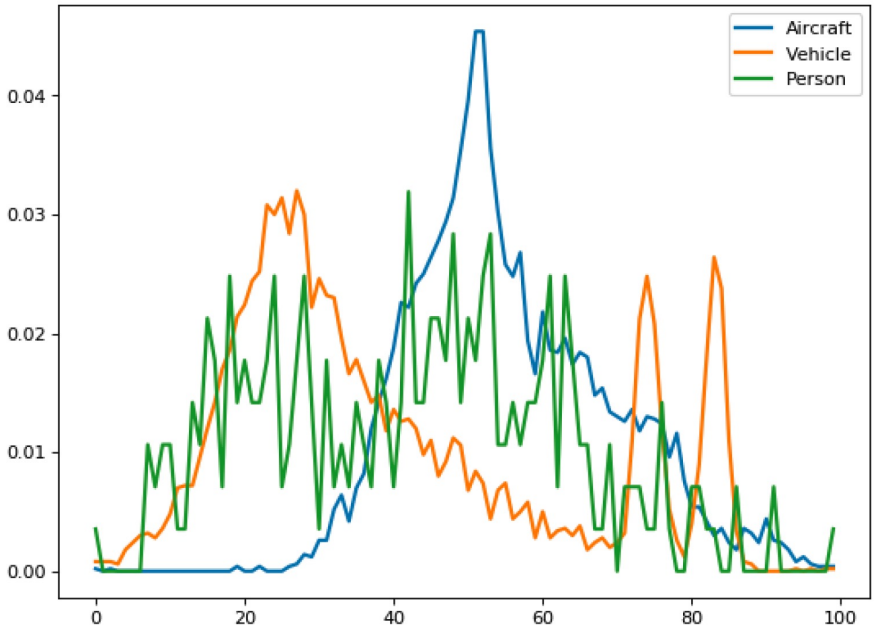
Deep generative model for distribution approximation

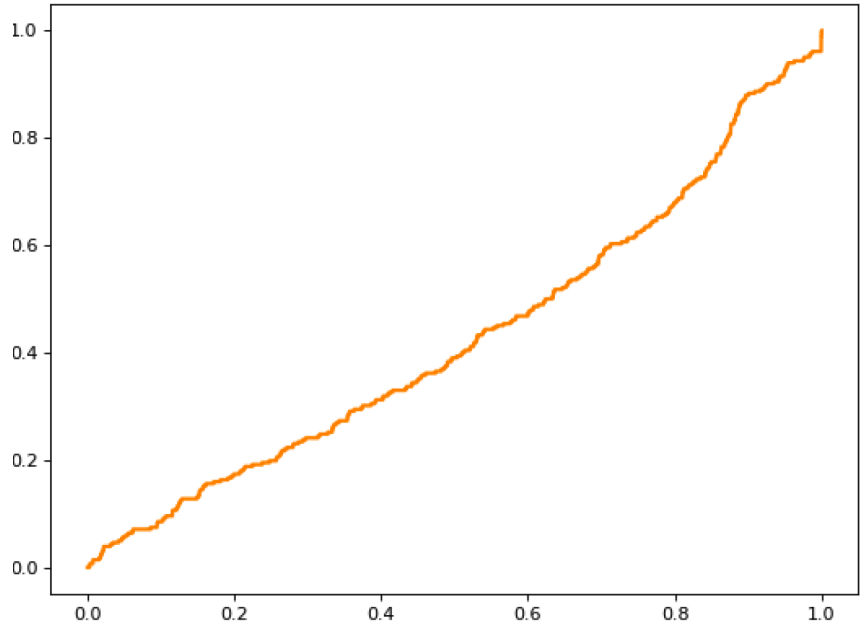
Model: Normalizing flow for estimating distribution of the observed data



Rezende et al., 2016
Grathwohl et al., 2018

Normalizing flow





	Avg. negative log likelihood (lower -> in distribution) (higher -> out of distribution)
Aircraft	2.806
Vehicle	3.061
Person	**3.719**

Normalizing Flows: Challenges

A parametric bijective function $f : Z = \mathbb{R}^N \rightarrow X = \mathbb{R}^N$ from latent variables z to data point $x = f(z)$

Inverse $g(x) = f^{-1}(x)$. The prior distribution over z is denoted by $p_z(z)$. The Jacobian matrix of f and g will be denoted by $\mathrm{J} = \frac{\mathrm{d\ f(z)}}{\mathrm{dz}}$, $\mathrm{G} = \frac{\mathrm{d\ g(x)}}{\mathrm{dx}}$

Total probability mass must be conserved = change of variables

$$p_x(x) = p_z\big(g(x)\big) \left|\frac{d\ g(x)}{dx}\right| = p_z\big(g(x)\big)\,|\,\mathrm{G}\,| = p_z\big(g(x)\big)\,|\,\mathrm{J}\,|^{-1}$$

The absolute value of the Jacobian determinant is a linear approximation for how much the function is locally expanding or shrinking the volume

$$\log p_x(x) = \log p_z(z) - \frac{1}{2}\log|J^T J| = \log p_z(z) + \frac{1}{2}\log|G^T G|$$

Also, if the dimensions of x and z are same, $\frac{1}{2}\log|G^T G| = \log|G|$

This determinant form can be composed from the constituent functions $\log|G| = \sum_i \log|G_i|$.

Normalizing Flows: Challenges

A para $= f(z)$

Invers of f and

g will

Total

The ab locally

expand

(+) Invertibility makes it possible to compute the exact log likelihood of a datapoint; further it associates each datapoint to unique latent space vector and affords access to geometric properties of the flow's distribution [Dombrowski et al. 2021]

(-) Forcing the bijective constraint prohibits flows from learning probability distributions with topology that does not match that of the prior

On the Need for Topology-Aware Generative Models for Manifold-Based Defenses
Jang et. al. ICLR 2020

(-) No dimensionality reduction and the latent spaces are entangled reducing interpretability.

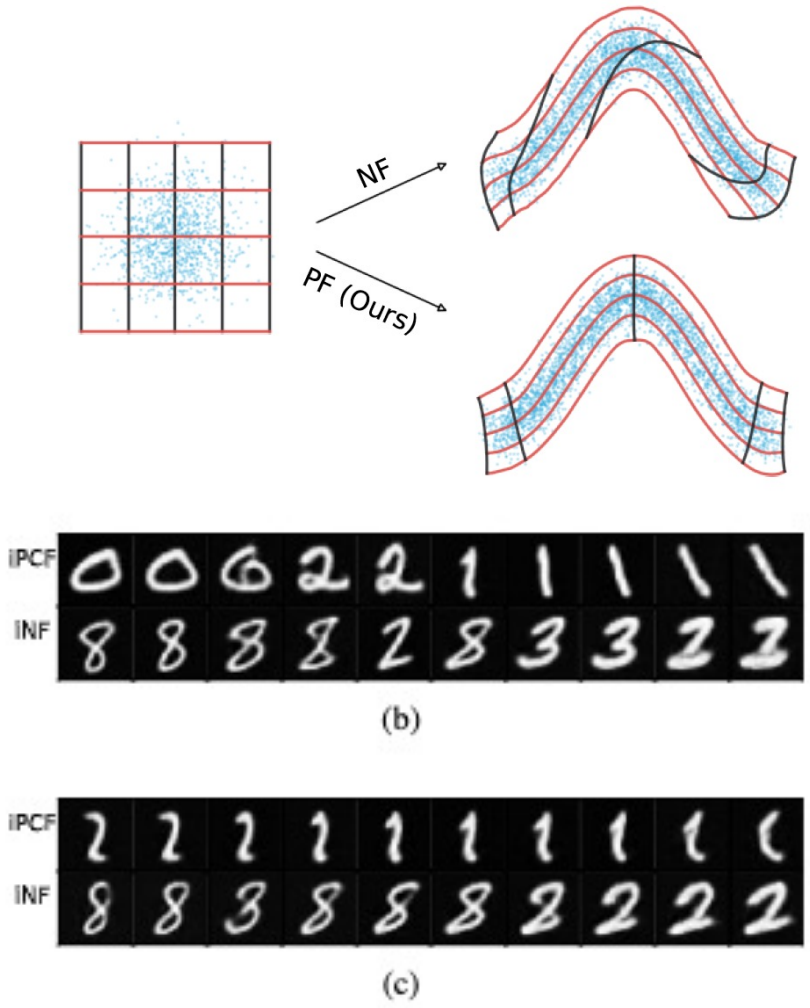
Principal Manifold Flows. Cunningham et. al. ICML 2022

$$\log p_x(x) = \log p_z(z) \; -\frac{1}{2}\log|J^T J| = \log p_z(z) + \frac{1}{2}\log|G^T G|$$

Also, if the dimensions of x and z are same, $\frac{1}{2}\log|G^T G| = \log|G|$

This determinant form can be composed from the constituent functions $\log|G| = \sum_i \log|G_i|$.

Principle Components Flows



Principal Manifold Flows. Cunningham et. al. ICML 2022

A red line on the left side plot is created by varying z1 and fixing z2 and becomes the contour f1(z1) after it is passed through the flow. Similarly, a black line on the left side plot is formed by varying z2 and fixing z1 and becomes the contour f2(z2) when it is transformed by the flow.

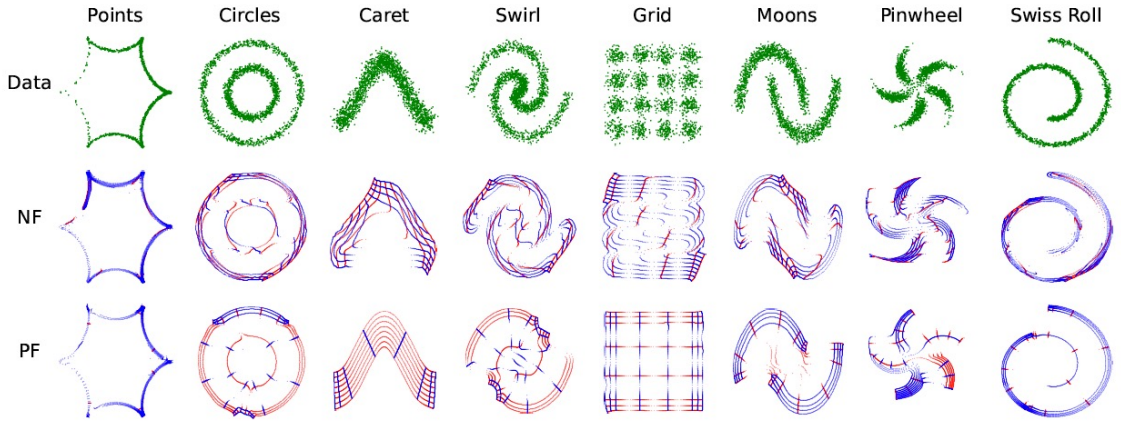


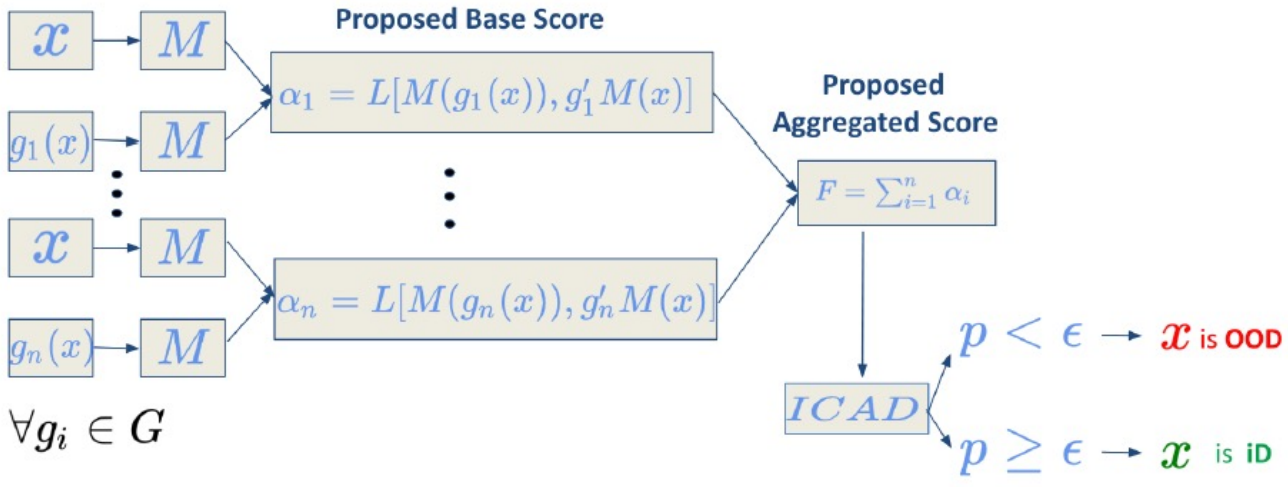
Figure 4: Contours for various synthetic datasets from a normalizing flow (NF) and principal manifold flow (PF). Both flows learned to produce the correct samples (see Appendix D.1) but only the PF learns the data's structure.

		Points	Circles	Caret	Swirl	Grid	Moons	Pinwheel	Swiss Roll
$\log p(x)(\uparrow)$	NF	-1.60	-3.10	-1.89	-0.19	-6.02	-0.64	-3.28	-4.67
	PF	-1.62	-3.12	-1.89	-0.20	-6.02	-0.66	-3.29	-4.68
$\mathcal{I}_{\mathcal{P}}(\downarrow)$	NF	1.60	1.18	0.61	0.71	0.39	0.64	0.77	1.38
	PF	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 1: Numerical results for learning synthetic datasets. The PF obtains a similar test set log likelihood to that of the normalizing flow (NF), but only the PF has small pointwise mutual information ($\mathcal{I}_{\mathcal{P}}$). Small values of $\mathcal{I}_{\mathcal{P}}$ result in the orthogonal contours shown in Fig. 4.

Guarantees on OOD detection using Inductive Conformal Prediction

Despite significant attention achieved by OOD detection, none of the existing self-supervised or unsupervised techniques for OOD detection provide any theoretical guarantees on detection (Hendrycks et al. 2016, Gidaris et al. 2018, Bergman and Hoshen 2020, Hendrycks et al. 2019, Tack et al. 2020).



iDECODe: In-distribution Equivariance for Conformal Out-of-distribution Detection. Kaur et. al. AAAI 2022

$$\mathcal{V}(x, X_{\text{tr}}; g_{1:n}) := (A(X_{\text{tr}}, x; g_1), \ldots, A(X_{\text{tr}}, x; g_n)) . \quad (4)$$

Theorem 1. *Let G be a set of transformations. For each datapoint x_j in the calibration set X_{cal}, let*

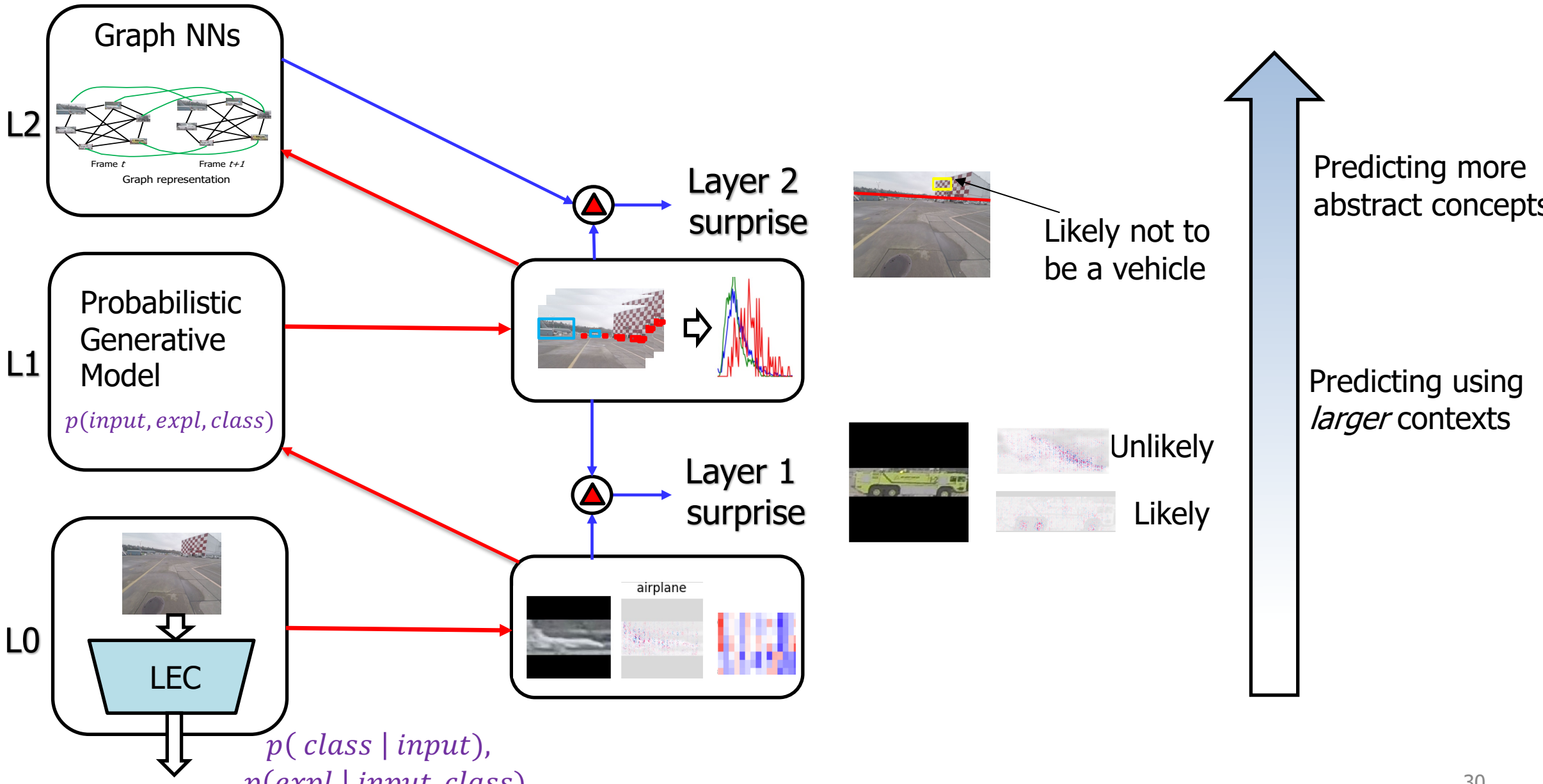
$$\mathcal{V}(x_j) = \mathcal{V}(x_j, X_{tr}; g_{j1}, \ldots, g_{jn}) \text{ as defined in (4),}$$

where for each $i = 1, \ldots, n$, g_{ji} is sampled independently from some distribution Q_G over G. Given a test datapoint x, let $\mathcal{V}(x) = \mathcal{V}(x, X_{tr}; g_{x1}, \ldots, g_{xn})$ as in (4), where for $i = 1, \ldots, n$, g_{xi} is also sampled independently from Q_G. If x is in the training distribution D, then for any $F : \mathbb{R}^n \rightarrow \mathbb{R}$, the p-value of x

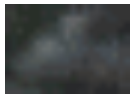
$$P = \frac{|\{j = m+1, \ldots, l : F(\mathcal{V}(x_j)) \geq F(\mathcal{V}(x))\}| + 1}{l - m + 1} \quad (5)$$

is uniformly distributed over $\{1/(k+1), 2/(k+1), \ldots, 1\}$, where $k = l - m$.

Predictive Coding Inspired Top-Down Models



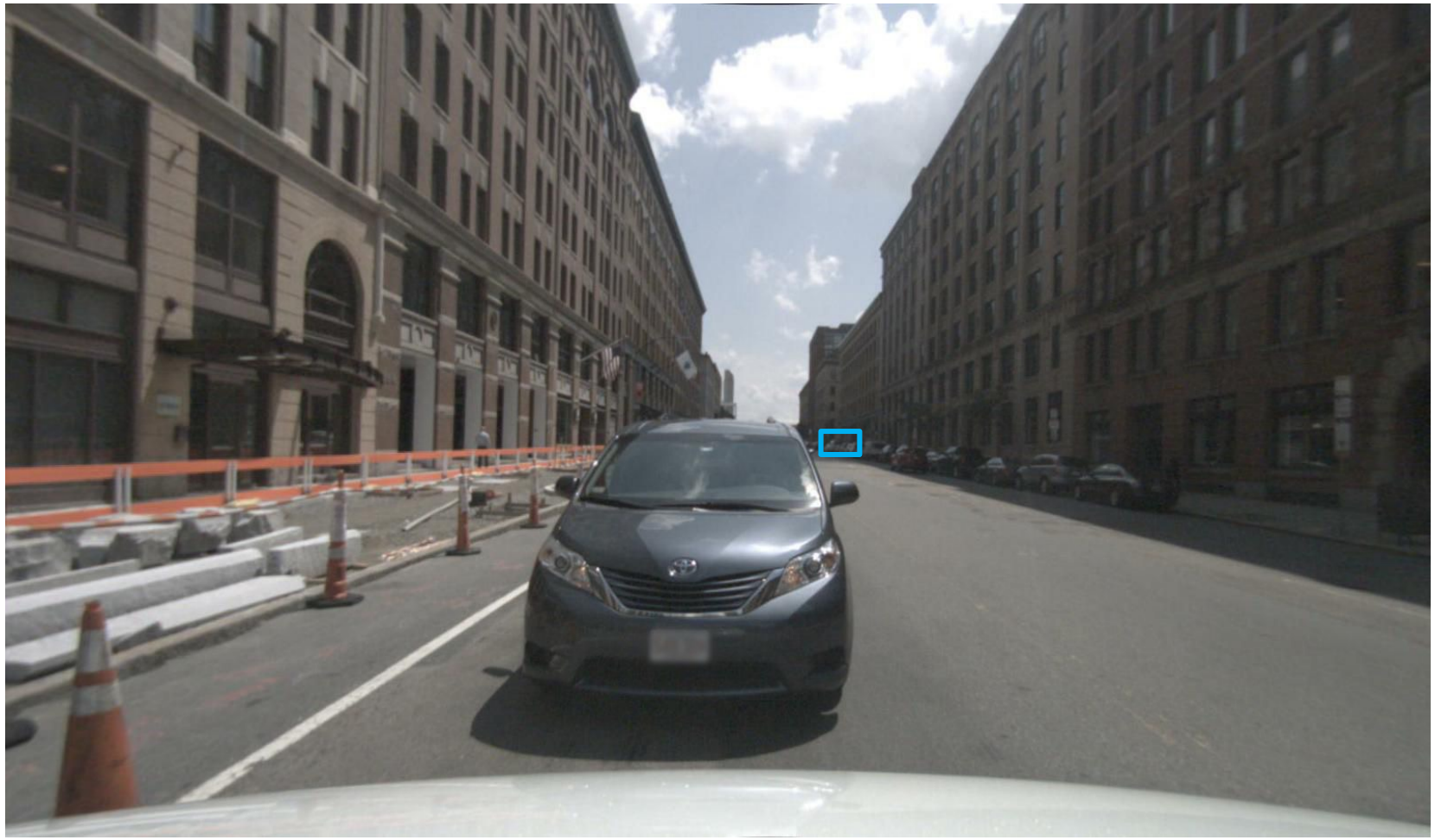
Prediction Using Wider Context



What is this?

Prediction Using Wider Context

Now one can tell – given the context!



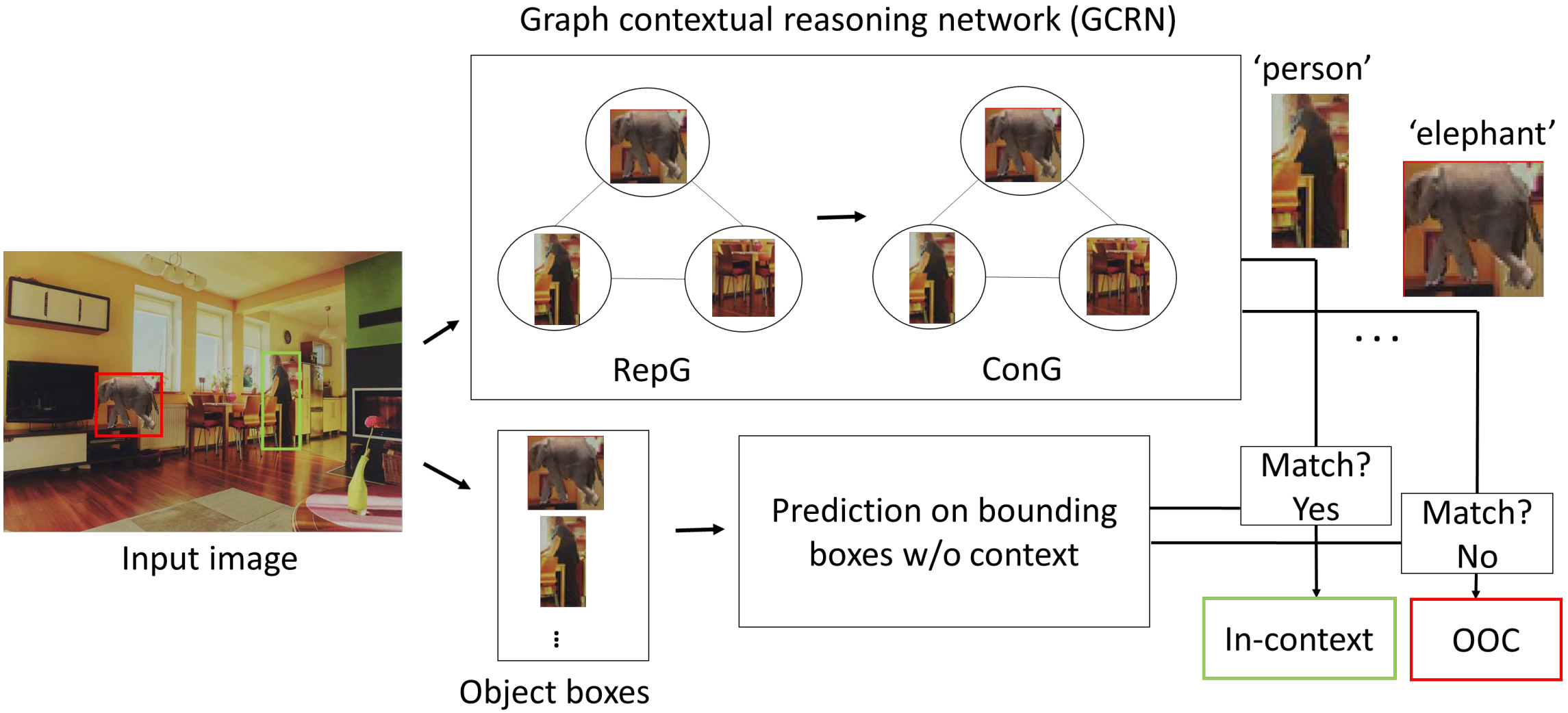
Prediction Using Wider Context

What is this?

Prediction Using Wider Context

Now one can tell given the context.

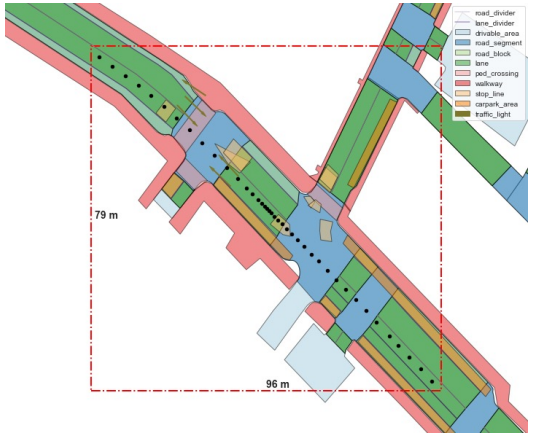
Graph Contextual Reasoning Network

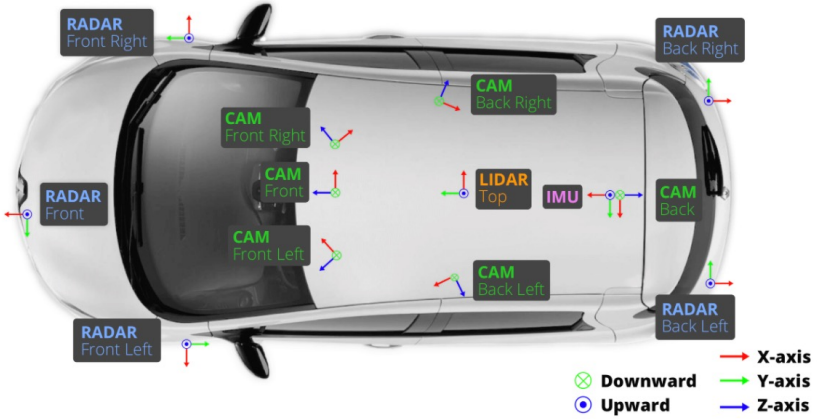


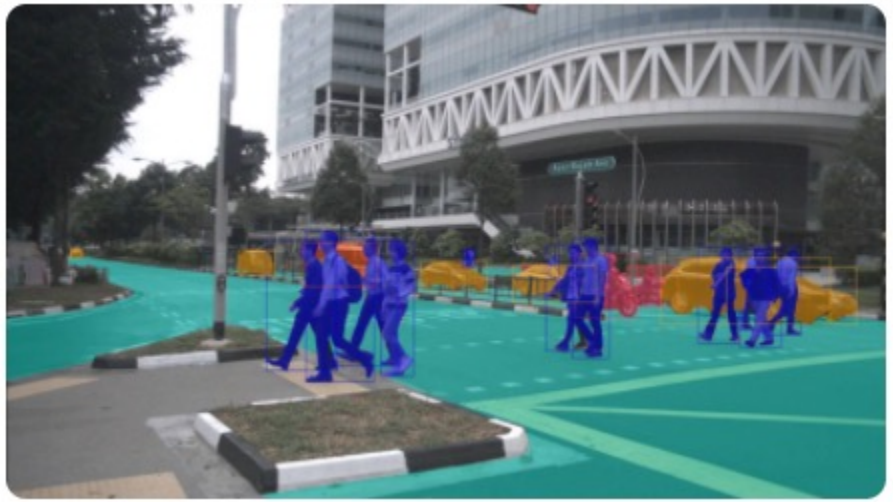
Detecting out-of-context objects using graph contextual reasoning network. Acharya at. Al. IJCAI, 2022.

NuScenes

- Very rich dataset, useful for different tasks
- 1x LIDAR, 5x RADAR, 6x camera, IMU, GPS
- 1000 scenes of 20s each
- Two diverse cities: Boston and Singapore
- Detailed map information (segmentation)
- 1.4M 3D bounding boxes manually annotated for 23 object classes
- Attributes such as visibility, activity and pose
- Object bounding boxes (car, person, bike, traffic cone, etc.)
- 2D and 3D annotated boxes with occlusion details
- Semantic segmentation (road, sidewalk, etc)
- Map data of the city
- Presence of temporal sequences
- Presence of LIDAR and IMU data







NuScenes

Goal: Object classification using contextual cues

Object classes: We consider six object classes

- Object classes and frequency of samples:

human (19.46%), **bicycle (1.04%), motorcycle (1.11%)**, car (43.62%), truck (12.70%), movable_object (22.05%)

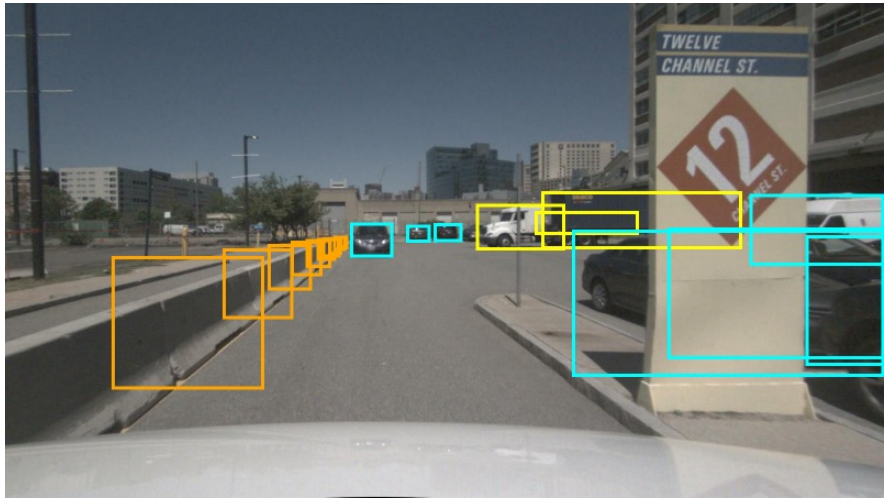
NuScenes

Results on occluded bounding boxes to test the robustness of GCN

Model	Occlusion (%)	Overall accuracy	Class-wise accuracy					
			human	bicycle	motor-cycle	car	truck	movable object
CNN - ResNet (Baseline)	No occlusion	88.65	92.44	57.24	61.31	92.59	69.74	90.69
CNN - ResNet (Baseline)	30%	83.24	90.99	12.52	20.90	92.48	71.15	71.36
CNN - ResNet (Baseline)	50%	79.17	94.93	2.36	12.48	87.33	58.94	67.95
Trinity	No occlusion	95.51	98.38	66.25	73.37	97.13	82.17	98.62
Trinity	30%	94.70	98.72	66.66	65.40	96.62	81.31	96.73
Trinity	50%	93.13	97.53	31.36	64.88	94.17	82.10	96.34

Less frequent (~1%) classes

NuScenes: Qualitative



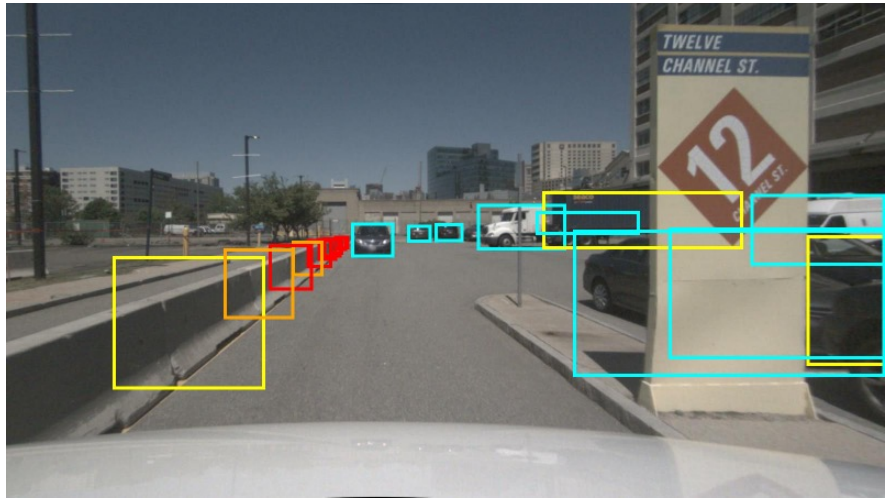
Ground truth

person

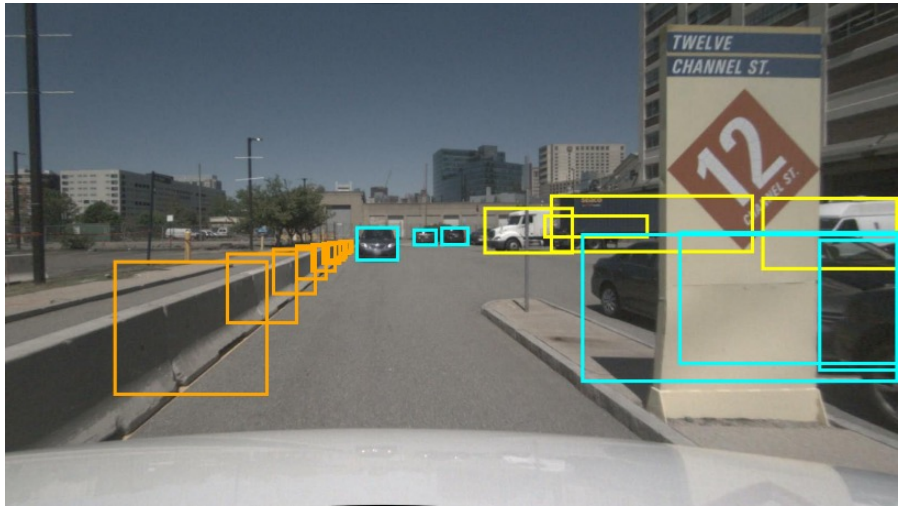
movable object

car

truck



CNN

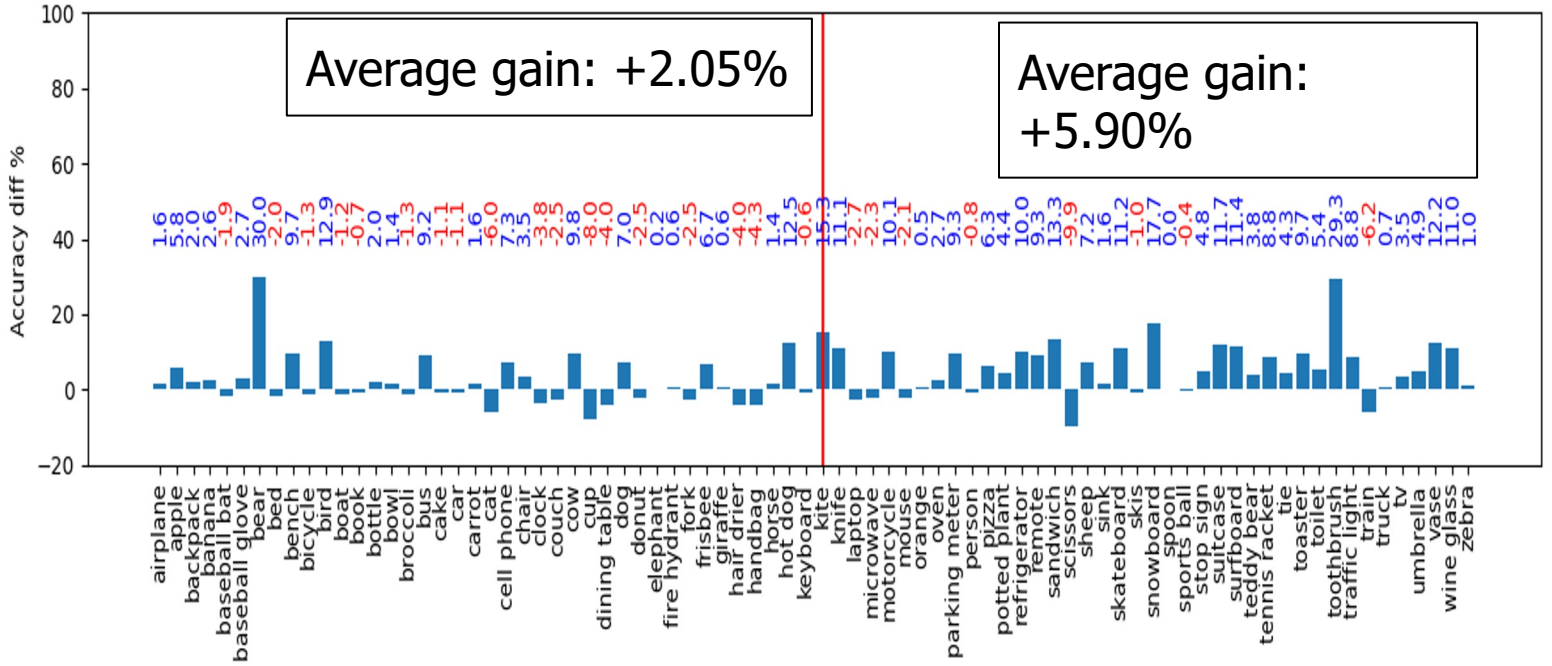


Trinity

Prediction Using Wider Context: Novel Classes

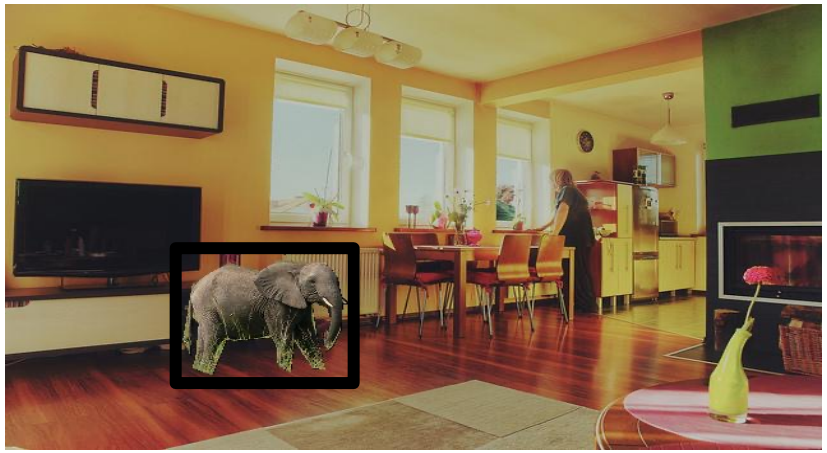
- **Coco Dataset**: 80 classes, 80K in training set and 40K in test set.
- Train FastRCNN on the alphabetically first 40 classes as the feature extractor.
- Train/test the downstream MLP and GraphCNN on all the 80 classes.

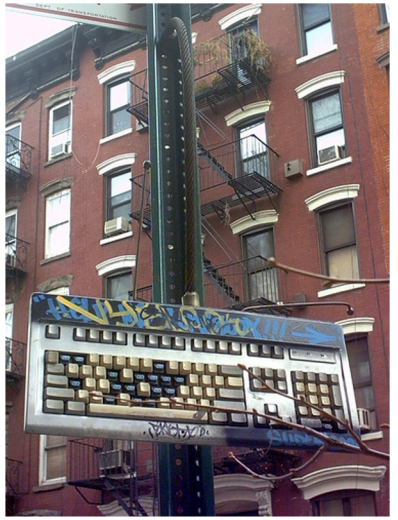
Modeling context using GraphCNN improves prediction particularly over the novel classes.



(GCN - MLP) accuracy difference.
Blue = GCN is better
Red = MLP is better
Classes right to the middle vertical line are the 40 novel classes.

Out-of-context Inputs





In-context object

Out-of-context object

OOC Results on COCO-OOC and OCD

Approach	AUC score
Softmax confidence	0.043
GCRN (w/o ConG)	0.589
GCRN	0.980

Approach	AUC score
GCRN (oracle boxes + labels)	0.980
GCRN (oracle boxes, pred labels)	0.897
GCRN (pred boxes)	0.771

Approach	AUC score
Softmax confidence	0.402
GCRN (oracle boxes, pred labels)	0.587
GCRN (oracle boxes + labels)	0.709



COCO OOC dataset

OCD dataset



Evaluation on Open Datasets

Novel object detection:

Datasets: Tiny imagenet with 200 object classes. 20 classes are available during training and rest 180 classes considered as the novel objects

Metrics: Area under ROC (AUROC) for novel object recognition and detection accuracy (DTACC) for the closed set recognition

Novel object recognition:

TinyImageNet	OpenMax (CVPR16)	G-OpenMax (BMVC17)	OSRCI (ECCV18)	C2AE (CVPR19)	CROSR (CVPR19)	Gen-dis (CVPR20)	Ours
AUROC	57.6	58.0	58.6	58.1	58.9	64.7	**73.26**

Closed set recognition:

TinyImageNet	Gen-dis (CVPR20) Resnet-18	Gen-dis (CVPR20) WideResnet-28-10	Ours
DTACC	49.2	55.9	**74.74**

OOD detection:

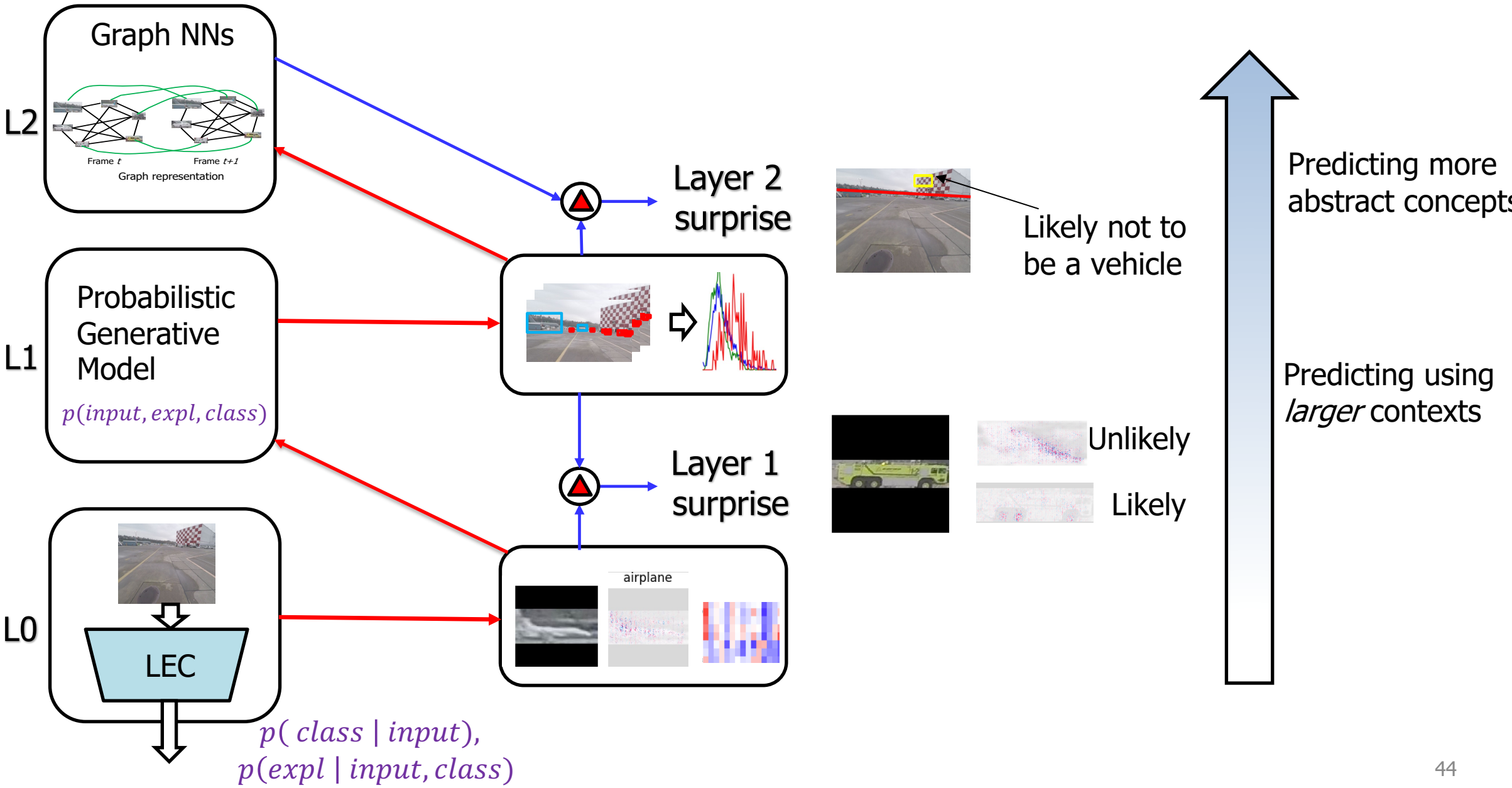
Datasets: MNIST, KMNIST, F-MNIST, CIFAR-10, CIFAR100, STL10, SVHN, LSUN, ImageNet

Metrics: True negative rate (TNR) @ true positive rate (TPR) = 95%, Area under ROC (AUROC), detection accuracy (DTACC)

In-dist (model)	OOD dataset	TNR (TPR=95%)	AUROC	DTACC
MNIST (LeNet5)	KMNIST	67.72 / 80.52 / **91.82**	92.98 / 96.53 / **98.3**	85.99 / 90.82 / **94.01**
	F-MNIST	58.47 / 63.33 / **74.49**	90.76 / 94.11 / **95.55**	83.21 / 87.76 / **90.98**
CIFAR10 (ResNet34)	STL10	10.63 / 13.9 / **17.4**	61.56 / 66.47 / **67.52**	59.22 / 62.75 / **63.7**
	SVHN	72.85 / 53.16 / **88.2**	93.85 / 93.85/ **97.69**	85.4 / 89.173 / **92.14**
	Imagenet	46.54 / 68.41 / **74.53**	90.45 / 95.02 / **95.73**	83.06 / 88.63 / **89.73**
	LSUN	45.16 / 77.53 / **81.23**	89.63 / 96.51 / **96.87**	81.83 / 90.64 / **91.19**
	SCIFAR100	37 / 38.39 / **61.11**	86.13 / 88.86 / **94.74**	78.5 / 82.51 / **90.53**
CIFAR10 (ResNet50)	STL10	12.19 / 10.33 / **16**	60.29 / 61.95 / **66.39**	58.57 / 59.36 / **62.28**
	SVHN	86.61 / 34.49/ **91.06**	84.41 / **98.19** / 91.98	91.25 / 76.72 / **93.2**
	Imagenet	73.23 / 29.48 / **75.96**	94.91 / 84.3 / **95.79**	88.23 / 77.19 / **89.26**
	LSUN	80.72 / 32.18 / **81.38**	96.51 / 87.09 / **96.93**	90.59 / 80.07 / **91.79**
	SCIFAR100	47.44 / 21.06 / **48.33**	86.16 / 77.42/ **92.98**	78.69 / 71.43 / **88.27**
SVHN (DenseNet)	STL10	45.91 / 81.66 / **87.76**	77.6 / 96.97 / **97.63**	72.62 / 92.29 / **93.35**
	CIFAR10	37.23 / 80.82 / **86.42**	73.14 / 96.8 / **97.37**	68.92 / 92.27 / **92.86**
	Imagenet	62.76 / 85.44 / **93.44**	85.41 / 97.29 / **98.38**	79.94 / 93.39 / **94.53**
	LSUN	62.91 / 76.87 / **89.73**	86.06 / 96.37 / **97.73**	80.04 / 92.43 / **93.55**
	SCIFAR100	48.17 / 86.06 / **96.72**	78.94 / 97.43 / **98.24**	73.72 / 93.02 / **96.26**

Comparison: ODIN [Liang et al., 2017] , Mahalanobis [Lee et al., 2018]

Predictive Coding Inspired Top-Down Models



Feature Attribution for Deep Neural Networks

From cooperative game theory, we have classic equations to compute Shapley values

$$a_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!\,(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}\left(x_{S \cup \{i\}}\right) - f_S(x_S) \right]$$

Young (1985) demonstrated that Shapley values are the only set of values that satisfy the three properties: local accuracy, sensitivity, and consistency.

Apply sampling approximations to above equation and approximate the effect of removing a variable from the model by integrating over samples.

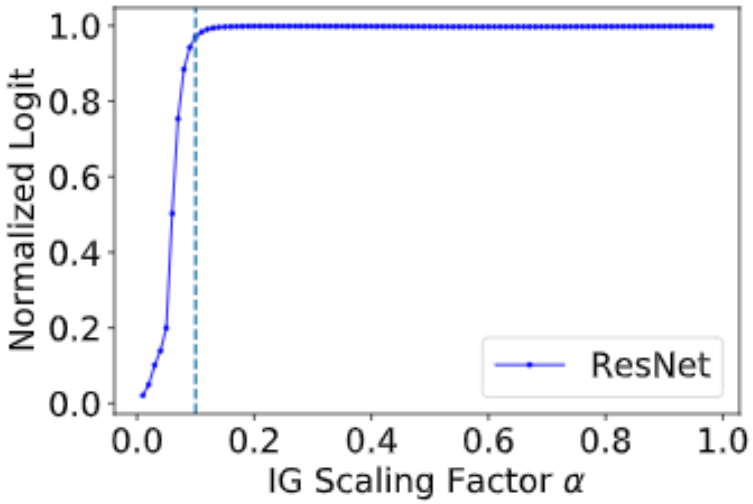
Friedman, Eric J. (2004) Paths and consistency in additive cost sharing. Journal of Game Theory, 501–518
Deep learning applications: Integrated Gradients (IG. Sundararajan et. al.'17), DeepShap

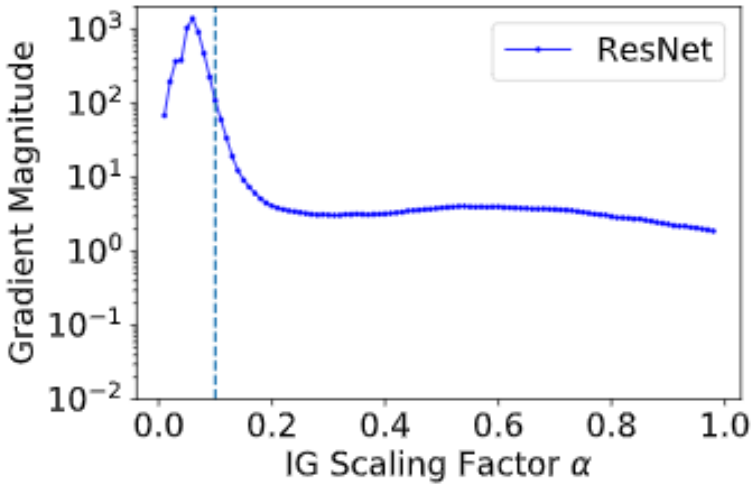
Given $\gamma = (\gamma_1, \ldots, \gamma_n)$: $[0,1] \rightarrow R^n$ be a smooth function specifying a path in R^n from baseline x^b to input x, that is, $\gamma(0) = x^b, \gamma(1) = x$.

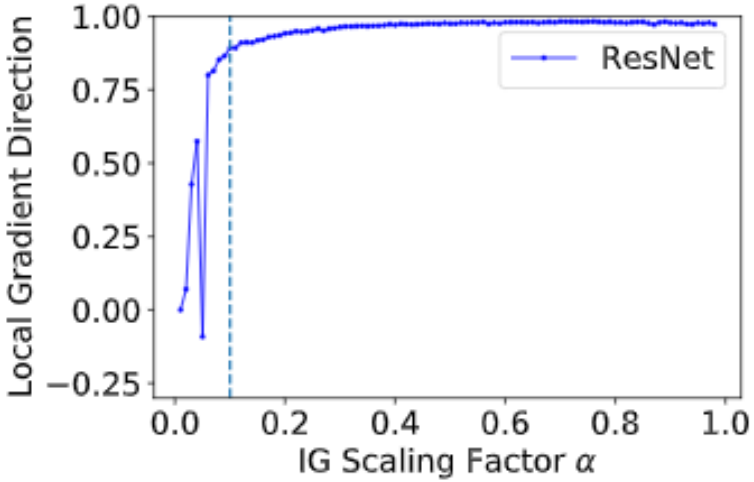
$$\int_{\alpha=0}^{1} \frac{\partial F(\gamma(\alpha))}{\partial \gamma_i(\alpha)} \frac{\partial \gamma_i(\alpha)}{\partial \alpha} d\alpha \qquad \mathcal{A}_j^i(\mathbf{x}) = (\mathbf{x}_j - \mathbf{x}_j^b) \times \int_{\alpha=0}^{1} \partial_j \mathcal{F}^i(\mathbf{x}^b + \alpha(\mathbf{x} - \mathbf{x}^b)) d\alpha$$

Integrated Gradient : Challenges

Gradient Magnitude and Correlation after Saturation

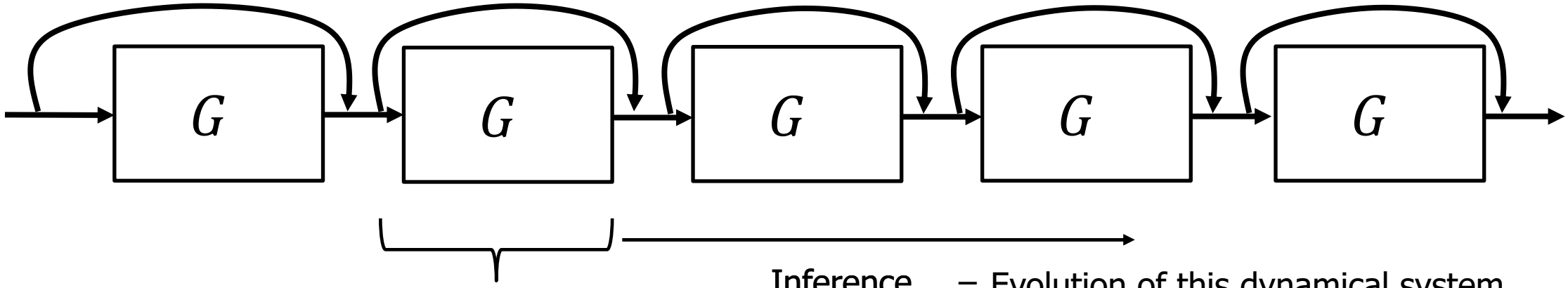






Resnets

Trustworthy AI: Neural Stochastic Differential Equation Models for Robustness



$$x_{l+1} = G(x_l, w_l) + x_l$$

$x_{l+1} = G(x_l, w_l) + x_l \qquad y_l = f(x_L)$ We introduce a temporal partition: $t_l = \frac{l}{L}$ where $l = 0,1,2, \ldots$ with $\Delta t = \frac{1}{L}$

$$x(t_{l+1}) = \overline{G}(x(t_l), w(t_l))\,\Delta t\; + x_l \qquad y_l = f(x(1))$$

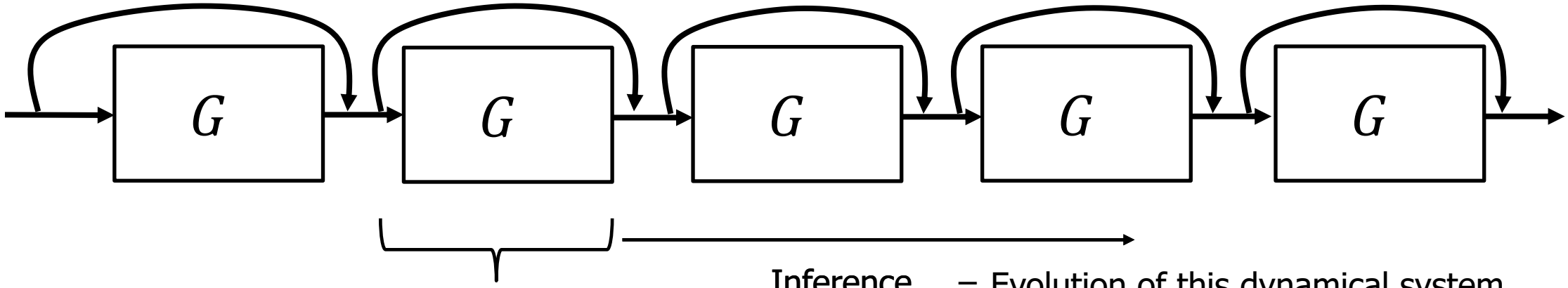
The above time-difference equation is the Euler discretization of the following ODE.

$$\frac{dx(t)}{dt} = \overline{G}(x(t_l), w(t_l))$$

Let $u(x,t)$ be a quantity that is constant along the flow, then it satisfies the following transport equation.

$$\frac{d}{dt}(u(x(t),t) = \frac{\partial u(x,t)}{\partial t} + \overline{G}\big(x(t_l), w(t_l)\big)\,\nabla u(x,t) = 0 \qquad u(x,1) = f(x)$$

Trustworthy AI: Neural Stochastic Differential Equation Models for Robustness



$$x_{l+1} = G(x_l, w_l) + x_l$$

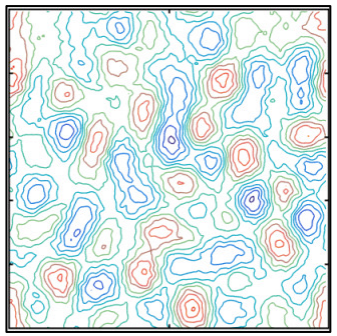
Backpropagation in resnet can be modeled as finding the velocity field $\overline{G}(x(t_l), w(t_l))$ for the following transport eqn.

$$\frac{\partial u(x,t)}{\partial t} + \overline{G}\big(x(t_l), w(t_l)\big)\,\nabla u(x,t) + \frac{1}{2}\sigma^2 \Delta u(x,t) = 0$$

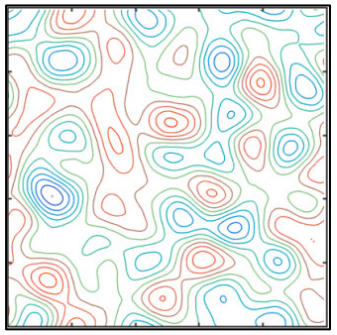
$$u(x,1) = f(x) \qquad u(x_i, 0) = y_i \text{ for all } (x_i, y_i) \text{ in the dataset}$$

$u(x,0)$ serves as the classifier and the velocity field $\overline{G}(x, w(t))$ encodes ResNet's architecture and weights.

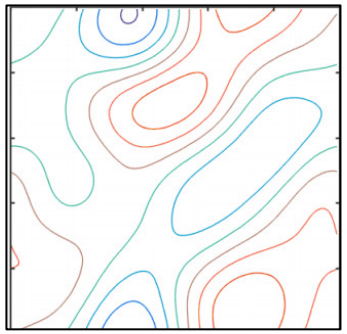
Trustworthy AI: Neural Stochastic Differential Equation Models for Robustness



$\sigma = 0$



$\sigma = 0.01$



$\sigma = 0.1$

Backpropagation in resnet can be modeled as finding the velocity field $\overline{G}(x(t_l), w(t_l))$ for the following transport eqn.

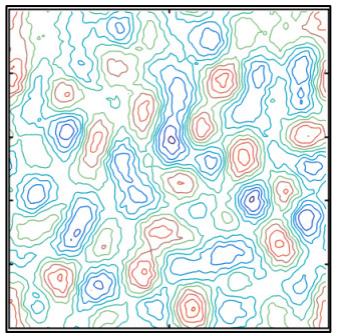
$$\frac{\partial u(x,t)}{\partial t} + \overline{G}\big(x(t_l), w(t_l)\big)\,\nabla u(x,t) + \frac{1}{2}\sigma^2 \Delta u(x,t) = 0$$

$$u(x,1) = f(x) \qquad u(x_i, 0) = y_i \text{ for all } (x_i, y_i) \text{ in the dataset}$$

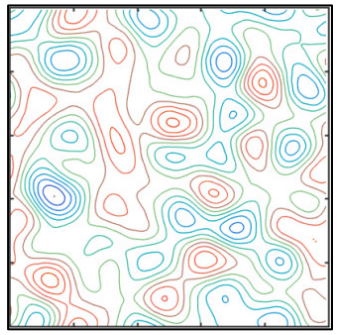
$u(x,0)$ serves as the classifier and the velocity field $\overline{G}(x, w(t))$ encodes ResNet's architecture and weights.

When $\overline{G}$ is very complex, $u(x,0)$ might be highly irregular i.e. a small change in the input x can lead to a massive change in the value of $u(x,0)$

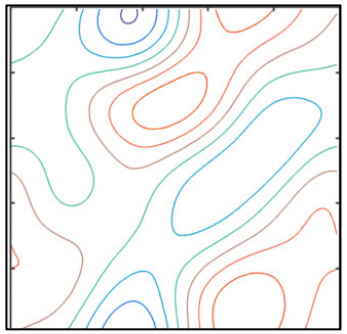
Trustworthy AI: Neural Stochastic Differential Equation Models for Robustness



$\sigma = 0$



$\sigma = 0.01$



$\sigma = 0.1$

Feature Robustness Theorem: If $G(x(t), W(t))$ is Lipschitz function in both x and t, the target classifier being learned is a compactly supported bounded function and $0 < \sigma \leq 1$, then the solution $u(x,t)$ for the equation above satisfies

$$|u(x+\delta, 0) - u(x,0)| \leq \alpha \left(\frac{|\delta|}{\sigma}\right)^{\beta}$$

for any small perturbation δ where $\beta > 0$ and α depends on the infinity norm of $G(x(t), W(t))$

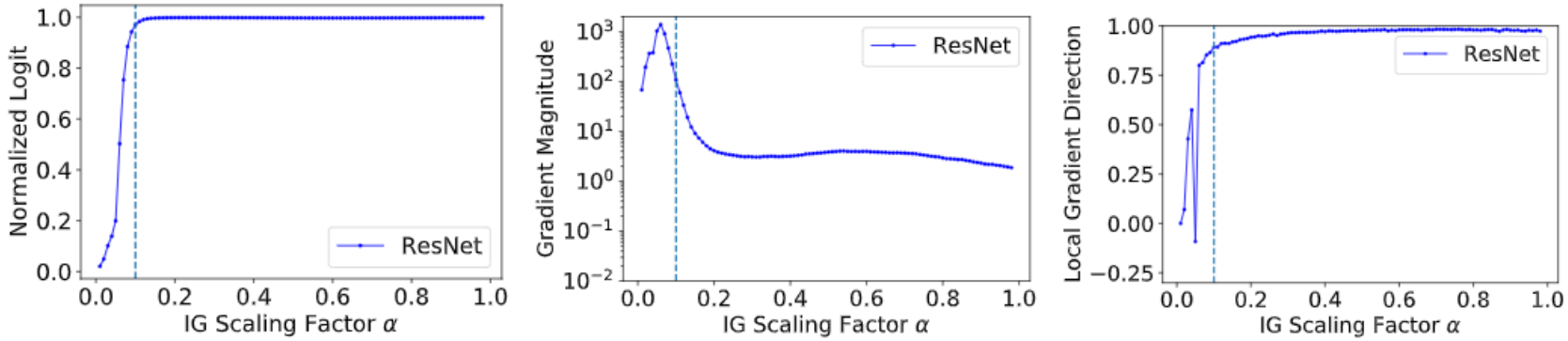
Explanation Robustness Theorem: If $G\big(x(t), W(t)\big)$ is a continuously differential function in both x and t, the target classifier being learned is a compactly supported bounded function and $0 < \sigma \leq 1$, then the solution $u(x,t)$ for the equation above satisfies

$$|\nabla u(x,1)| \leq \alpha e^{-\sigma^2+\beta}$$

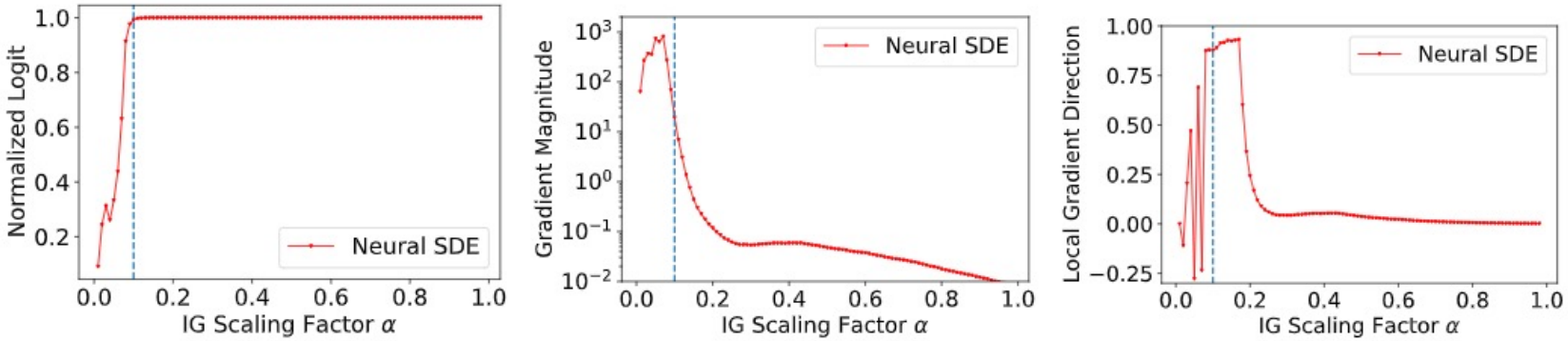
For any small perturbation β depends on ∇G and α depends on the infinity norm of the classifier and its gradient.

Improved attribution over input features for ML decisions using Neural SDEs

Gradient Magnitude and Correlation after Saturation



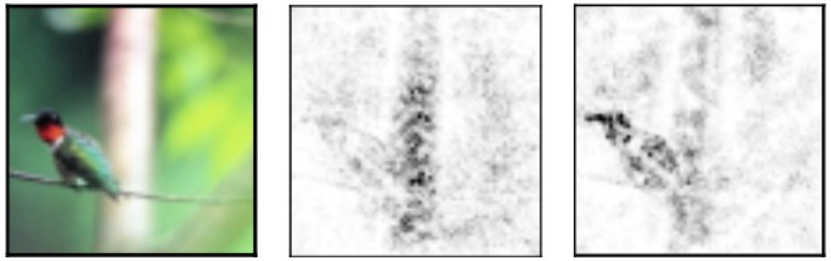
Resnets



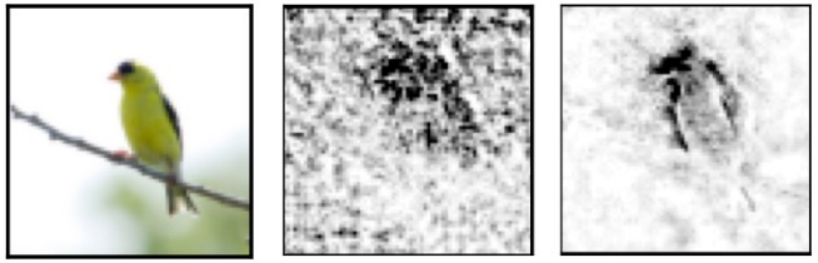
Neural SDEs

On Smoother Attributions using Neural Stochastic Differential Equations. Jha et al. IJCAI'21
Shaping Noise for Robust Attributions in Neural Stochastic Differential Equations. Jha et al. AAAI'22 (Oral)

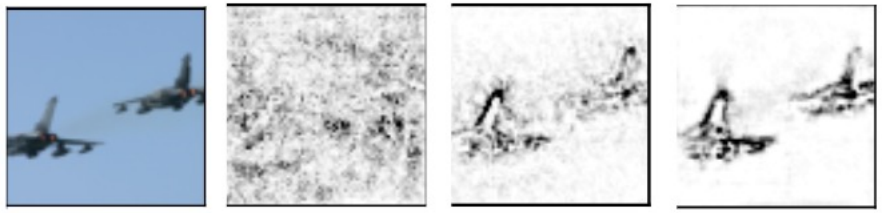
Improved attribution over input features for ML decisions using Neural SDEs



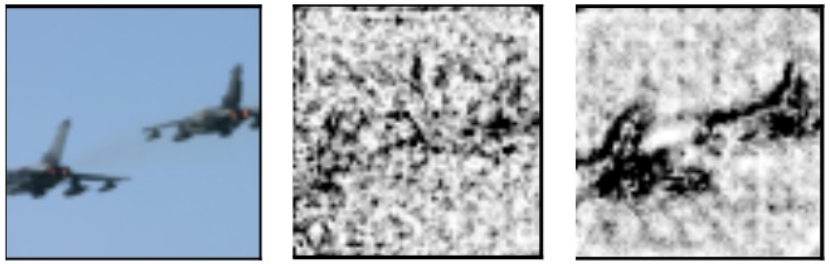
DeepLIFT



Integrated Gradient



Integrated Gradient + Noise Tunnel



DeepShap

On Smoother Attributions using Neural Stochastic Differential Equations. Jha et al. IJCAI'21
Shaping Noise for Robust Attributions in Neural Stochastic Differential Equations. Jha et al. AAAI'22 (Oral)

Improved attribution over input features for ML decisions using Neural SDEs

Model	Attribution	Reference	Sensitivity Metric		
			Standard	Noise	Attribution-driven Noise
ResNet-50	IG	[6]	0.576	0.450	0.420
	IG + NT	[11]	1.036	–	0.866
	Saliency Map	[1]	0.596	0.551	0.478
	DeepLIFT	[8]	0.729	0.613	0.554
	DeepSHAP	[9]	0.363	0.323	0.318
WideResNet-101	IG	[6]	0.561	0.494	0.461
	IG + NT	[11]	1.433	–	1.408
	Saliency Map	[1]	0.577	0.548	0.501
	DeepLIFT	[8]	0.777	0.667	0.643
	DeepSHAP	[9]	0.344	0.323	0.316
ResNeXt-101	IG	[6]	0.590	0.498	0.401
	IG + NT	[11]	1.443	–	1.440
	Saliency Map	[1]	0.616	0.557	0.462
	DeepLIFT	[8]	0.775	0.713	0.546
	DeepSHAP	[9]	0.379	0.330	0.321

Model	Method	Reference	SIC
ResNet-50	Gradients	[1]	0.510
ResNet-50	IG	[6]	0.544
ResNet-50	IG + Noise Tunnel	[11]	0.590
Attribution-Driven Noise	IG	Our Approach	**0.683**

Softmax Information Curve (SIC): Contents are re-introduced in a blurred (bokeh) version of the image to avoid sharp boundary effects and the output is monitored. We use the proportion of the original input's label output or softmax score as the performance

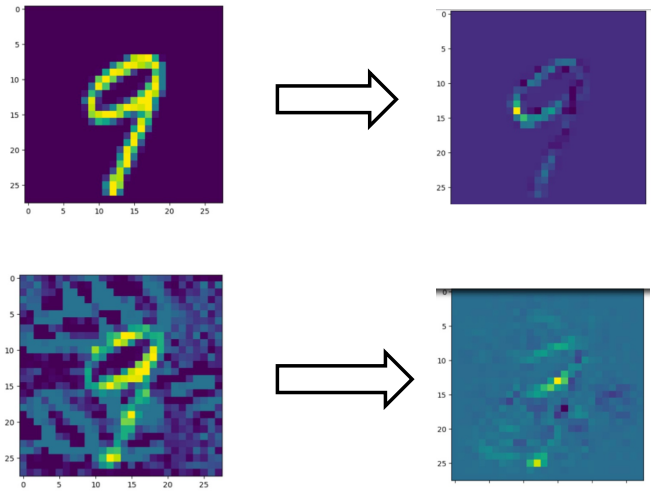
Attribution Robustness

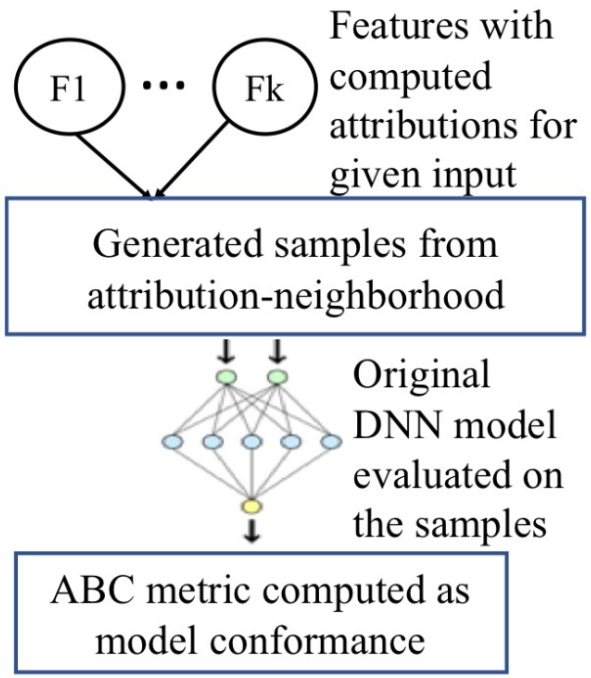
Perturbations to inputs that do not change the model output substantially should not change the attribution significantly. The computed attributions should be robust to such small perturbations of the input.

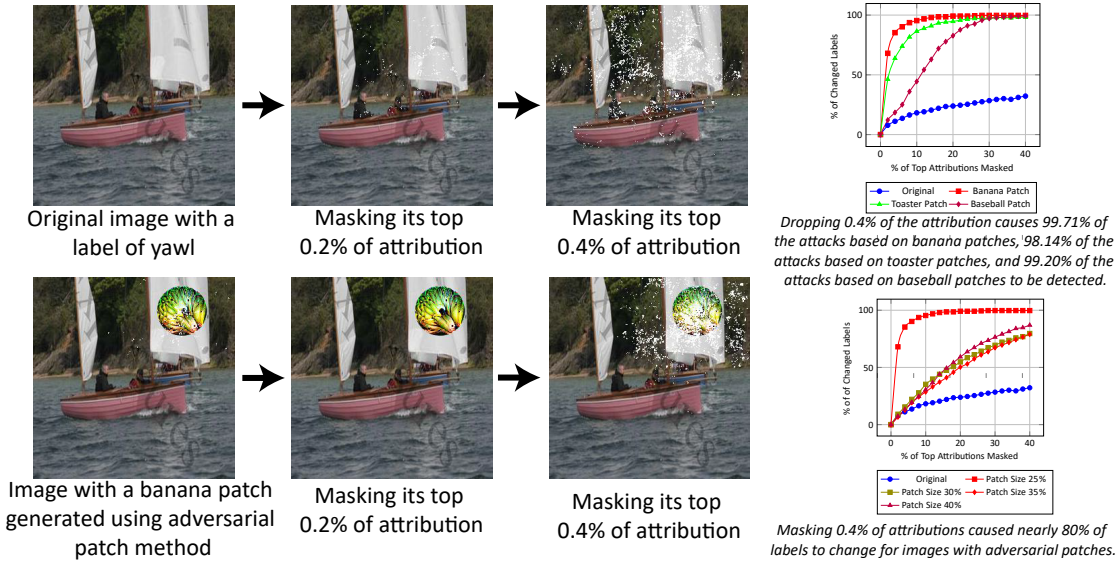
Attribution scores should be faithful to the model – removing the top or bottom features should lead to decrease or increase in the model's output (logit) for the class of the original input

$$\mathbb{S}_r(\mathcal{A}, \mathcal{D}) = \sum_{\mathbf{x} \in \mathcal{D}} \frac{\max_{\|\delta\|_\infty \leq r} \|\mathcal{A}(\mathbf{x}+\delta) - \mathcal{A}(\mathbf{x})\|_2}{\|\mathcal{A}(\mathbf{x})\|_2} \text{ such that } \forall \|\delta\|_\infty \leq r,\ F(x+\delta) = F(x)$$

Feature Attribution for Deep Neural Networks



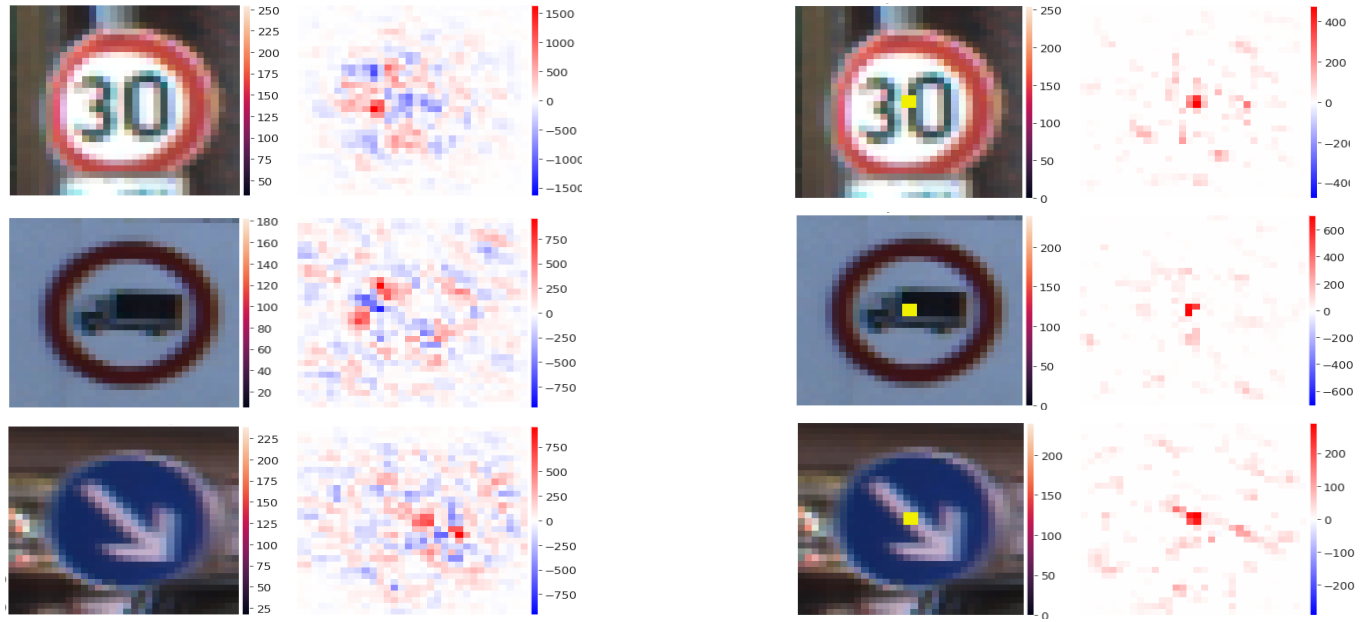




Attribution-Based Confidence (ABC) Metric For Deep Neural Networks. Jha et. al. Thirty-third Conference on Neural Information Processing Systems (NeurIPS) 2019

Improving Resilience Using Attributions/Explanations

Adversarial perturbations cause disproportionally high concentration of attributions.

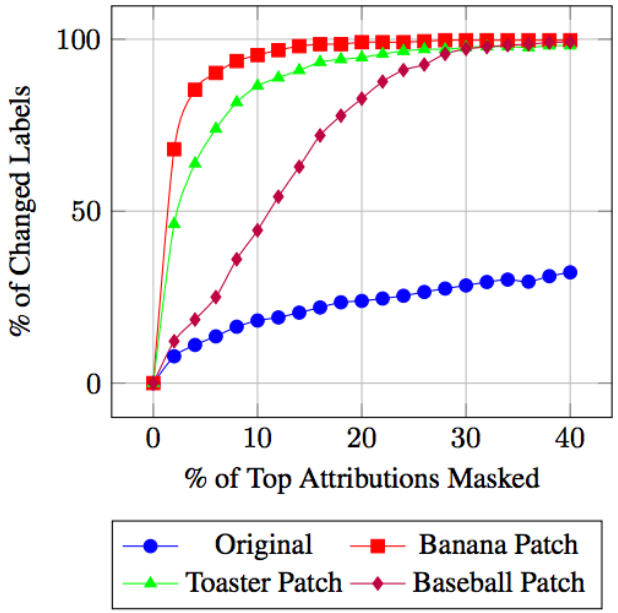


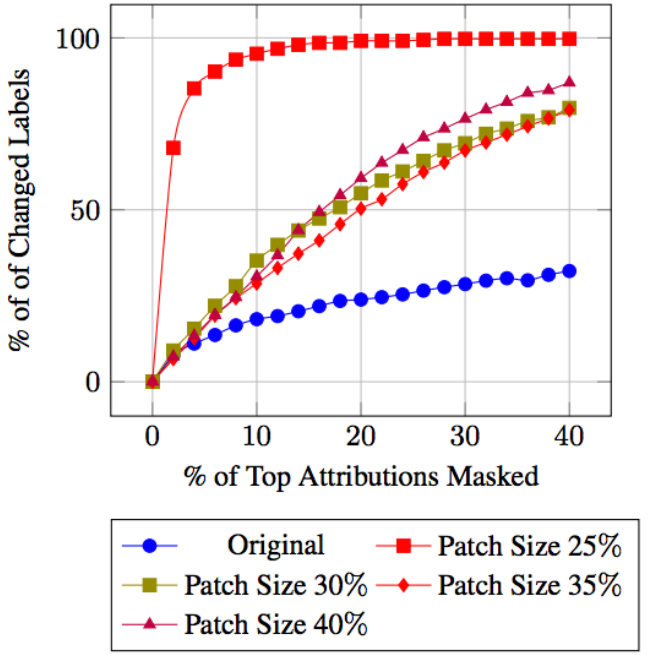
Improving Resilience Using Attributions/Explanations

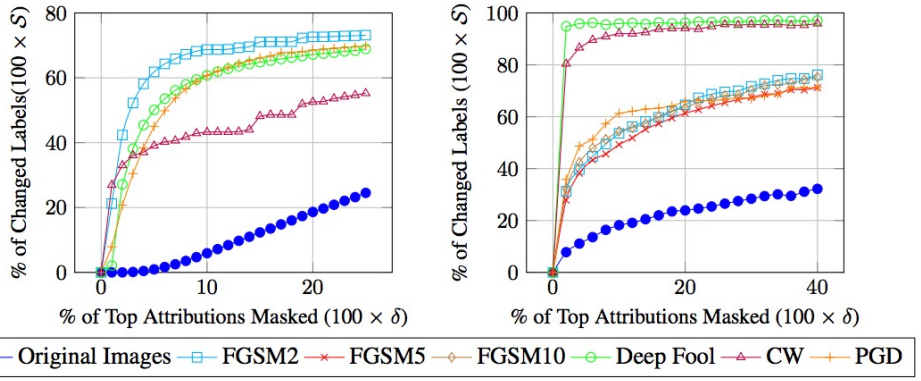
The decision of machine learning model changes when a small percentage of high attribution features of an adversarial input is masked.

Improving Resilience Using Attributions/Explanations

The decision of machine learning model changes when a small percentage of high attribution features of an adversarial input is masked.



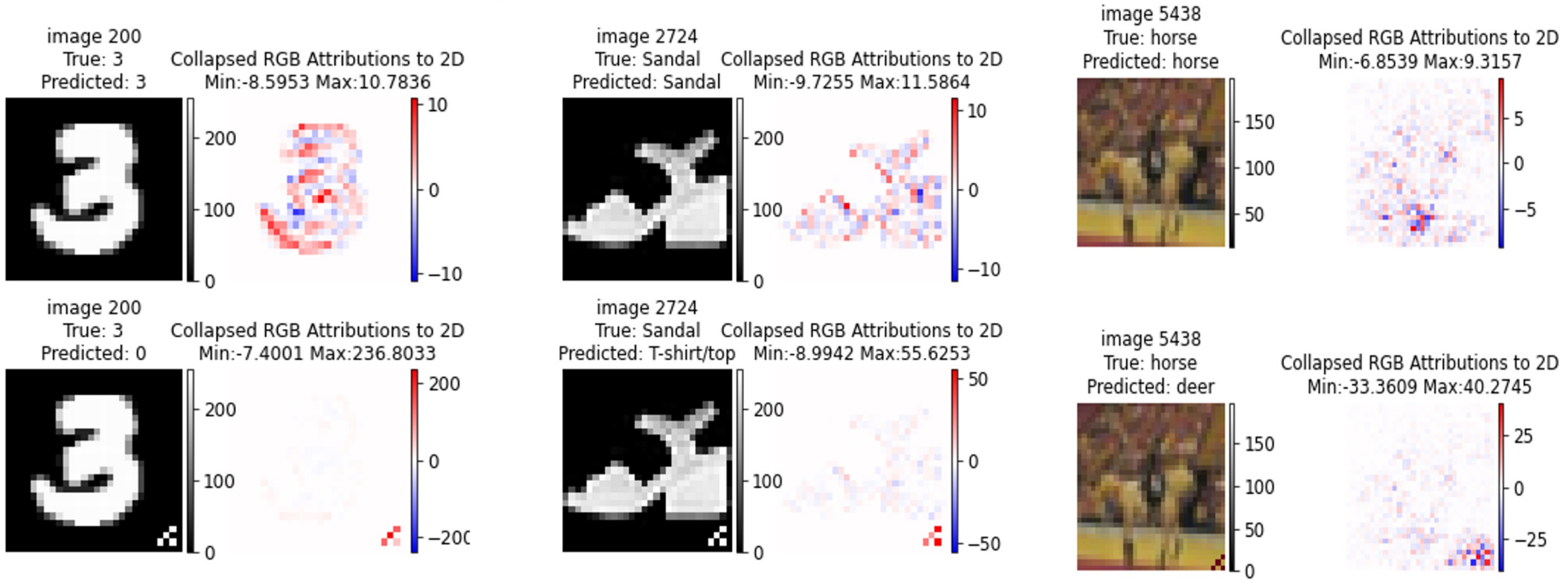




Attribution-Based Confidence (ABC) Metric For Deep Neural Networks. Jha et. al. (NeurIPS) 2019

Detecting Backdoors in ML Models using Attributions

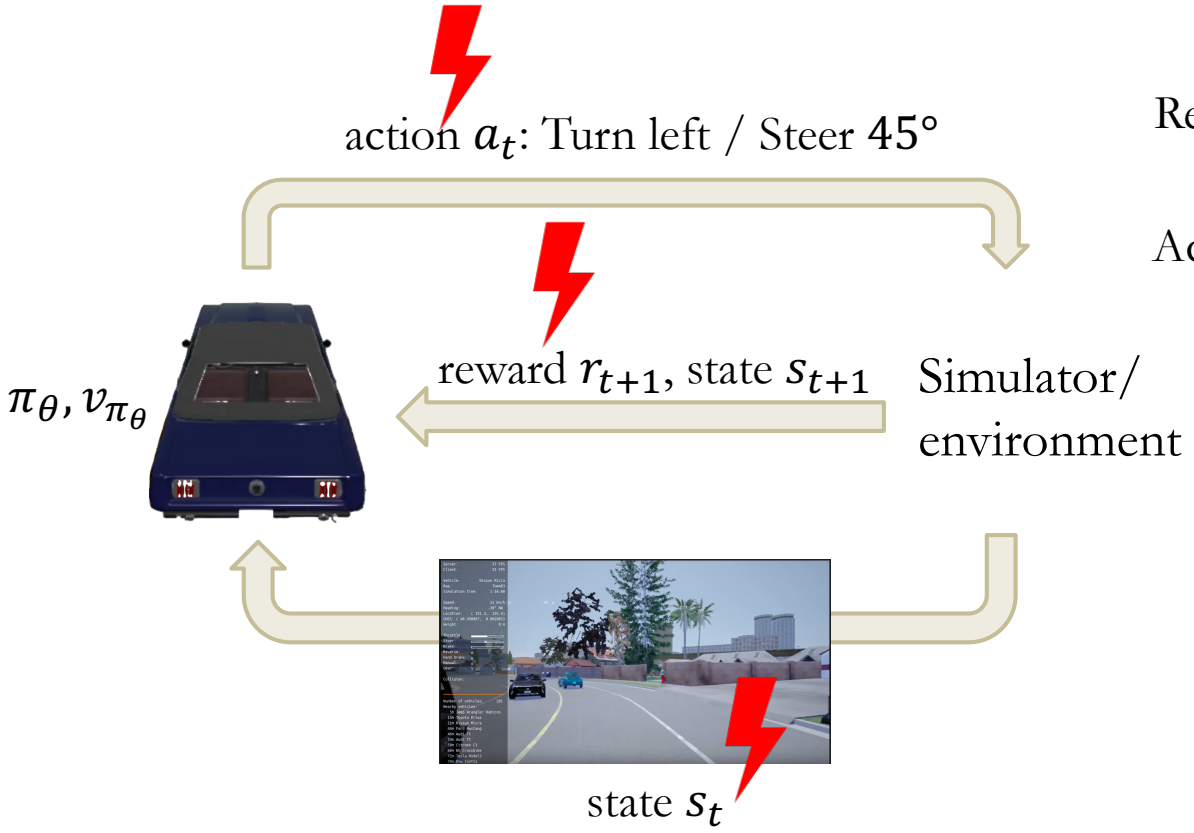
Trojan trigger causes disproportionally high concentration of attributions.



MISA: Online Defense of Trojaned Models using Misattributions. Kiourti et. al. ACSAC'21

Trojan/Backdoor Attacks on Reinforcement Learning

We had initially developed a Trojan attack on RL policies.



Return: $R_t = \sum_{k=t+1}^{T} \gamma^{k-t+1} r_k$

Action-value function:
$Q_\pi(s_t, a_t) = E_\pi[R_t | s_t, a_t]$

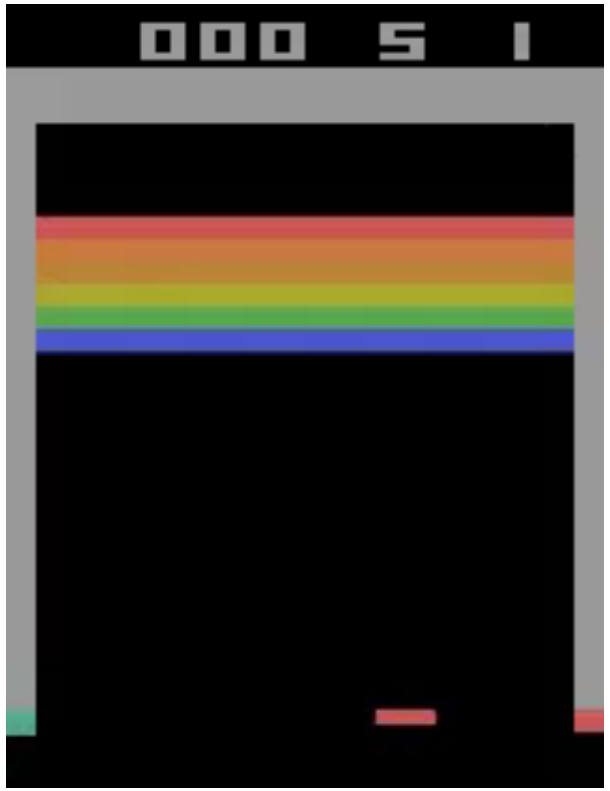
Value function:
$v_\pi(s_t) = E_\pi[Q_t \mid s_t]$

Advantage:
$A(s_t, a_t) = Q_\pi(s_t, a_t) - V_\pi(s_t)$

TrojDRL: Evaluation of Backdoor Attacks on Deep Reinforcement Learning. Kiourti et al. DAC'20

Trojan/Backdoor Attacks on Reinforcement Learning

We had initially developed a Trojan attack on RL policies.

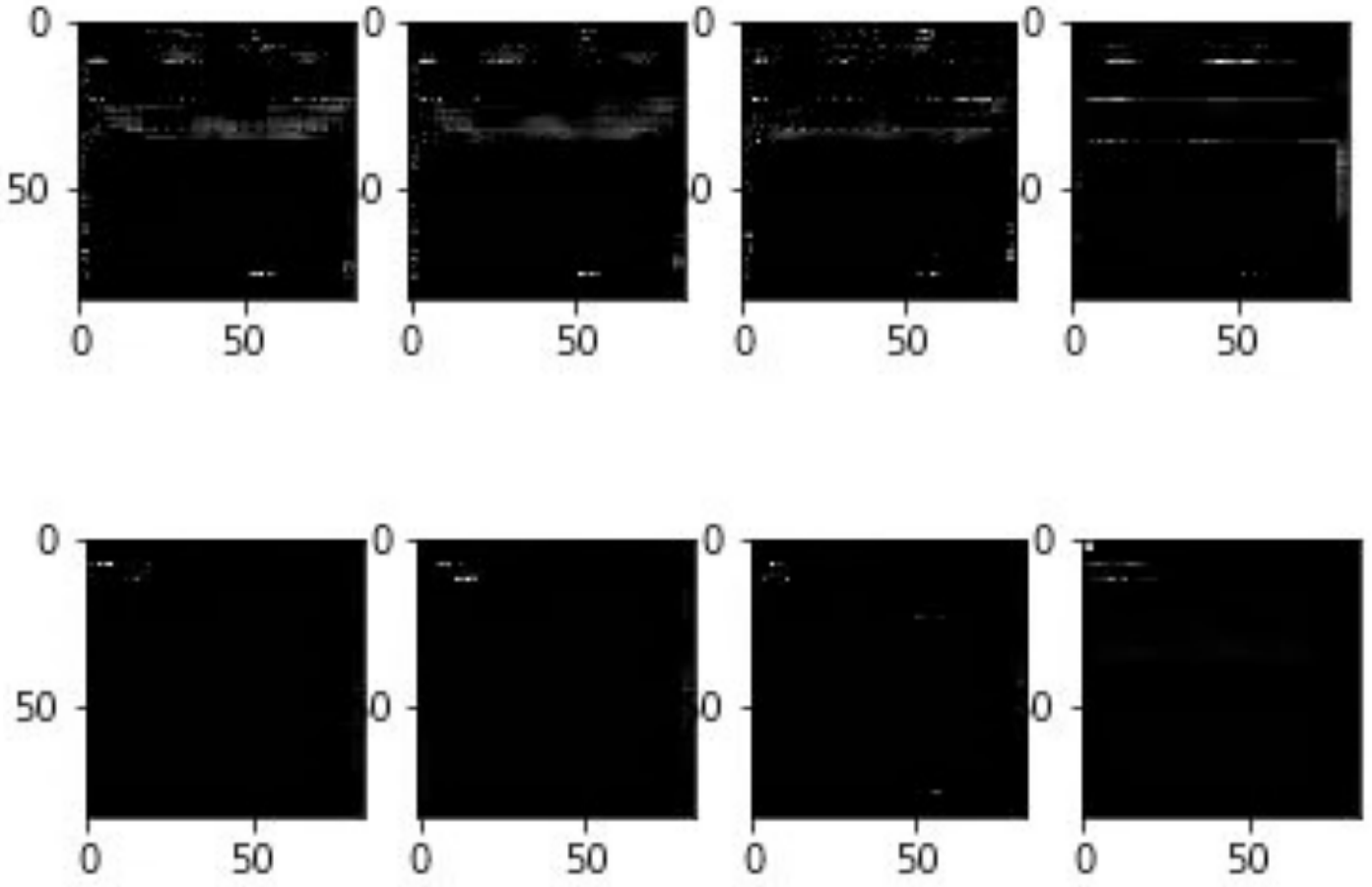


Game	Score during the attack					
	Targeted		Untargeted		Standard	
	Mean	Std	Mean	Std	Mean	Std
Breakout	1	1	2	2	250	147
Qbert	658	1176	965	1220	7890	2770
Seaquest	7	10	32	18	220	111
Space Invaders	13	12	50	47	161	230
Crazy Climber	0	0	0	0	13870	11562

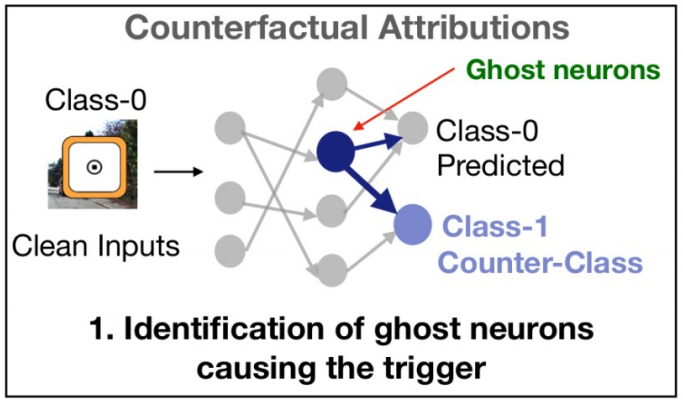
TrojDRL: Evaluation of Backdoor Attacks on Deep Reinforcement Learning. Kiourti et al. DAC'20

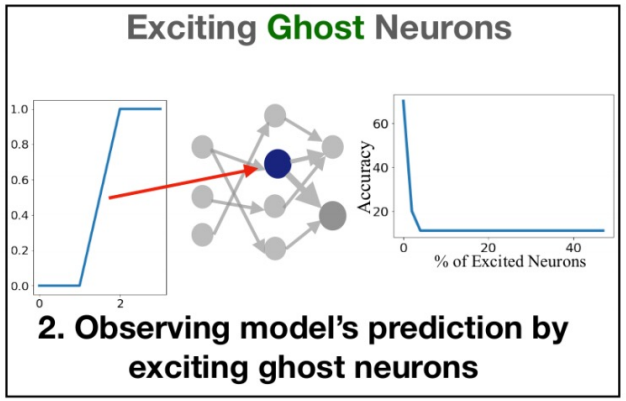
Detecting Trojans in RL Policy Using Attributions/Explanations

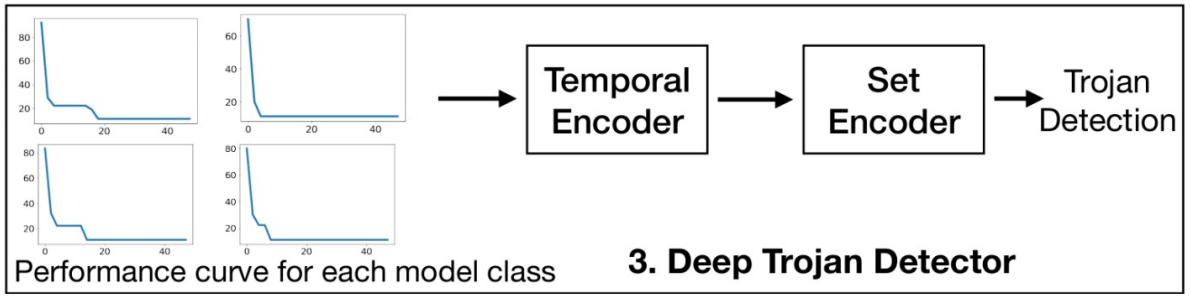
Attributions can detect Trojan triggers in backdoored observations.

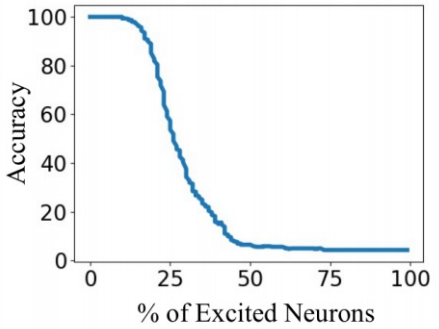


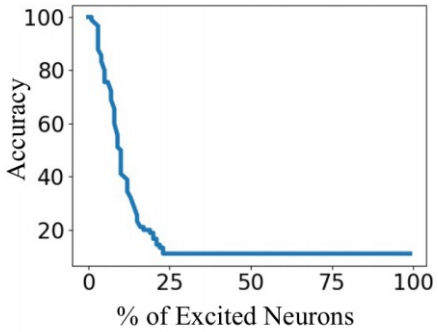
Attribution-based Offline Trojaned Model Detection Using Only Clean Data









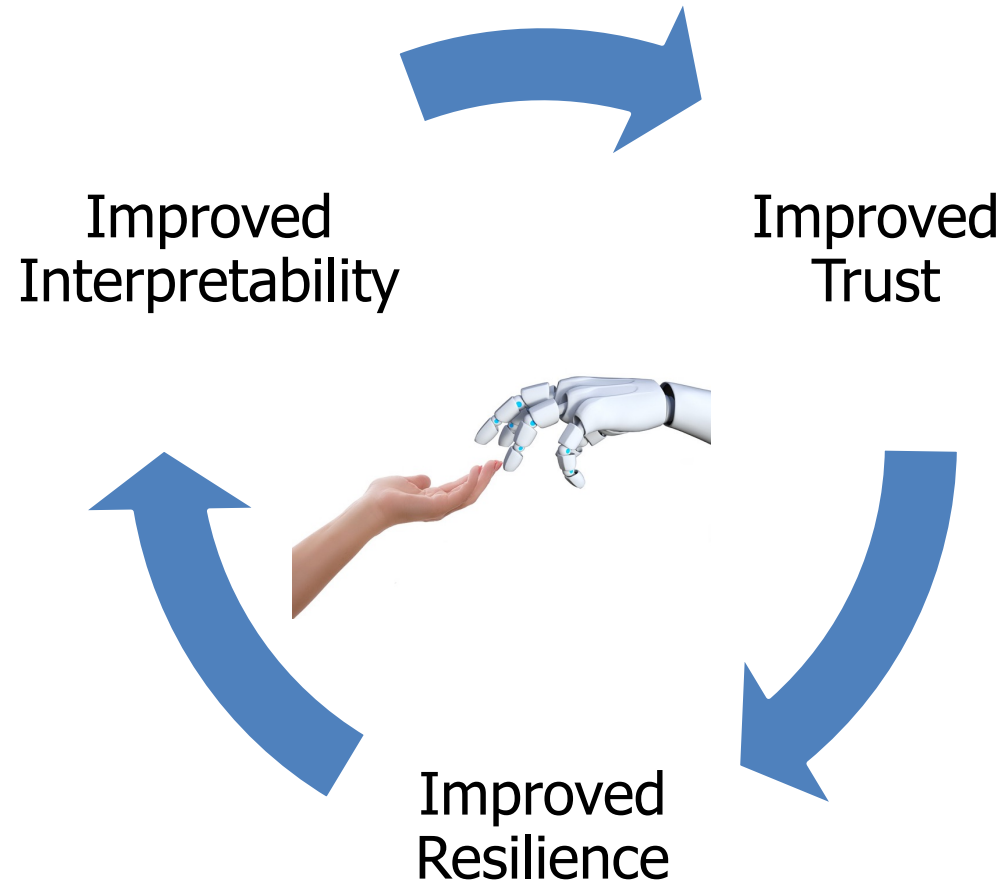


Model	Triggered-MNIST	TrojAI-Round1	TrojAI-Round2	TrojAI-Round3
Cassandra [62]	$\mathbf{0.97 \pm 0.010}$	0.88 ± 0.006	0.59 ± 0.096	0.71 ± 0.026
Neural Cleanse [55]	0.70 ± 0.045	0.50 ± 0.030	0.63 ± 0.043	0.61 ± 0.064
ULP [28]	0.54 ± 0.051	0.55 ± 0.058	–	–
TrinityAI-Conv-IG	0.89 ± 0.024	0.87 ± 0.020	0.73 ± 0.014	0.71 ± 0.038
TrinityAI-Tx-IG	0.95 ± 0.022	0.89 ± 0.029	0.75 ± 0.033	$\mathbf{0.72 \pm 0.038}$
TrinityAI-Conv-GradxAct	0.87 ± 0.030	0.88 ± 0.027	0.74 ± 0.030	0.67 ± 0.036
TrinityAI-GradxAct	0.96 ± 0.014	$\mathbf{0.90 \pm 0.027}$	$\mathbf{0.76 \pm 0.027}$	0.66 ± 0.029

Detecting Trojaned DNNs – ASAC'21, CVPR'22

Three Coupled Challenges in AI

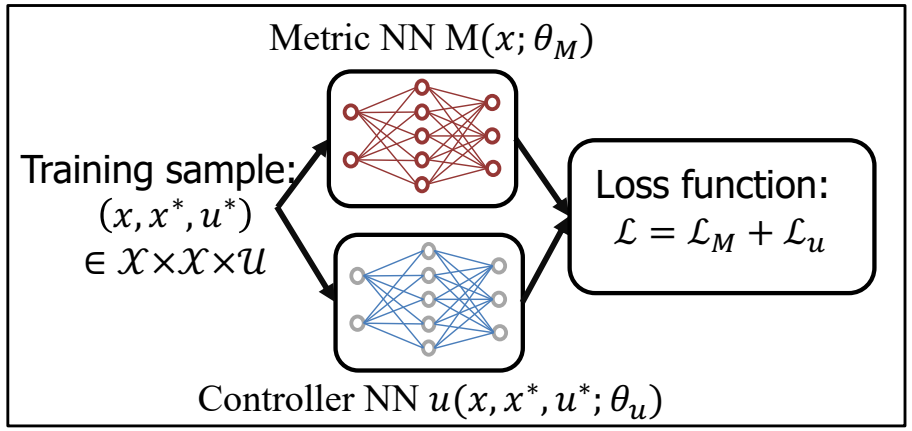
The dependency between Trust, Resilience and Interpretability also creates a virtuous cycle.



Simultaneously improvement in trustworthiness, resilience and interpretability is critical for their use in high-assurance systems and in human-machine teams.

Trinity-AI for Safety-critical Systems

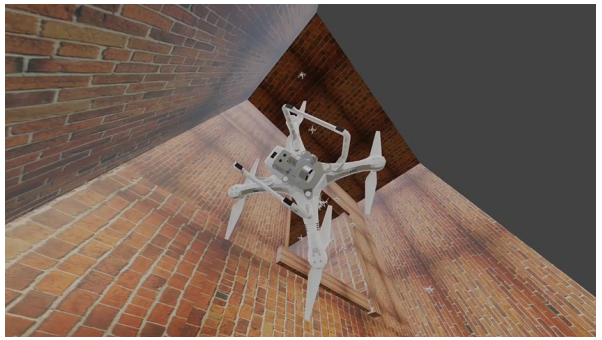
Assured Autonomy: High-assurance Reinforcement Learning for Control



Perturbed system $\dot{x} = f(x) + B(x)u + d(t)$

Theorem: Let $\underline{m}I \preccurlyeq M(x) \preccurlyeq \bar{m}I$. Assume that $\|d(t)\| \leq \epsilon$, then

$$\|x(t) - x^*(t)\| \leq \frac{R_0}{\sqrt{\underline{m}}} e^{-\lambda t} + \sqrt{\frac{\bar{m}}{\underline{m}}} \frac{\epsilon}{\lambda} \left(1 - e^{-\lambda t}\right)$$



where $R_0 = \int_{x(0)}^{x^*(0)} \sqrt{\delta_x^T M(x) \delta_x}$ is the initial geodesic distance between $x(0)$ and $x^*(0)$ under metric $M(x)$.

RV'17, NASA'17, Allerton Control'18, NeurIPS'18, AAAI-SS'19, JAR'18, SafeComp'20, CoRL'20

Thank you !

https://nusci.csl.sri.com/

Projects

Symbiotic Design for Cyber Physical Systems (DARPA)

Trojans in Artificial Intelligence (IARPA)

Assured Autonomy (DARPA)

Internet Of Battlefield Things (Army Research Lab)

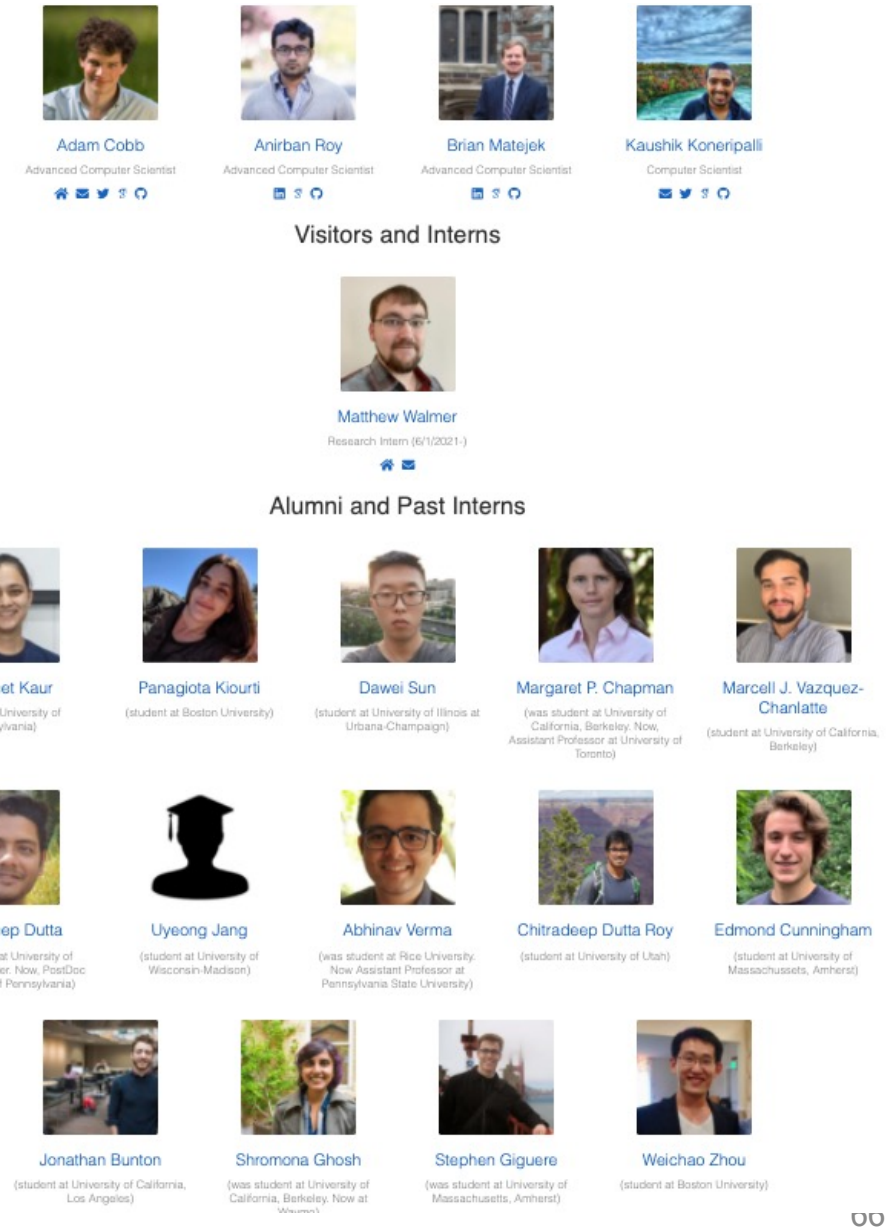
Quantum Computing and Quantum Machine Learning (IR&D)

Intent-Defined Adaptive Software (DARPA)

Self-Improving Cyber-Physical Systems (NSF CPS Small)

Duality-Based Algorithm Synthesis (NSF EAGER)

Technology to Review Online Videos for Education (NSF EAGER)



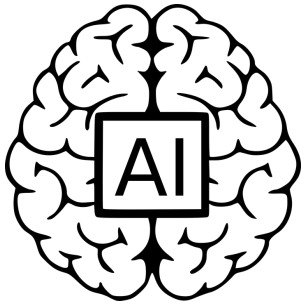

SRI
International

Creative AI: Co-Designer for Symbiotic Design of CPS

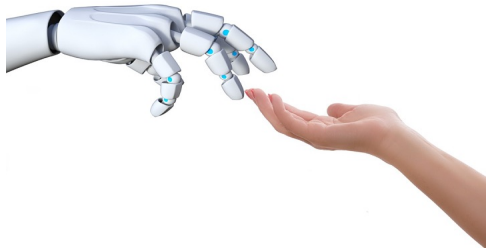
Faster UQ surrogates

Risk-aware exploration

Concept jumps



DARPA: Symbiotic Design



Domain knowledge

Debugging failures

Intuitive preferences

Commercialization

ML Surrogates for Slow Scientific Models

- Computational Fluid Dynamics Models
- Flight Dynamics Models

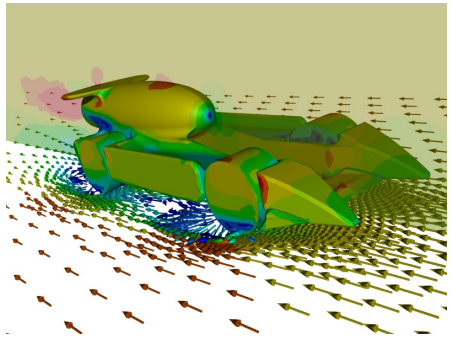
Uncertainty-aware ML for exploration

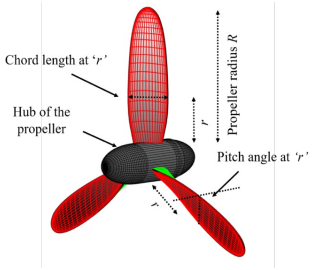
- Exploration leads ML out of its training distribution.
- Risk-aware exploration

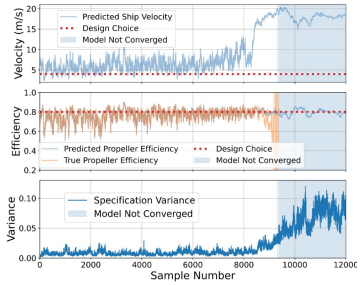
Design for Novelty and not just Optimization

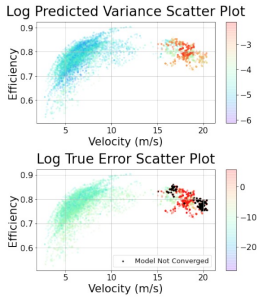
- Produce diverse designs outside human design silos
- Exploit symbolic knowledge

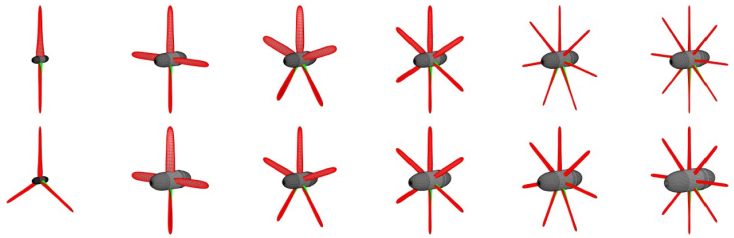
Exploit Rapidly Evolving SOTA in AR/VR for Simulation and Visualization

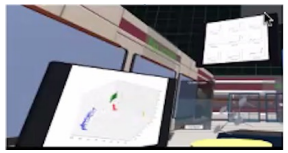


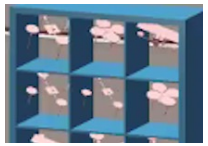


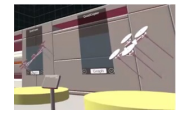




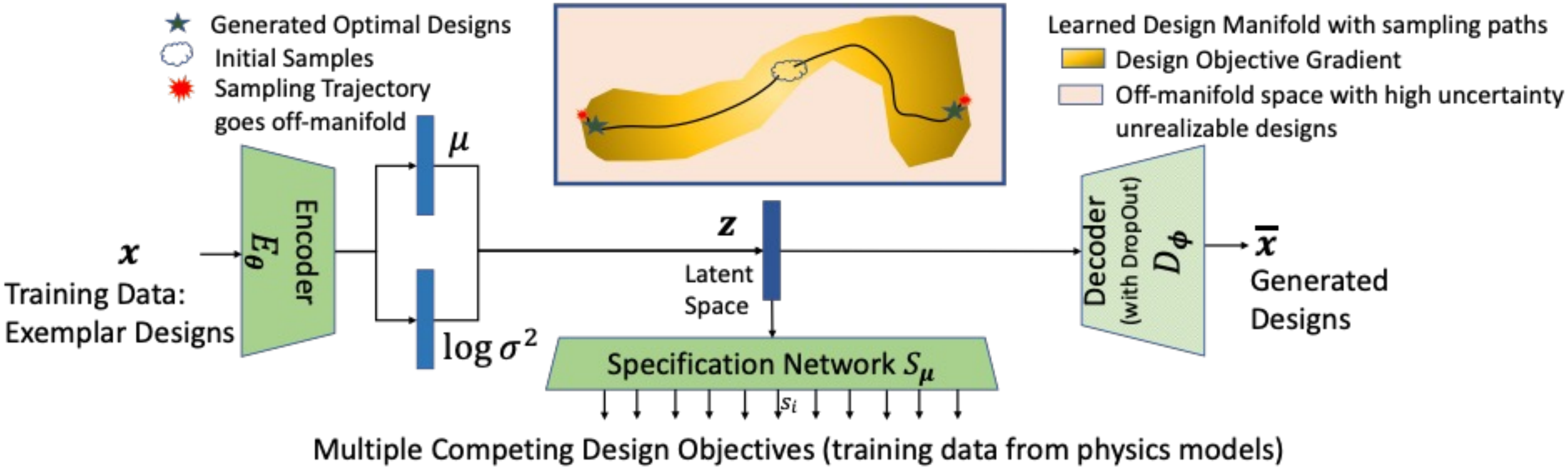








Hamiltonian MCMC over Design Manifold

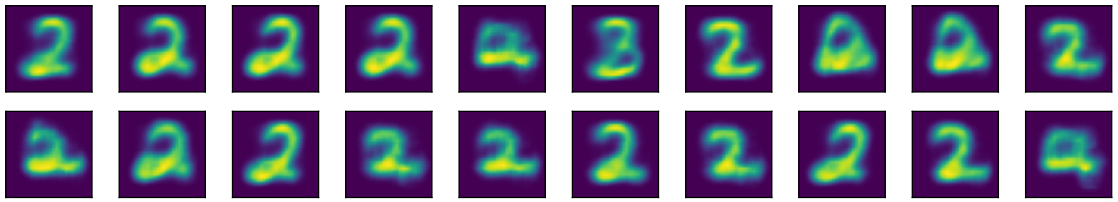


- Our approach uses exemplar designs to learn a variational encoder (VAE) where the decoder is trained with dropout.
- The specification network predicts the design objectives from the latent space.
- The VAE and the specification network are jointly trained on the exemplar designs and their evaluation on physics models. In the design exploration stage, we condition on the new target design objectives and use temperature annealed HMC to sample the latent space, moving towards optimal designs exploiting the gradient information.
- High variance/uncertainty implies off-manifold designs that may not be unrealizable.
- Controlling HMC walk yields diverse designs.

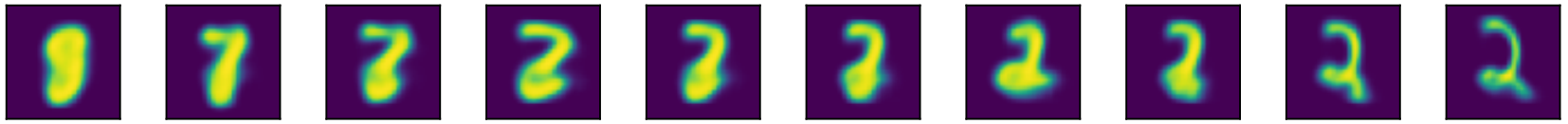
Illustrative example with MNIST

- As an initial example, we demonstrate the thickness and value of a digit as specifications of the design of a handwritten digit

- Just conditioning on the design specification of digit “2”



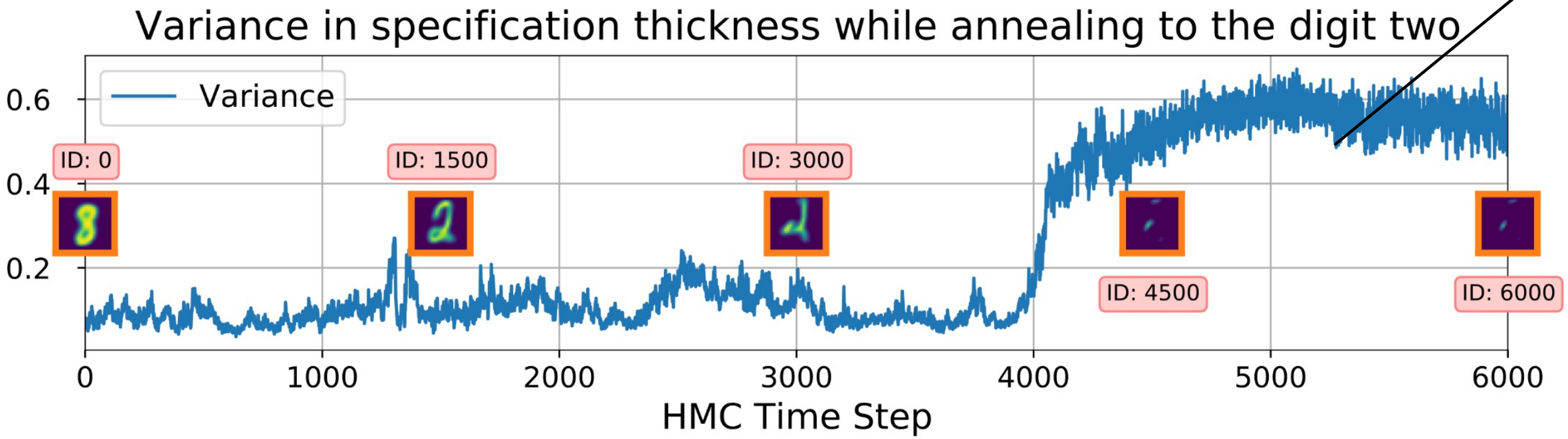
- Conditioning on digit “2” and reducing line thickness:



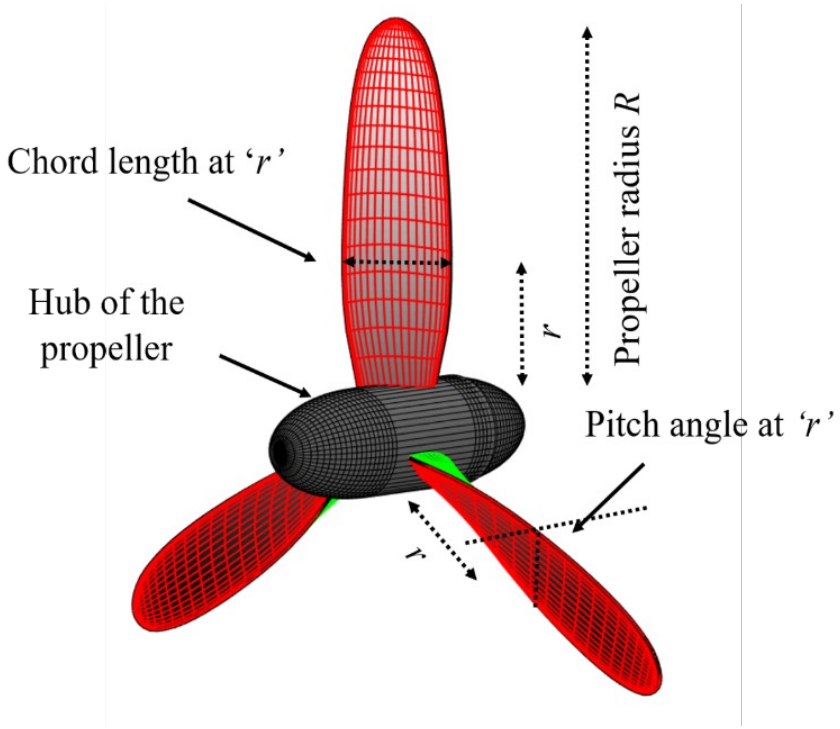
Anneal digit thickness temperature

Illustrative example with MNIST

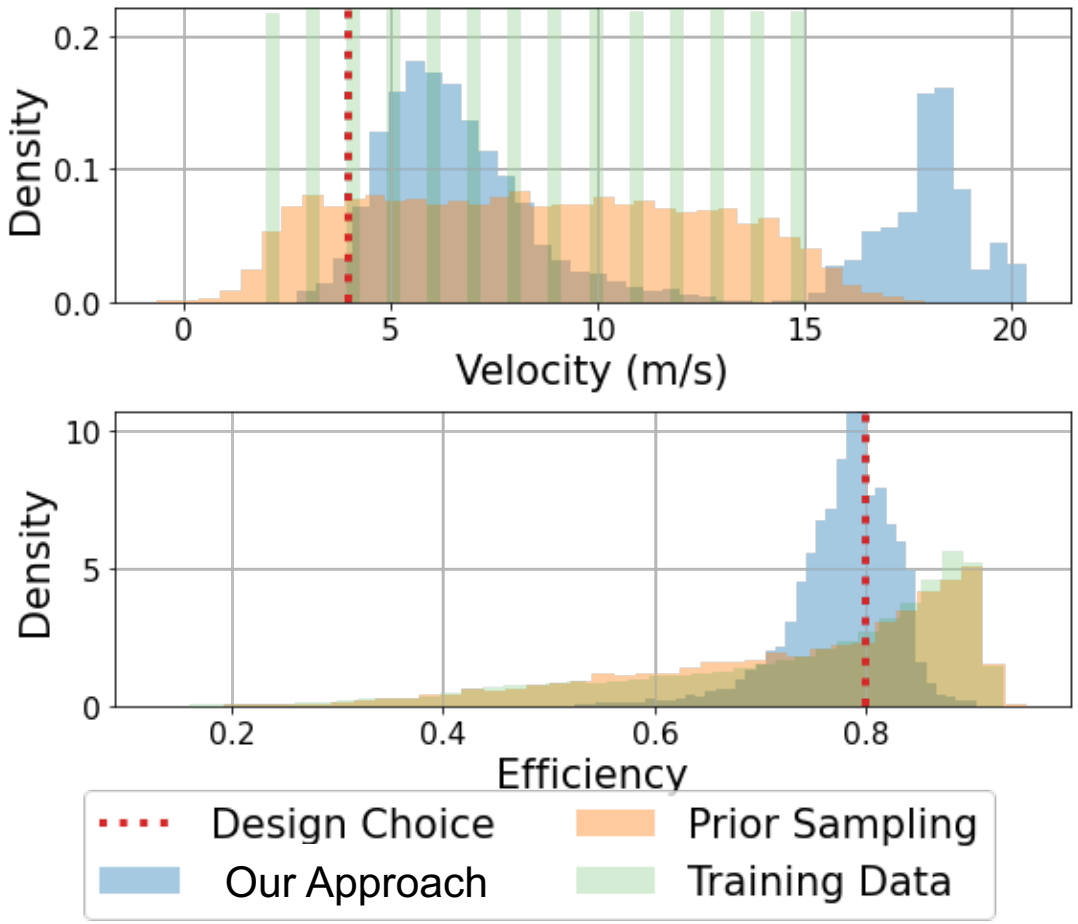
To ensure we explore the regions of the design manifold that we can trust, we employ uncertainty quantification to analyze the expected performance of a proposed design.



OpenProp Propeller Design

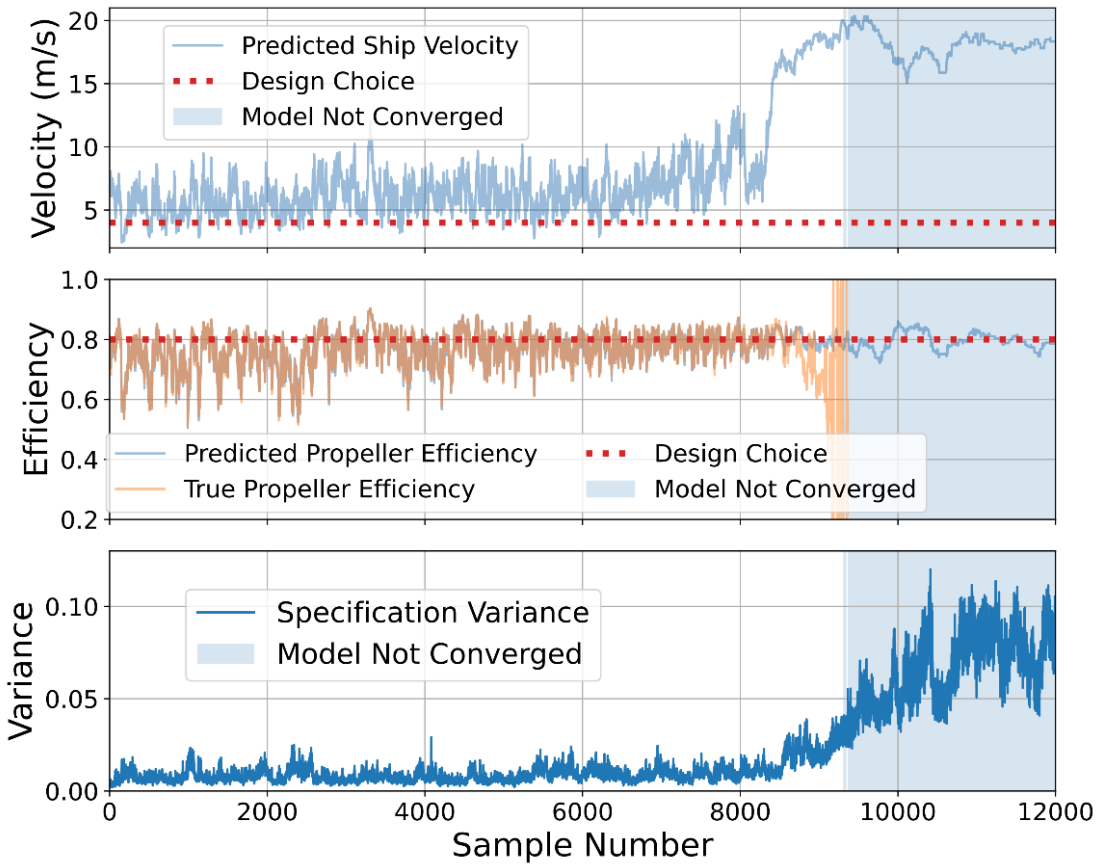


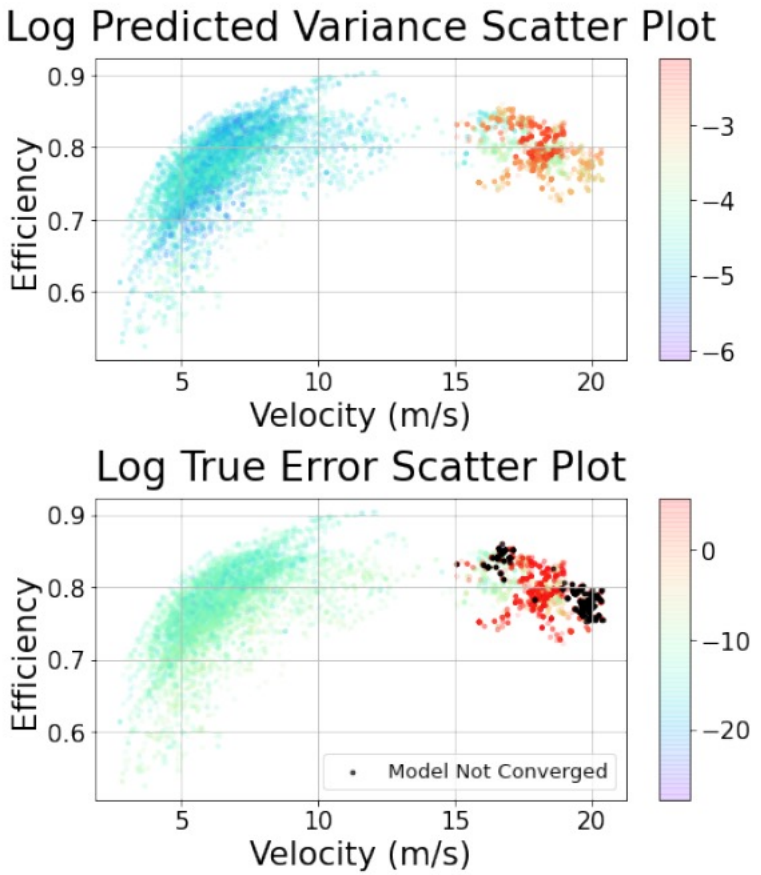
High efficiency at low velocity



Histograms of two competing design objectives. Simply sampling from the Gaussian prior in the latent space is not sufficient.

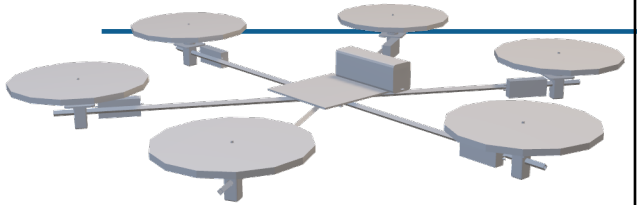
OpenProp Propeller Design



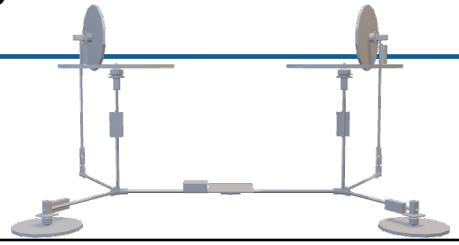


Sample trajectories of the velocity and the propeller efficiency, as well as the corresponding variance on the objectives. Around sample ID 9000, we see high velocities with high efficiency, but the corresponding variance is high, suggesting these are unreliable designs.

Hover Time: 345.6
Flight Dist.: 6197

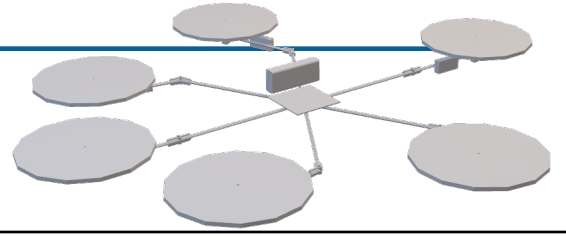


Hover Time: 57.3
Flight Dist.: 2005



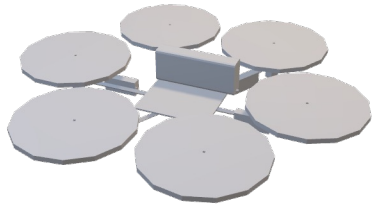
Hover Time: 254.1
Flight Dist.: 6733

Hover Time: 462.7
Flight Dist.: 8331

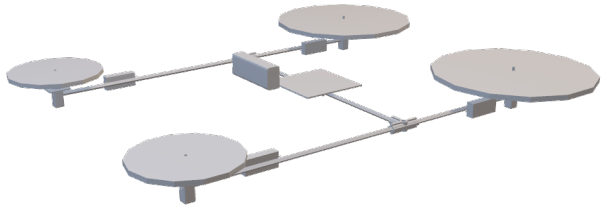


Hover Time: 260.2
Flight Dist.: 5861

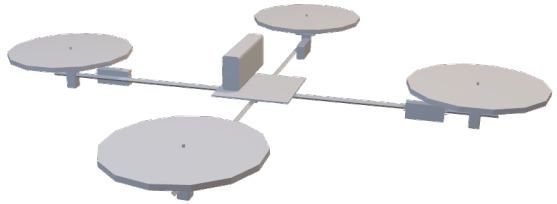
Hover Time: 425.2
Flight Dist.: 9013



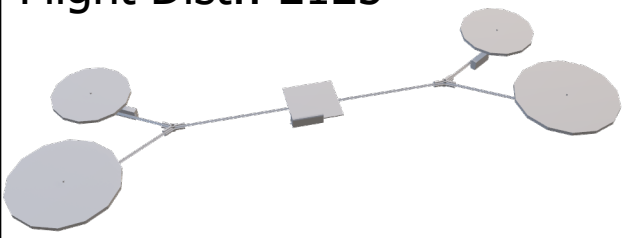
Hover Time: 310.7
Flight Dist.: 6885



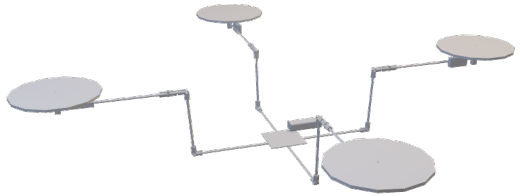
Hover Time: 320.0
Flight Dist.: 6390



Hover Time: 91.0
Flight Dist.: 2125



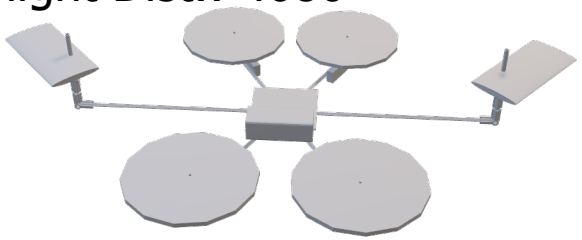
Hover Time: 522.1
Flight Dist.: 7460



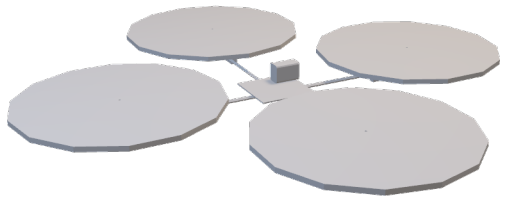
Hover Time: 237.3
Flight Dist.: 3896

Hover Time: 265.1
Flight Dist.: 4790

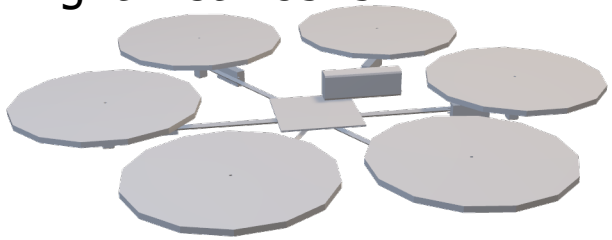
Hover Time: 341.2
Flight Dist.: 4680



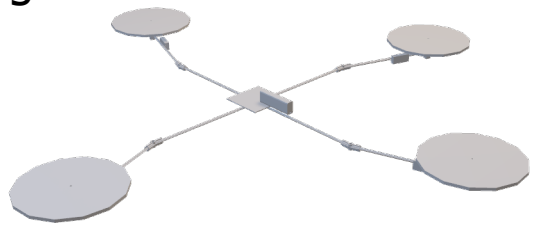
Hover Time: 143.4
Flight Dist.: 3527



Hover Time: 441.5
Flight Dist.: 8575



Hover Time: 300.0
Flight Dist.: 4120



Diversity of UAM Designs