TRINITY: Trust, Resilience and Interpretability in AI

Susmit Jha

Computer Science Laboratory

SRI

July, 2019

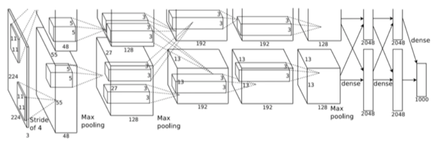
AI reaches human-level accuracy on benchmark datasets

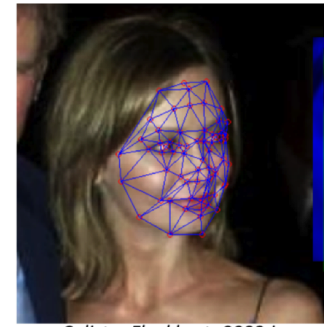
ImageNet Classification with Deep Convolutional Neural Networks. Krizhevsky et al, 2012

(a) Siberian husky

(b) Eskimo dog

Model	Top-1	Top-5
Sparse coding [2]	*47.1%*	*28.2%*
SIFT + FVs [24]	*45.7%*	*25.7%*
CNN	**37.5%**	**17.0%**

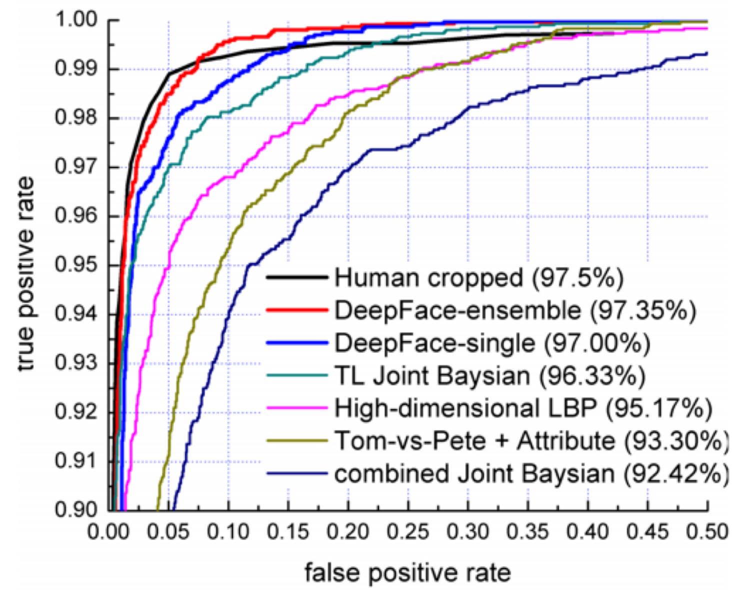




Calista_Flockhart_0002.jpg
Detection & Localization

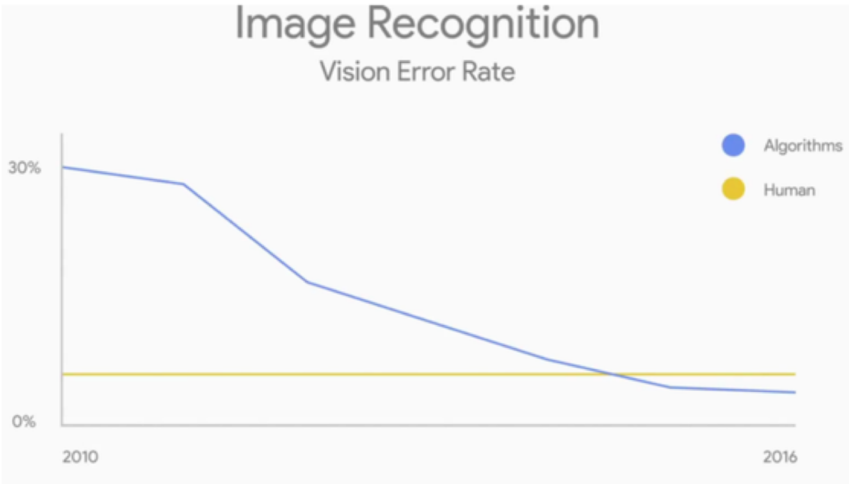
Team	Year	Place	Error (top-5)	Uses external data
SuperVision	2012	1st	16.4%	no
SuperVision	2012	1st	15.3%	Imagenet 22k
Clarifai	2013	1st	11.7%	no
Clarifai	2013	1st	11.2%	Imagenet 22k
MSRA	2014	3rd	7.35%	no
VGG	2014	2nd	7.32%	no
GoogLeNet	2014	1st	6.67%	no

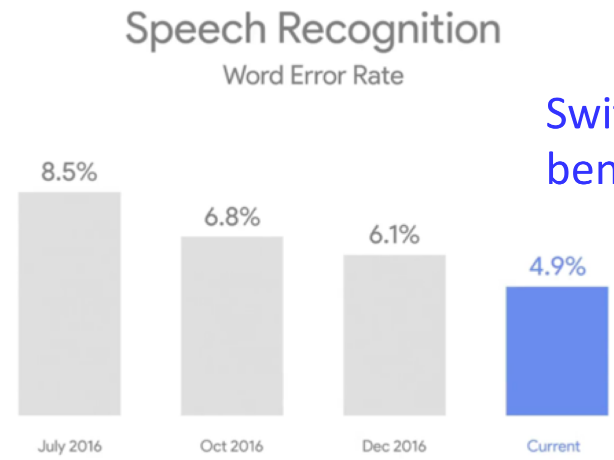
Going deeper with convolutions. (Inception) C Szegedy et al, 2014



Face Detection. Taigman et al, 2014

AI reaches human-level accuracy on benchmark datasets





Switchboard benchmark

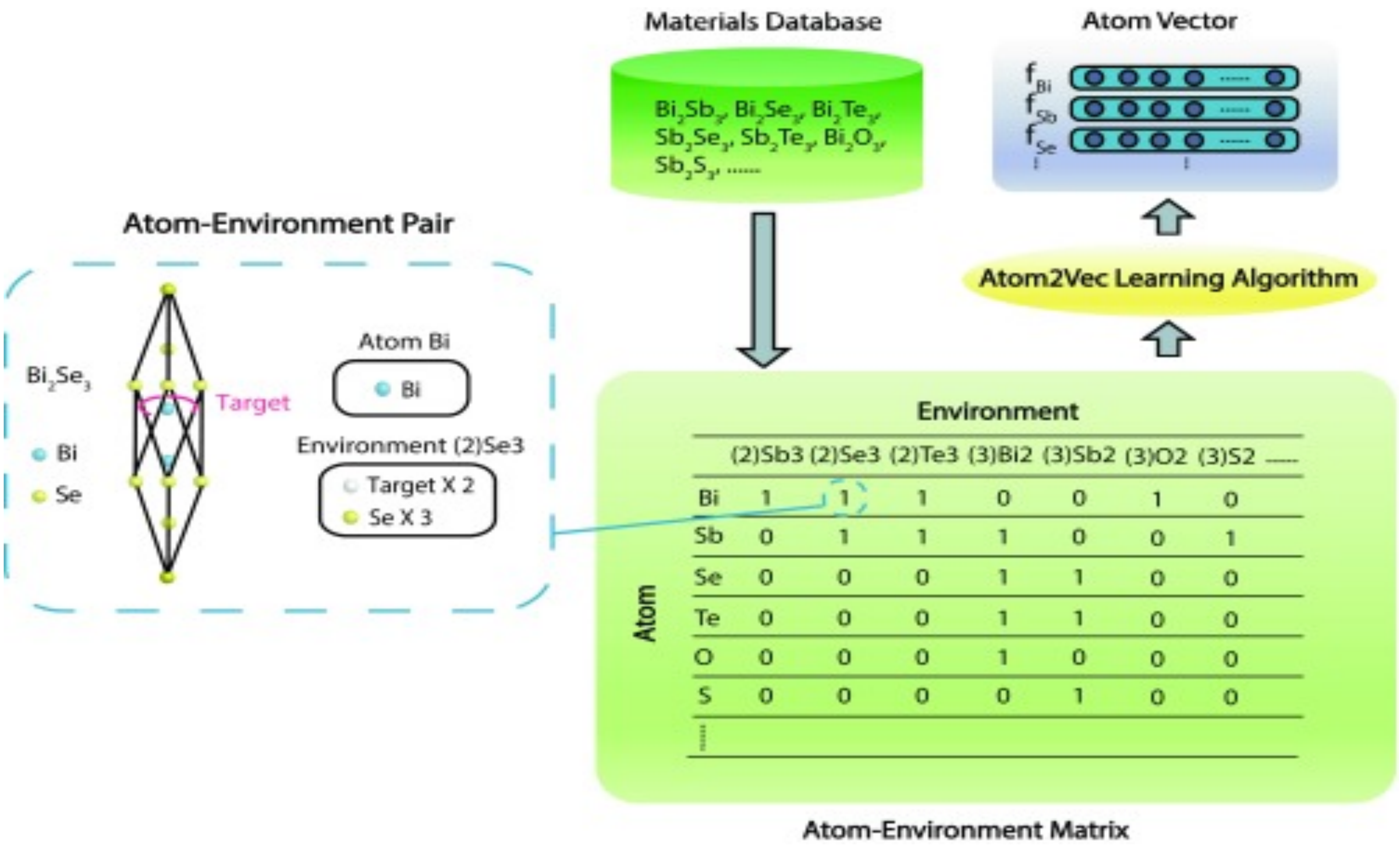
Google I/O, 2017

Microsoft recently reached a new milestone in its ability to recognize conversational speech, achieving a 5.1% word error rate (WER). The achievement, detailed in a Sunday blog post, bests Microsoft's previous record of 5.9% and is closer to human parity.

Microsoft, 2017

Solving CAPTCHA Goodfellow et al, 2013

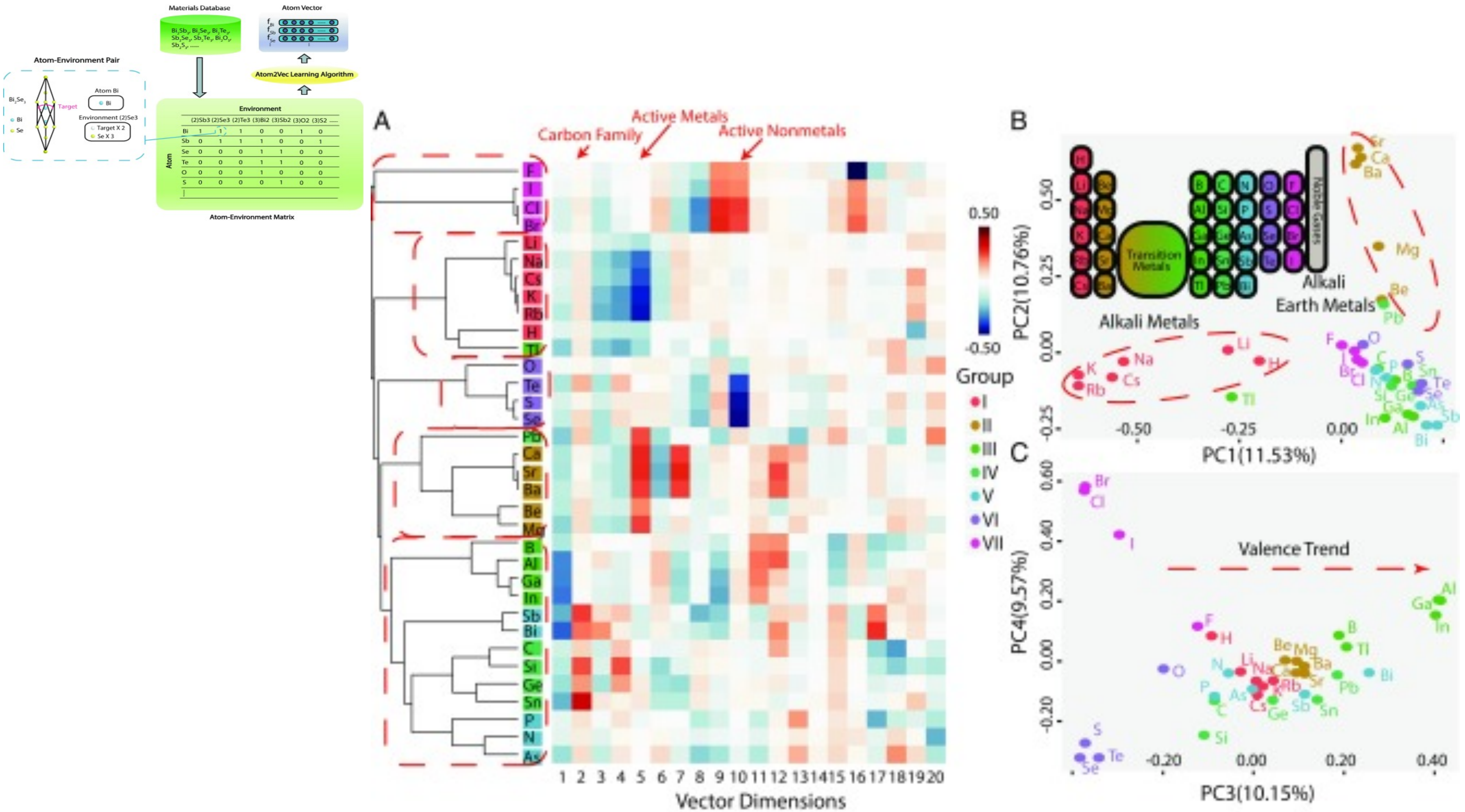
More recent results



Atom	(2)Sb3	(2)Se3	(2)Te3	(3)Bi2	(3)Sb2	(3)O2	(3)S2 ......
Bi	1	1	1	0	0	1	0
Sb	0	1	1	1	0	0	1
Se	0	0	0	1	1	0	0
Te	0	0	0	1	1	0	0
O	0	0	0	1	0	0	0
S	0	0	0	0	1	0	0
⋮							

Learning atoms for materials discovery. Zhou et. al. (PNAS), 2018

More recent results

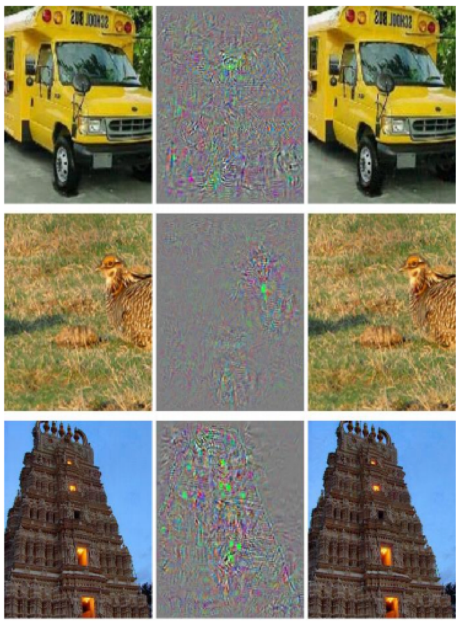


Learning atoms for materials discovery. Zhou et. al. (PNAS), 2018

AI in Adversarial Settings

Machine learning very susceptible
to adversarial attacks.

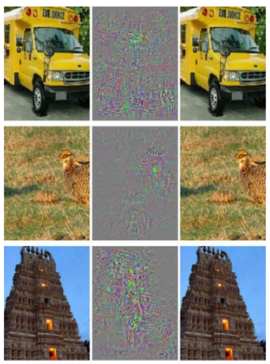
Szegedy et al, 2013, 2014

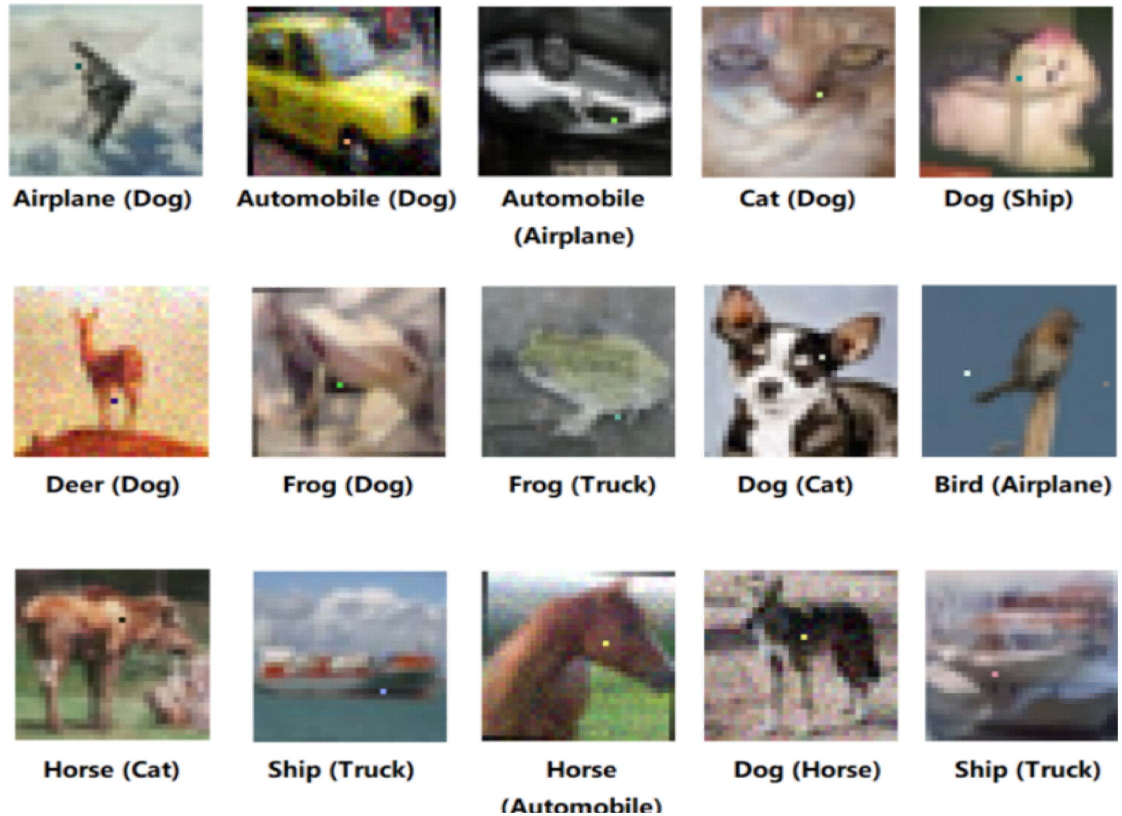


AI in Adversarial Settings

Machine learning very susceptible to adversarial attacks.

Szegedy et al, 2013, 2014



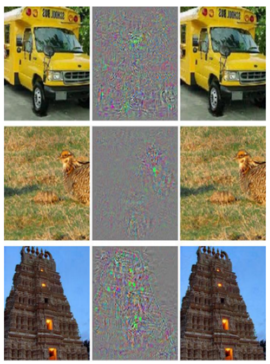


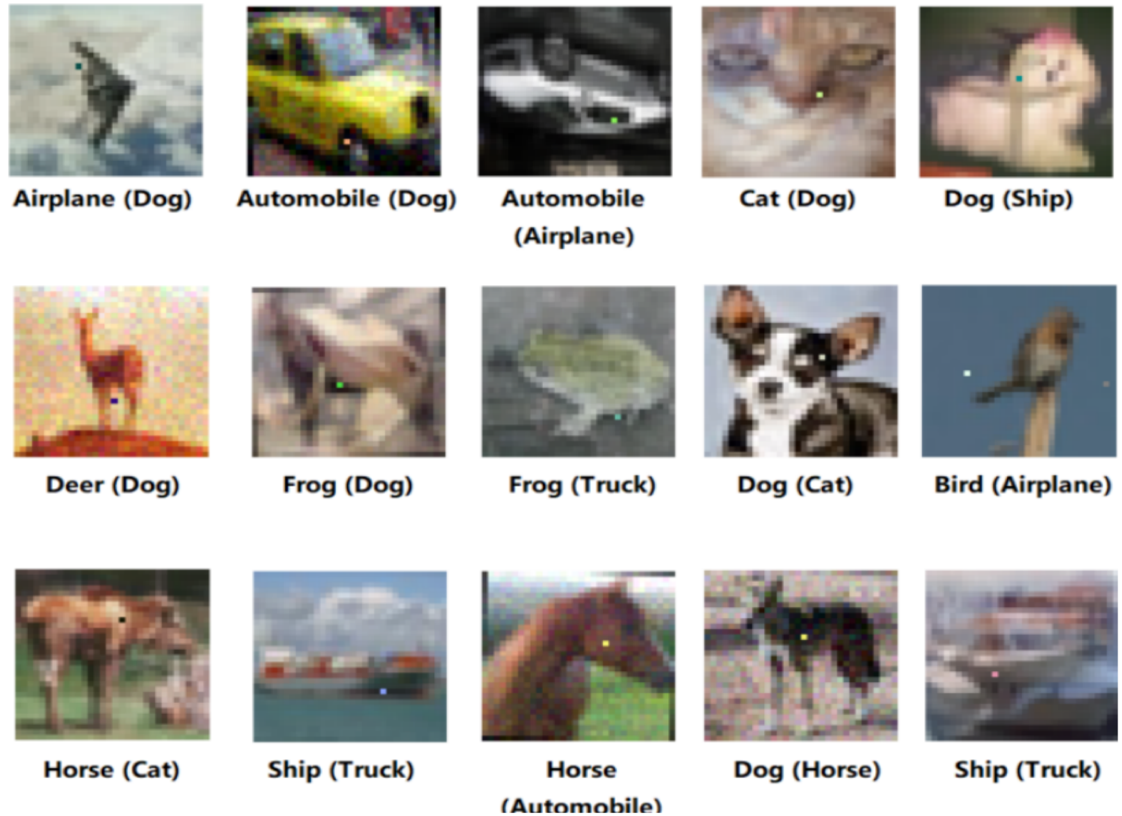
One pixel attack for fooling deep neural networks. Su et. al., 2017

AI in Adversarial Settings

Machine learning very susceptible to adversarial attacks.

Szegedy et al, 2013, 2014



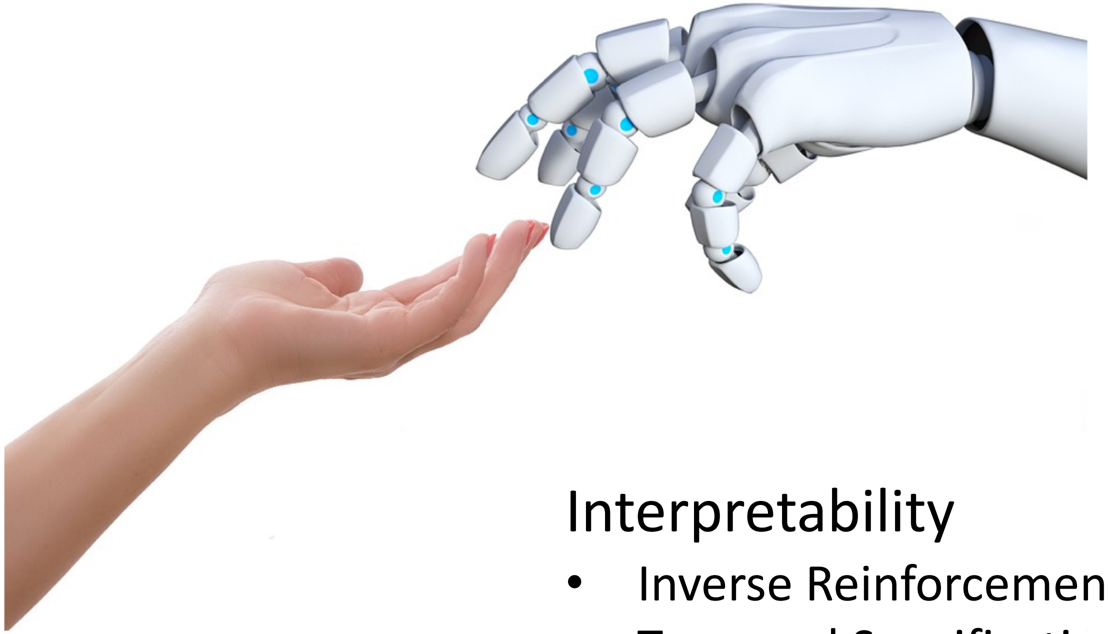


Only allowed to modify the value of 1 pixel. 70.97% of the natural images can be perturbed to at least one target class by modifying just one pixel with 97.47% confidence on average.

Rest of the Talk

Trust

- Global Assume/Guarantee Contracts on DNNs
- Extracting and Integrating Temporal Logic into Learned Control



Interpretability

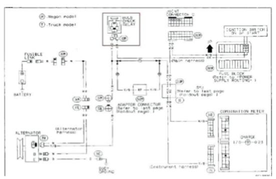
- Inverse Reinforcement Learning of Temporal Specifications

Resilience

- Adversarial Robustness

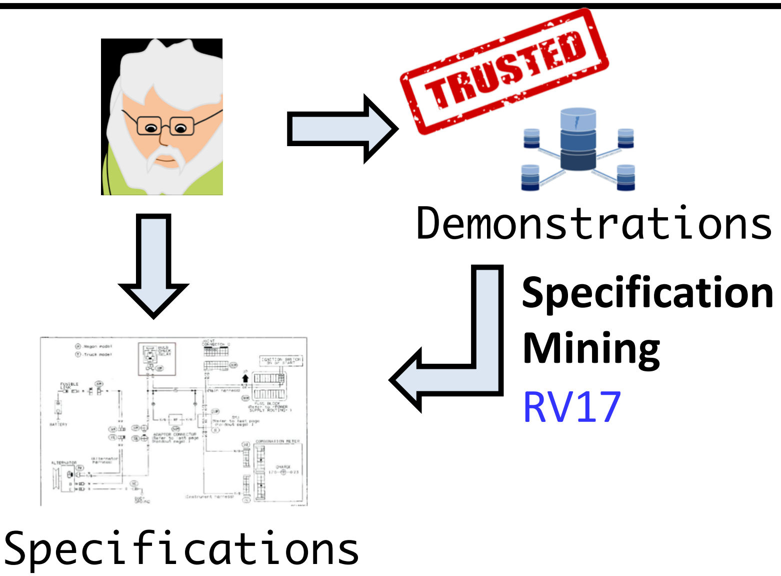
TRINITY: Trust, Resilience and Interpretability

Demonstrations



Specifications

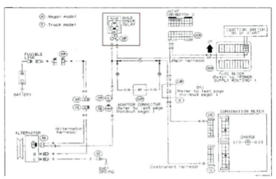
TRINITY: Trust, Resilience and Interpretability



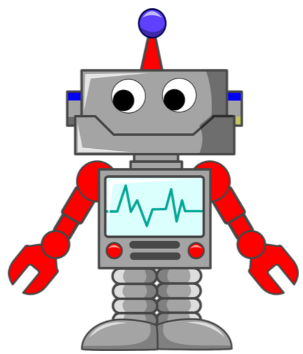
TRINITY: Trust, Resilience and Interpretability

Demonstrations

Specification Mining
RV17

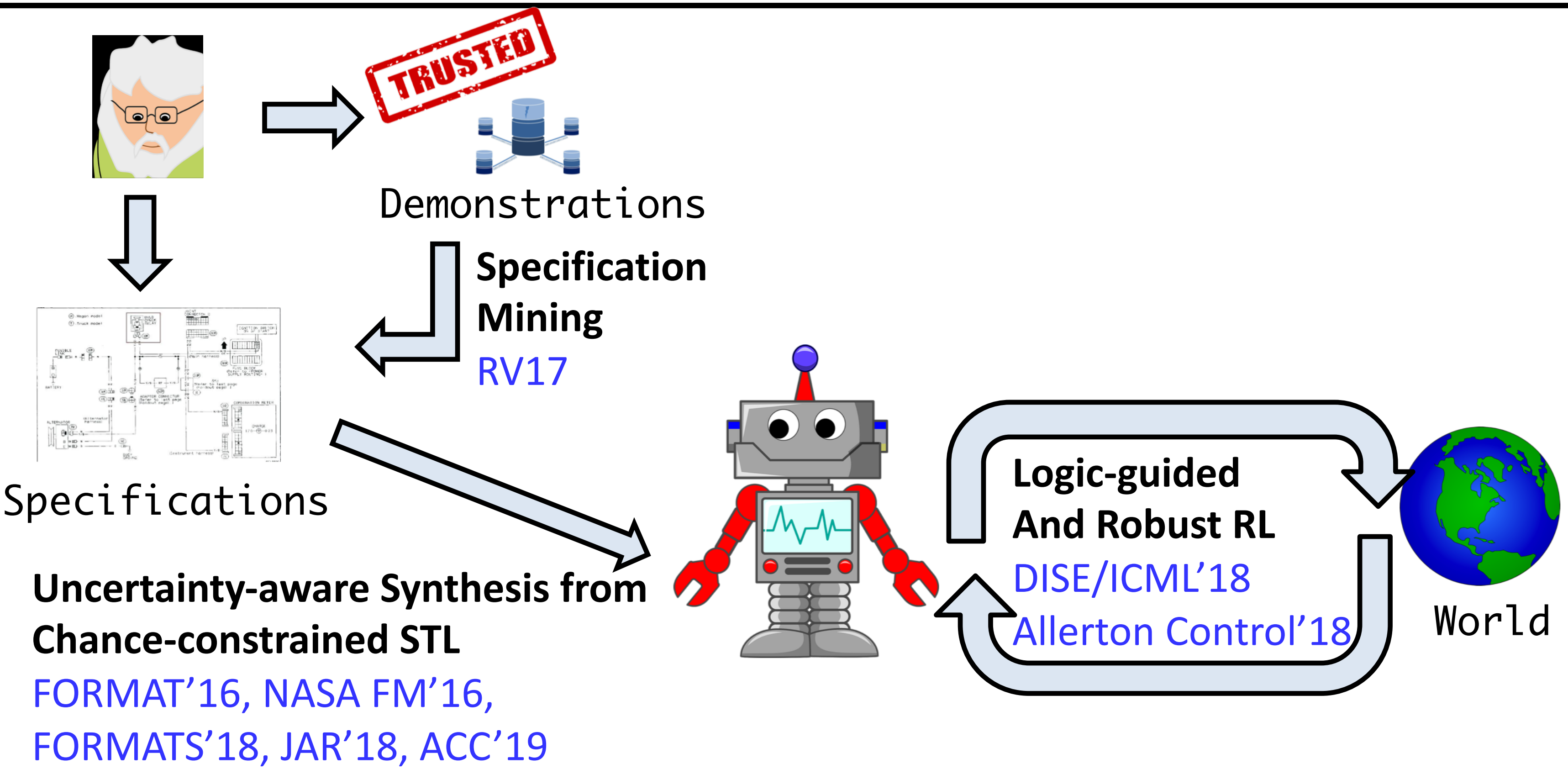


Specifications

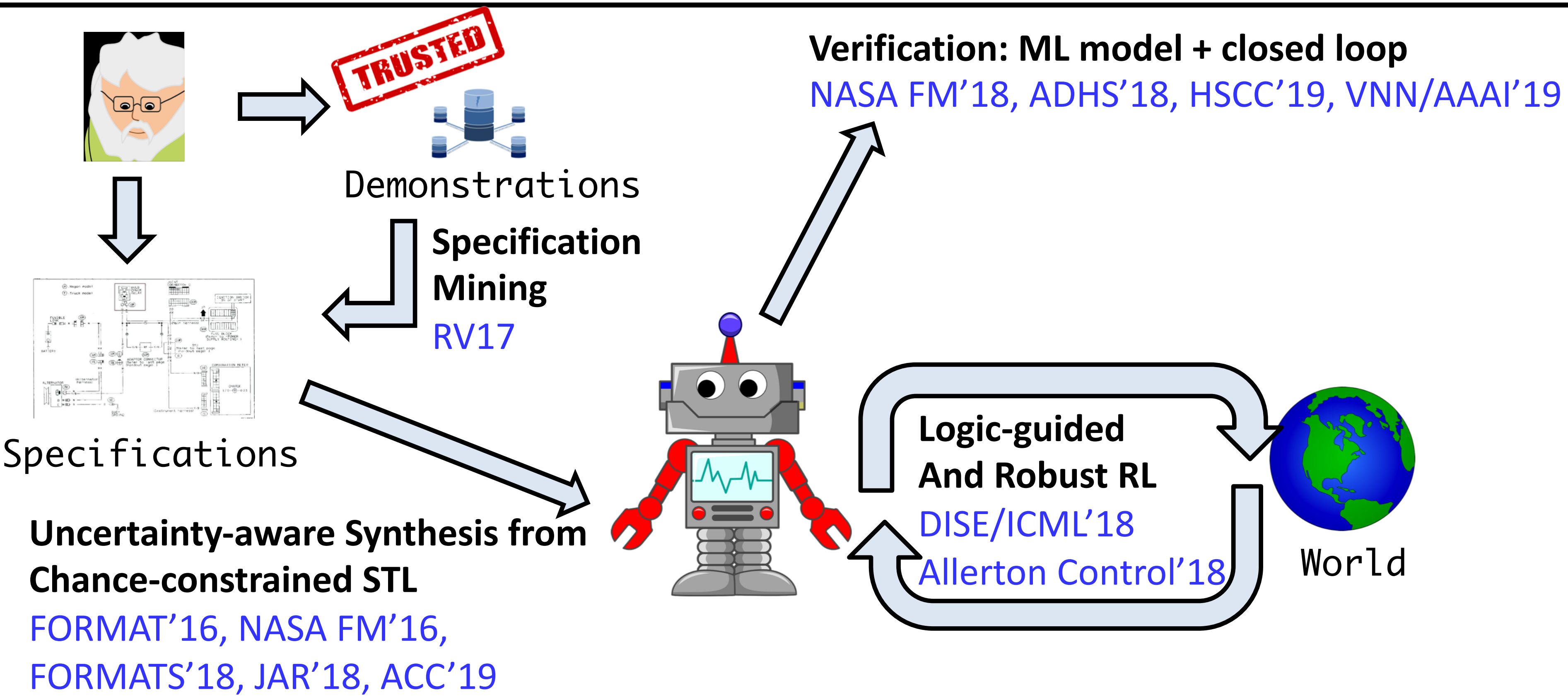


Uncertainty-aware Synthesis from Chance-constrained STL
FORMAT’16, NASA FM’16,
FORMATS’18, JAR’18, ACC’19

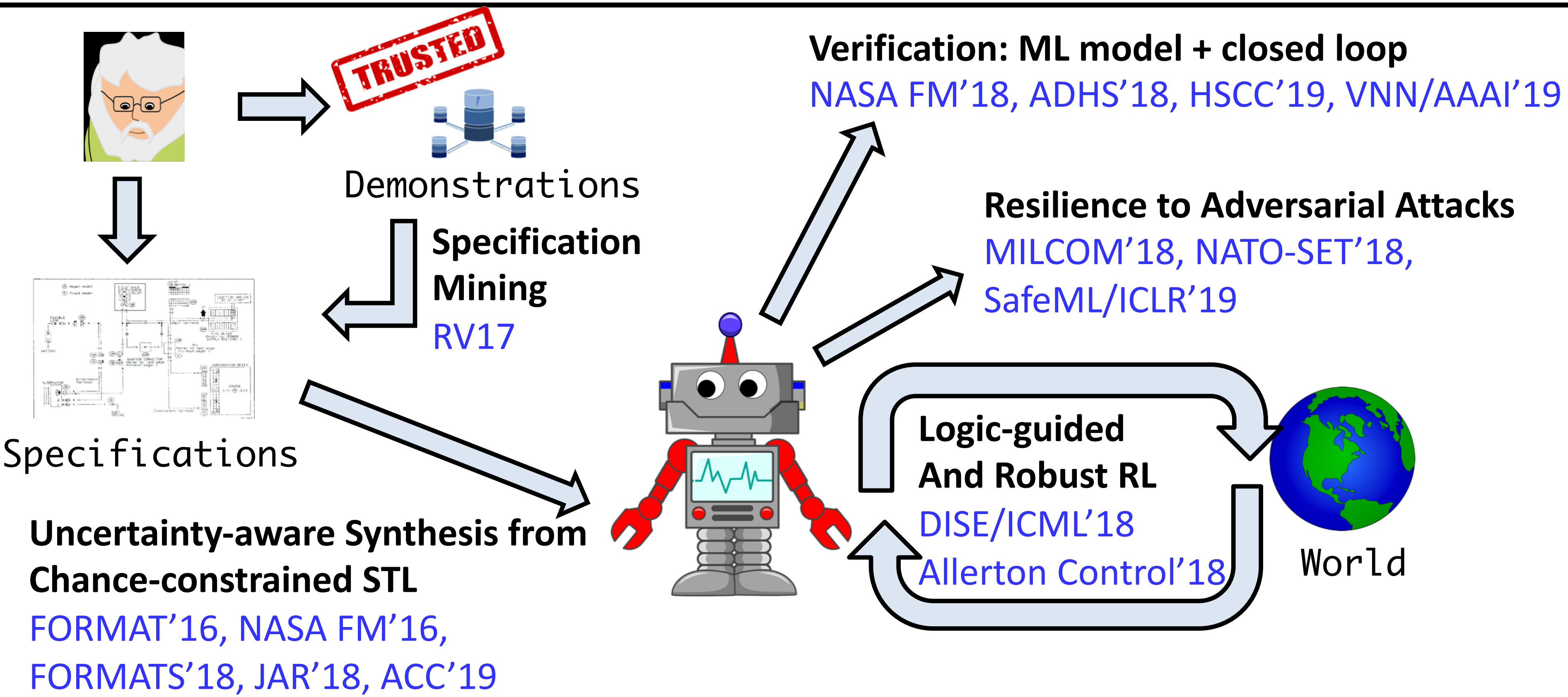
TRINITY: Trust, Resilience and Interpretability



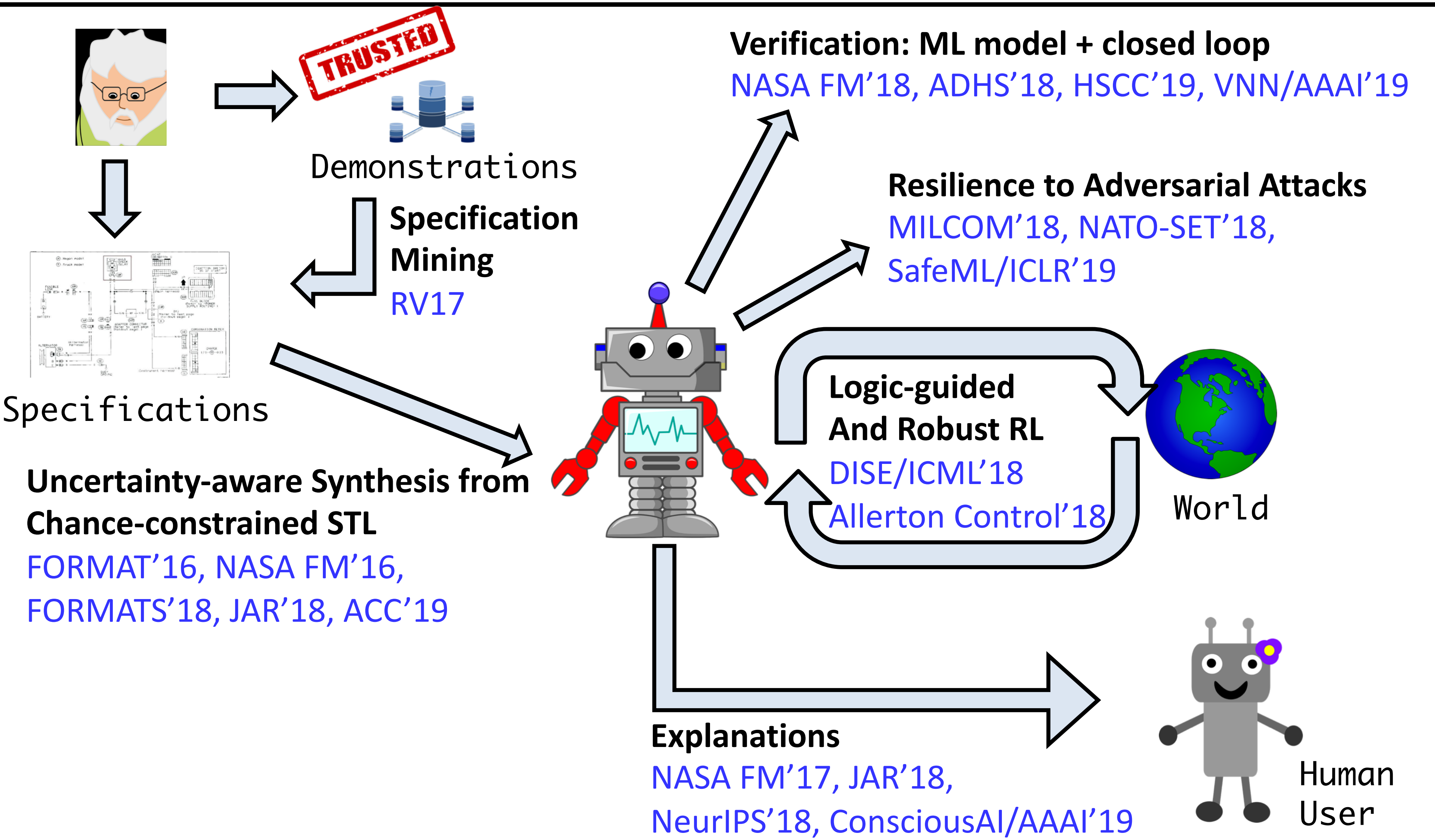
TRINITY: Trust, Resilience and Interpretability



TRINITY: Trust, Resilience and Interpretability



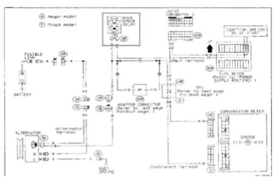
TRINITY: Trust, Resilience and Interpretability



TRINITY: Trust, Resilience and Interpretability

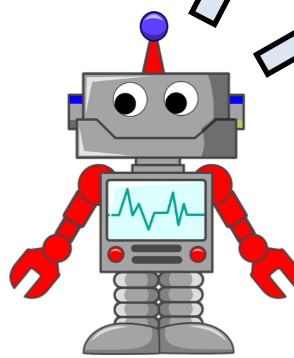
Demonstrations

Specification Mining
RV17



Specifications

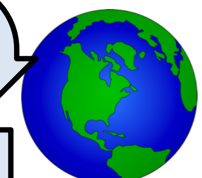
Uncertainty-aware Synthesis from Chance-constrained STL



Verification: ML model + closed loop
NASA FM’18, ADHS’18, HSCC’19, VNN/AAAI’19

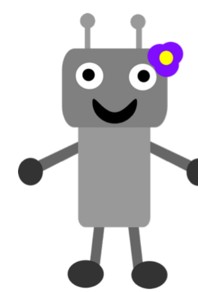
Resilience to Adversarial Attacks
MILCOM’18, NATO-SET’18, SafeML/ICLR’19

Logic-guided And Robust RL
DISE/ICML’18
Allerton Control’18



World

Explanations
NASA FM’17, JAR’18
NeurIPS’18, ConsciousAI/AAAI’19



Human User

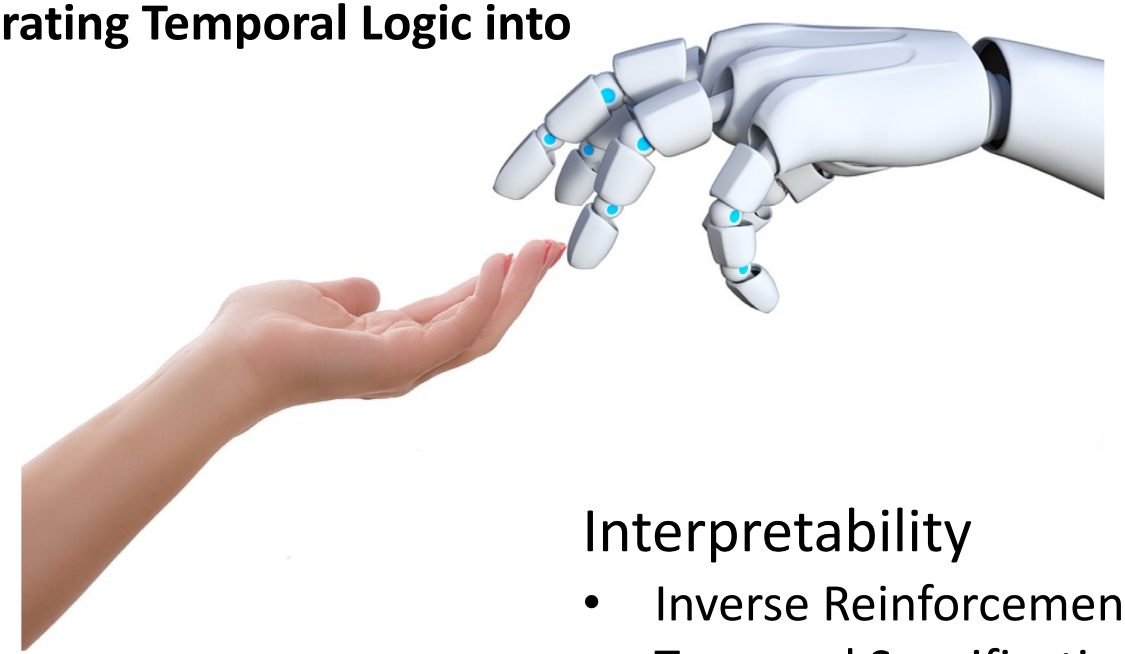
Ongoing Work

- U.S. Army Internet of Battlefield Things
- DARPA Assured Autonomy
- DARPA Competency-Aware Machine Learning

Rest of the Talk

Trust

- **Global Assume/Guarantee Contracts on DNNs**
- **Extracting and Integrating Temporal Logic into Learned Control**



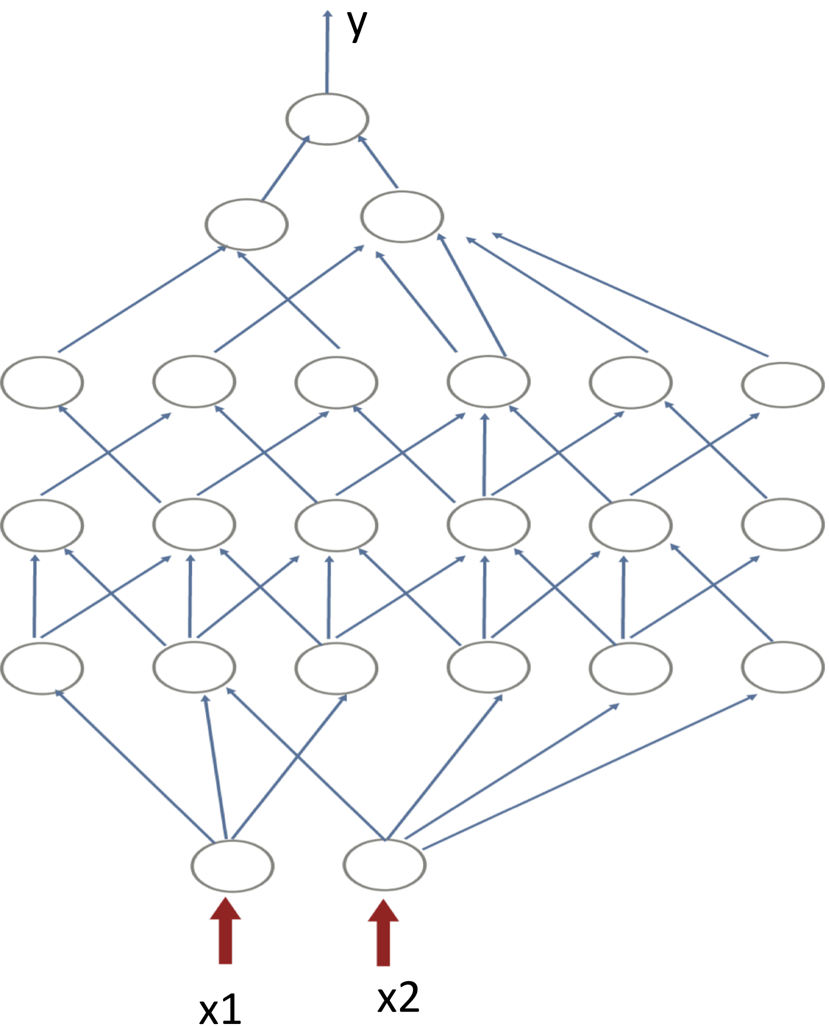
Interpretability

- Inverse Reinforcement Learning of Temporal Specifications

Resilience

- Adversarial Robustness

Formal Contracts on Feedforward Neural Networks

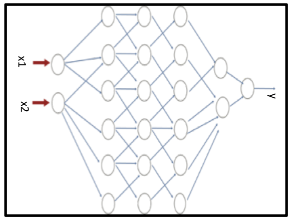


Example Specification.

Assumption: $L_1 \leq x_1 \leq U_1 \ \wedge L_2 \leq x_2 \leq U_2$

Guarantee: $L_o \leq y \leq U_0$

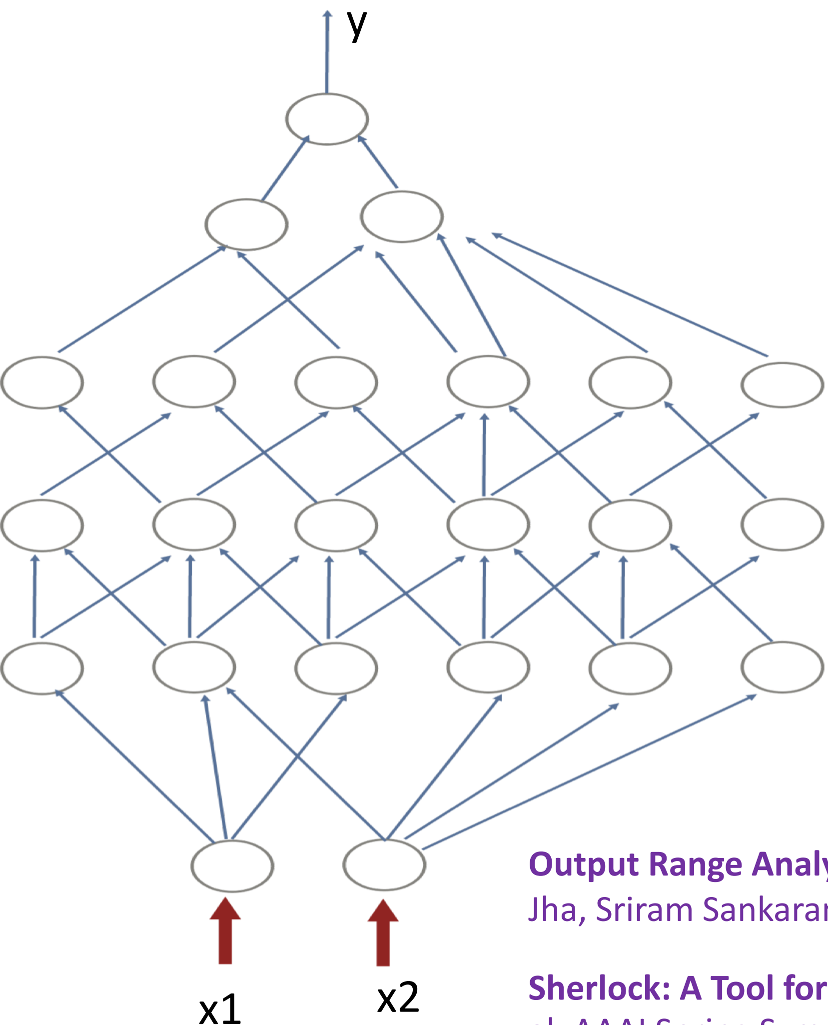
Assumption $\phi(x_1, x_2)$



Guarantee $\psi(x_1, x_2, y)$

Encapsulating ML components in A/G contracts can enable traditional Design by Contract approaches.

Formal Contracts on Feedforward Neural Networks



Example Specification.

Assumption: $L_1 \leq x_1 \leq U_1 \ \wedge L_2 \leq x_2 \leq U_2$

Guarantee: $L_o \leq y \leq U_0$

Key Idea: Can we improve scalability by combining local search (linear programming + gradient descent) with sparse calls to global search (mixed integer linear programming) ?

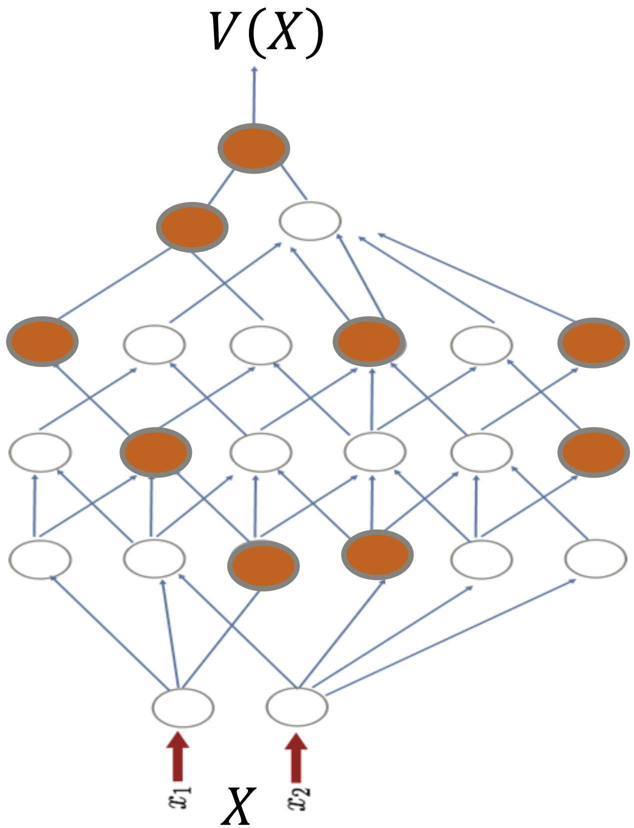
Implemented in **publicly available tool** since January, 2018 : Sherlock

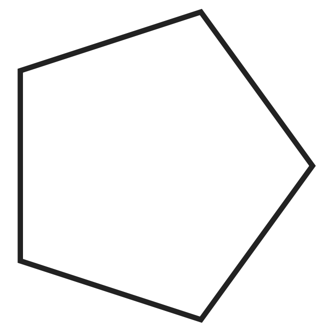
https://github.com/souradeep-111/sherlock

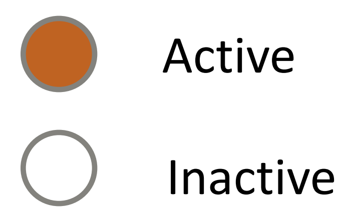
Output Range Analysis for Deep Feedforward Neural Networks. Souradeep Dutta, Susmit Jha, Sriram Sankaranarayanan, Ashish Tiwari. NASA Formal Methods (NFM), 2018

Sherlock: A Tool for Verification of Deep Neural Networks. Dutta et. al. AAAI Spring Symposium on Verification of Neural Networks, 2019.

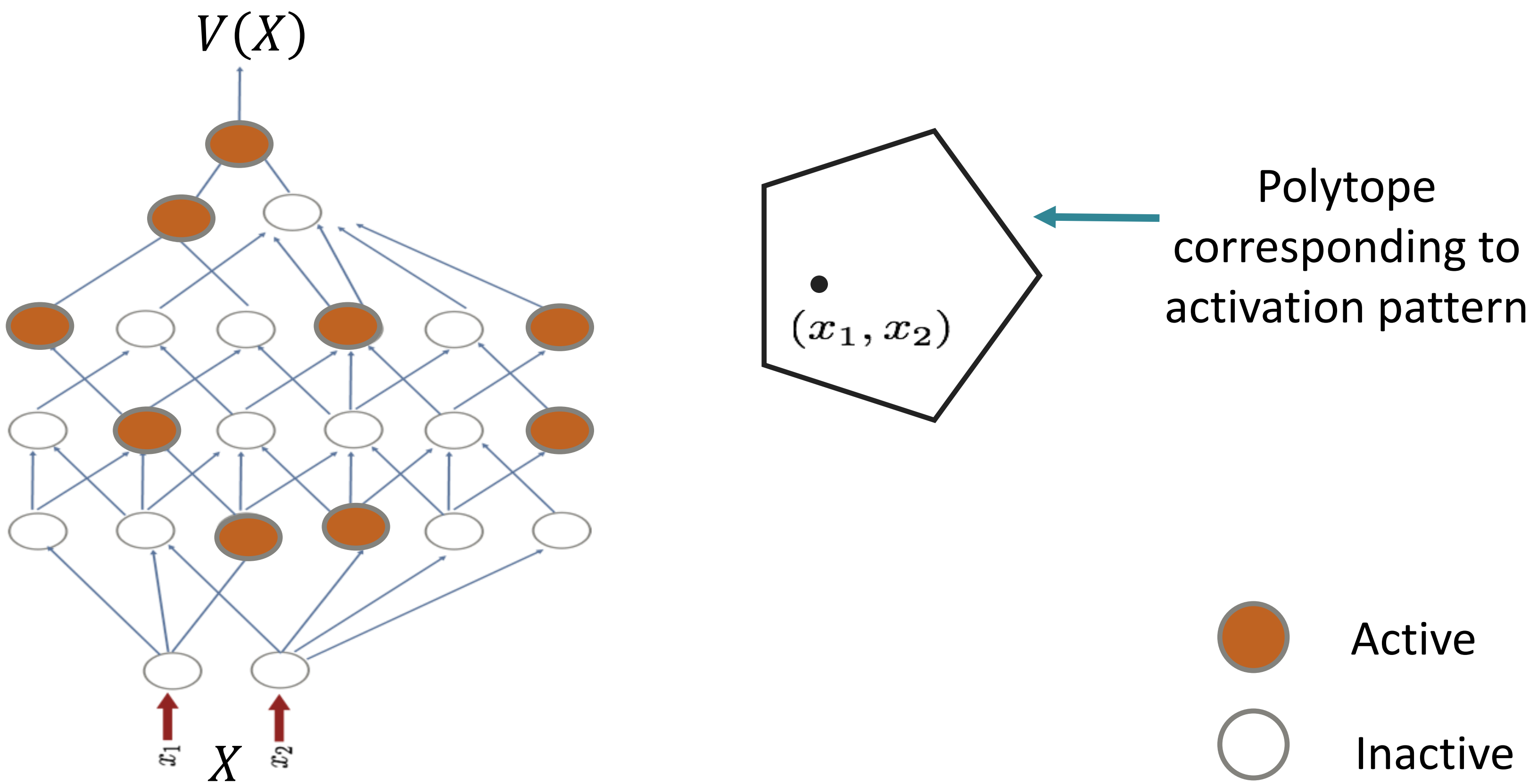
Combining local search and MILP



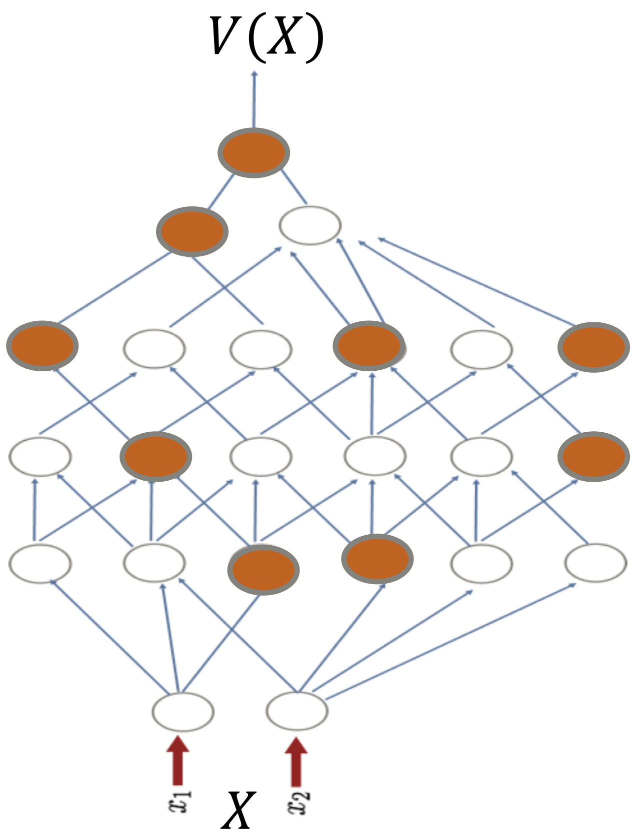


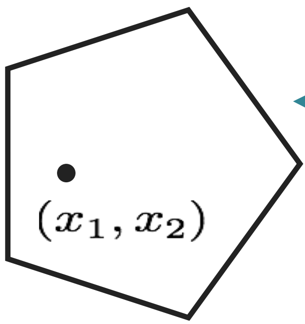


Combining local search and MILP

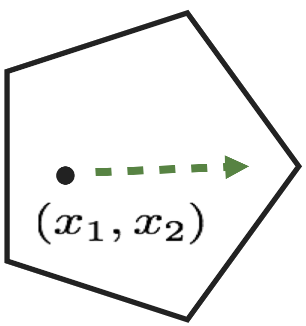


Combining local search and MILP



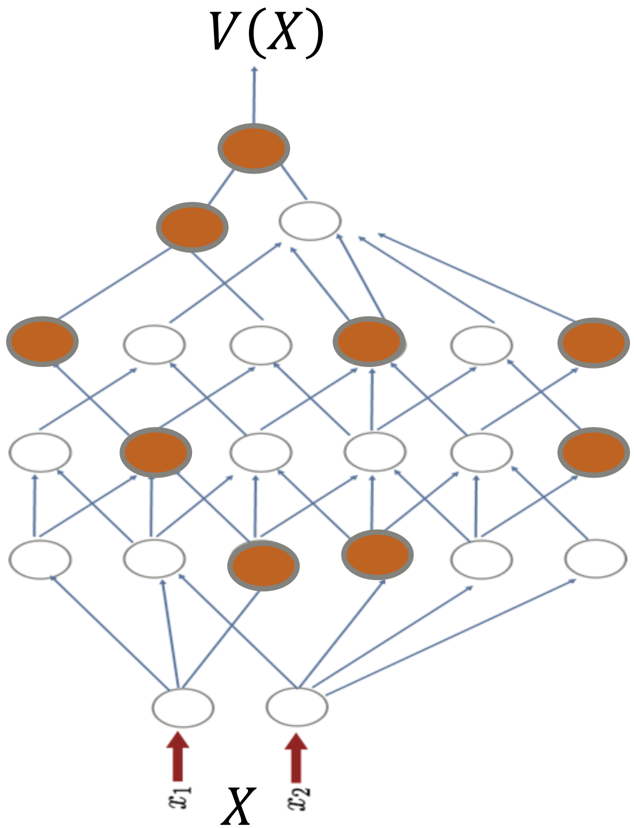


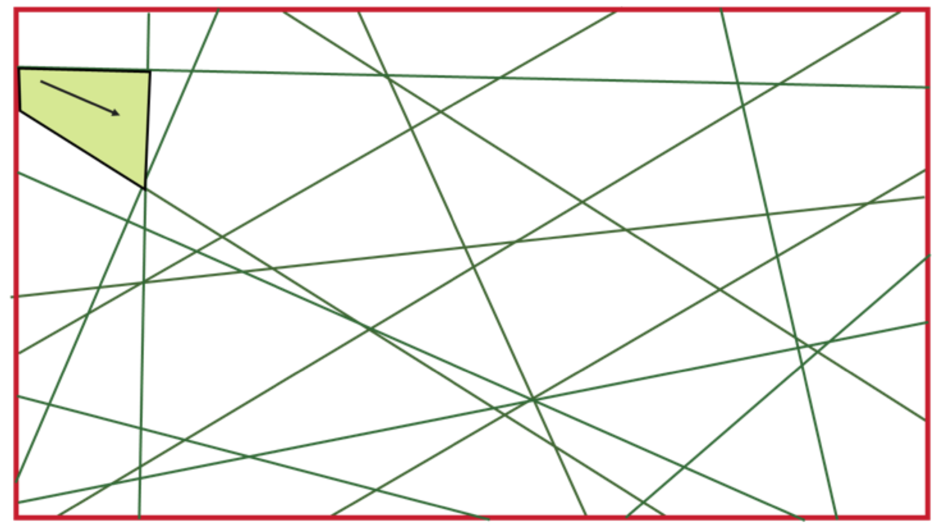
Polytope corresponding to activation pattern



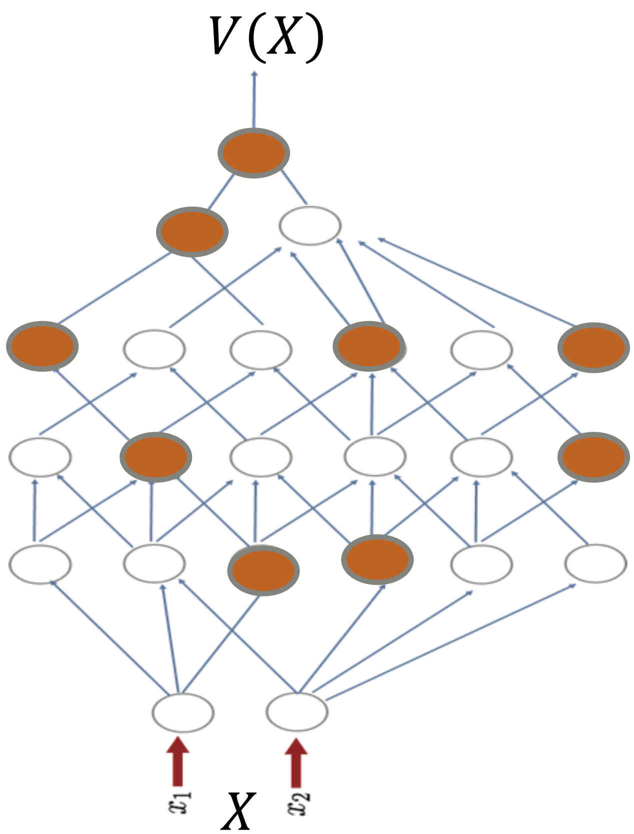
Solve LP to reach extremal corner

Combining local search and MILP

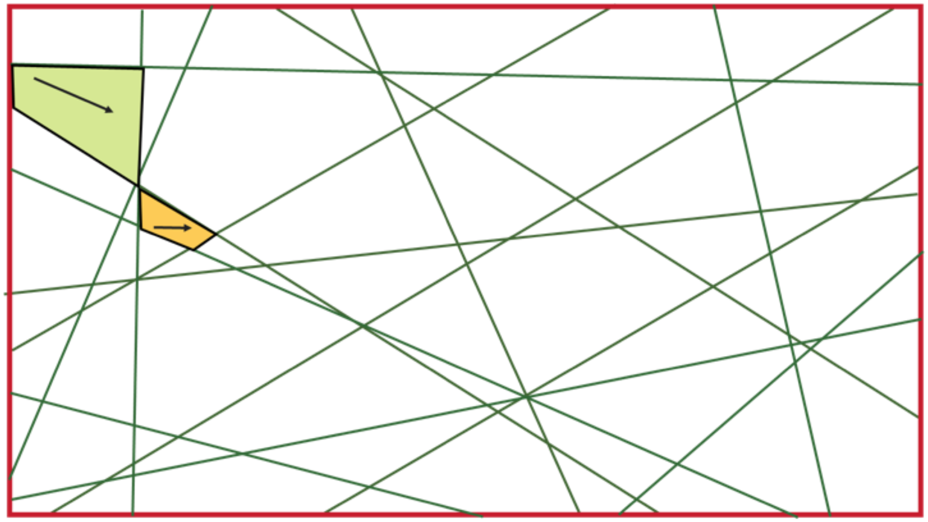




Combining local search and MILP

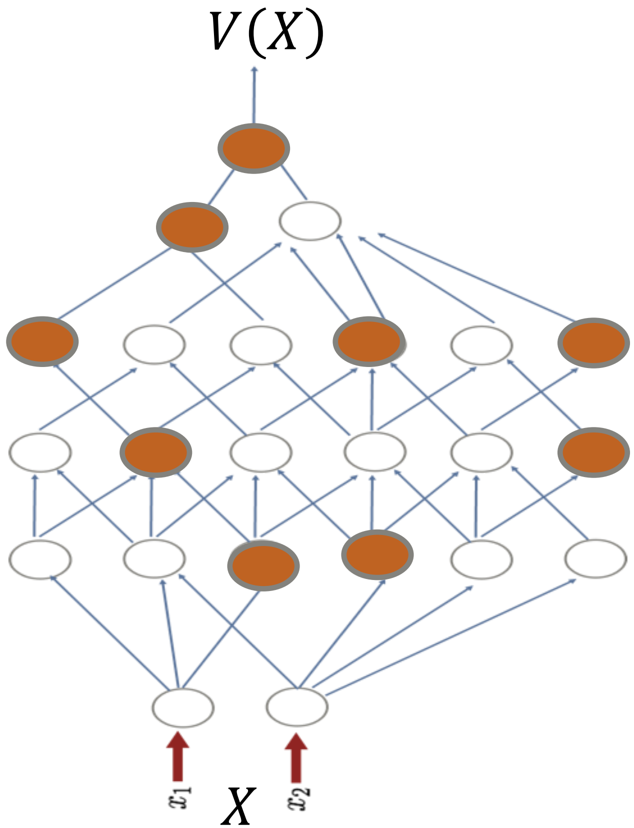


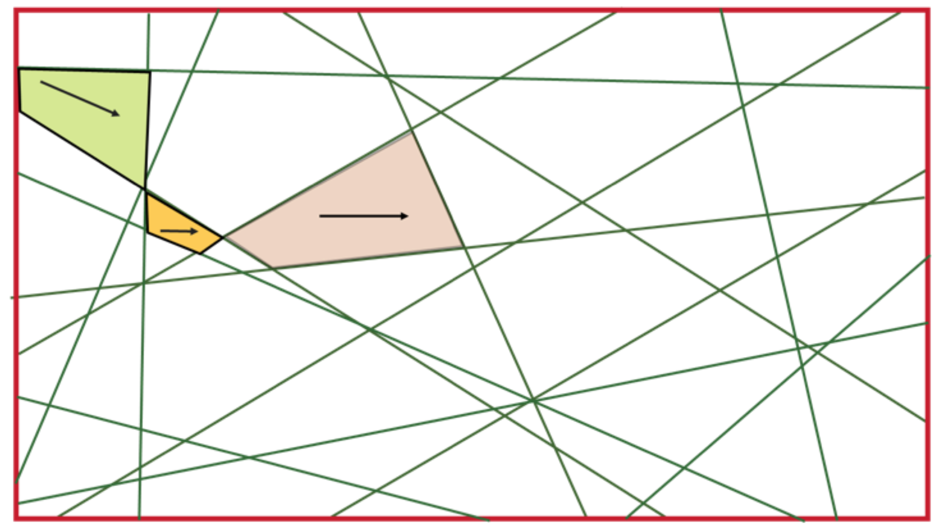
Use gradient to move to next activation pattern



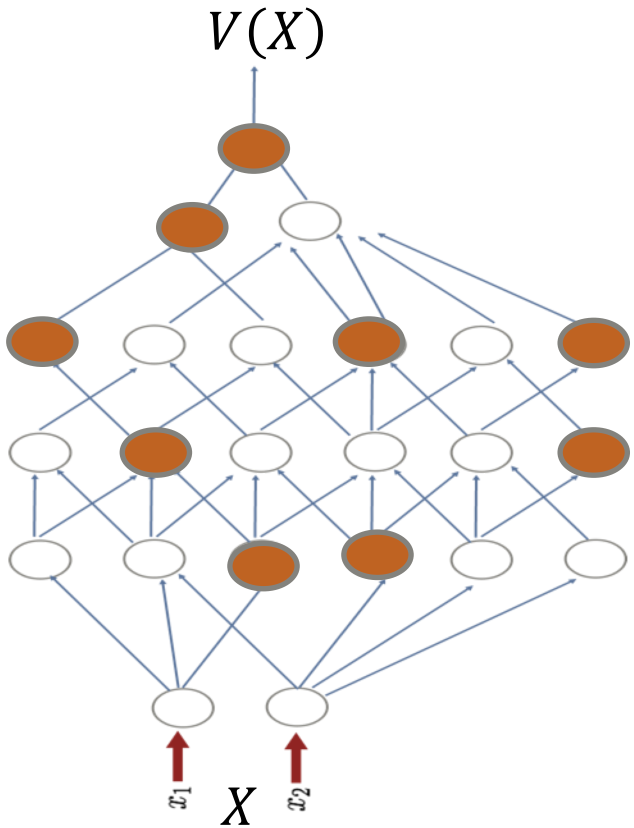
In some cases, gradient based local search works so well, that we skip LP step.

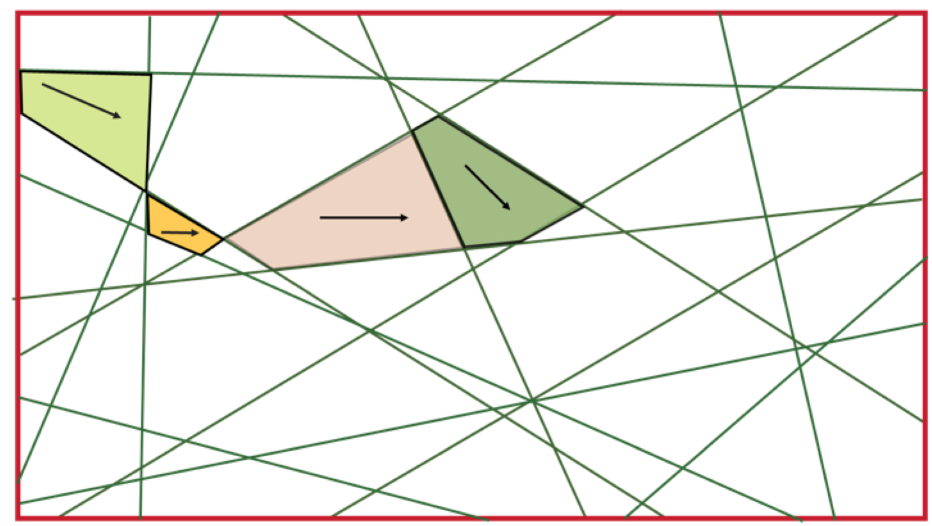
Combining local search and MILP



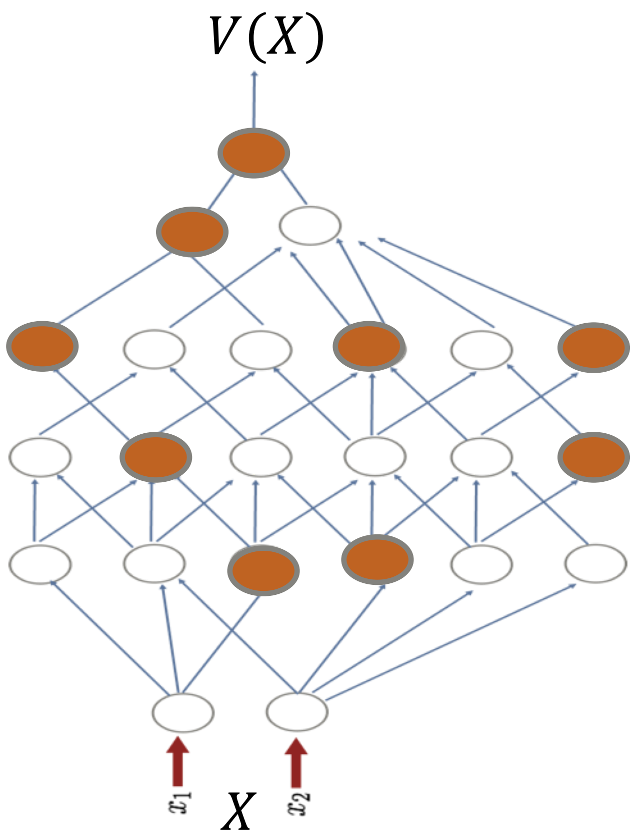


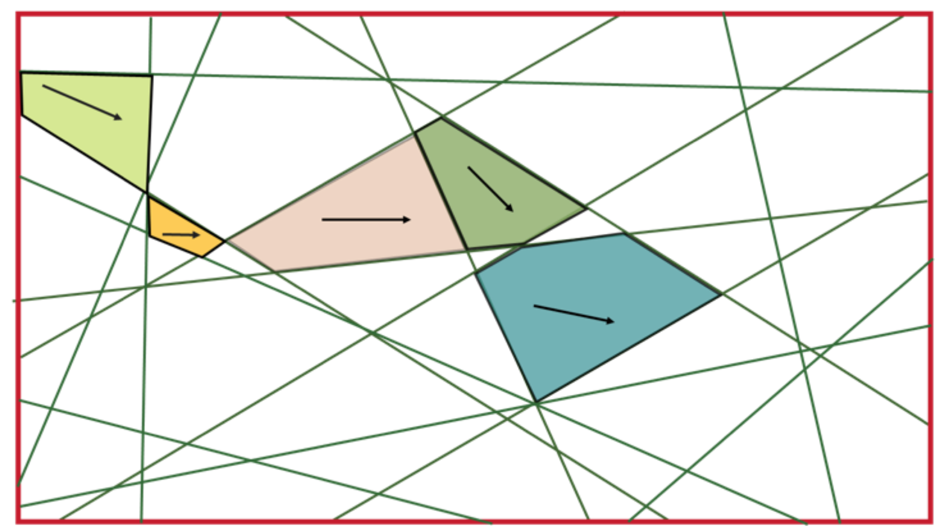
Combining local search and MILP



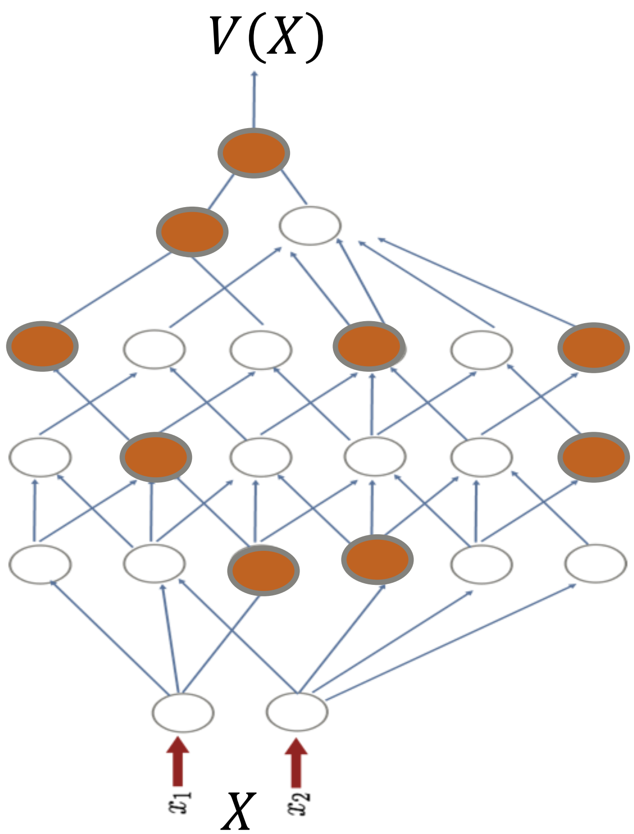


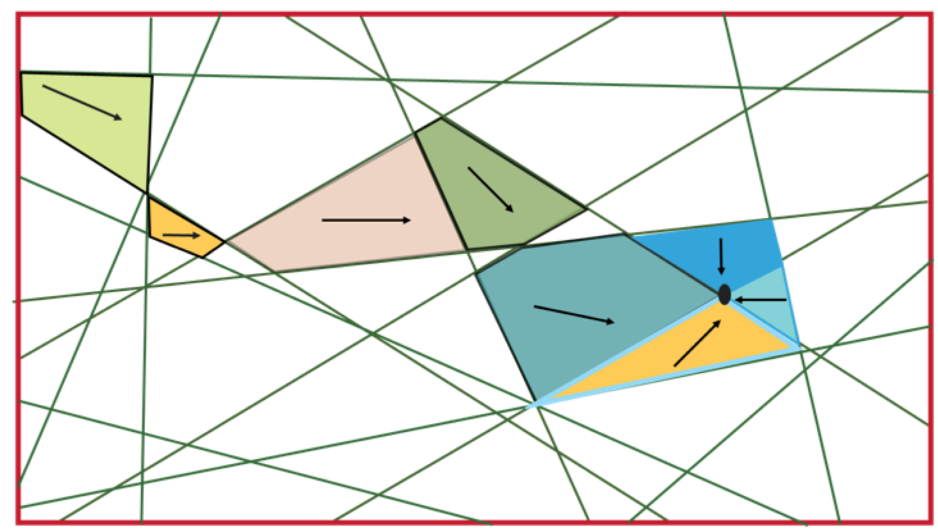
Combining local search and MILP



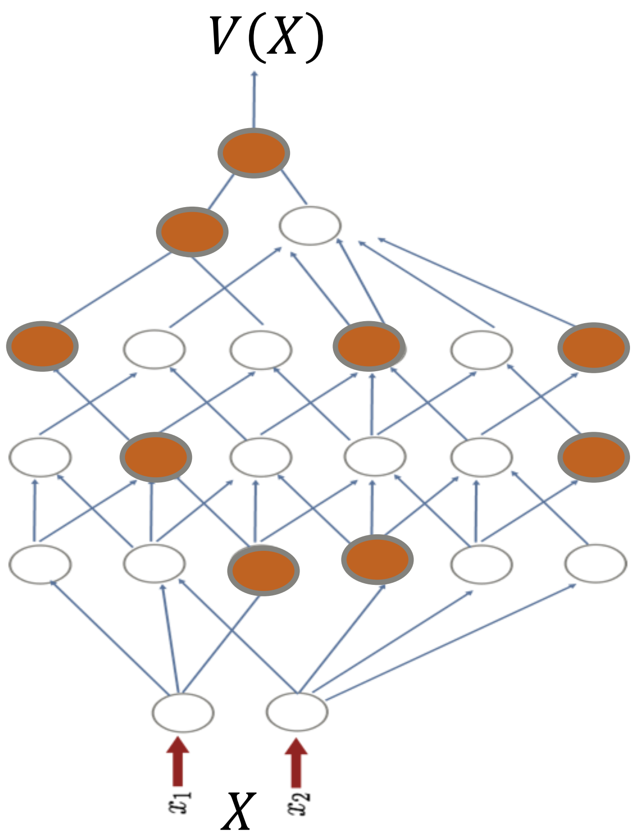


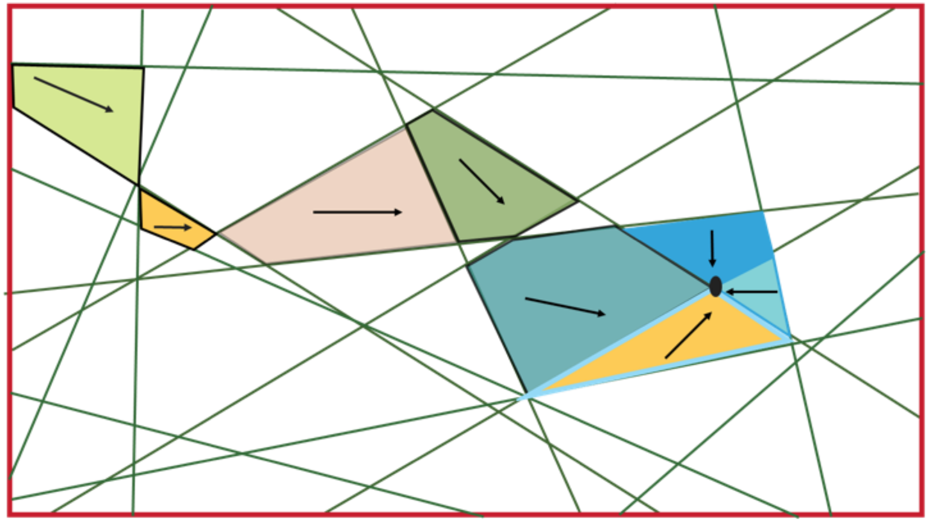
Combining local search and MILP





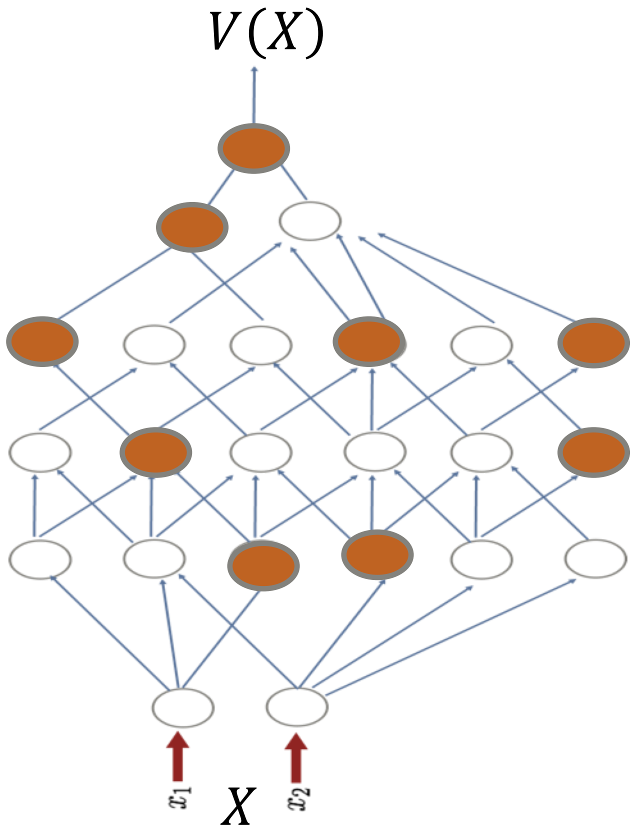
Combining local search and MILP

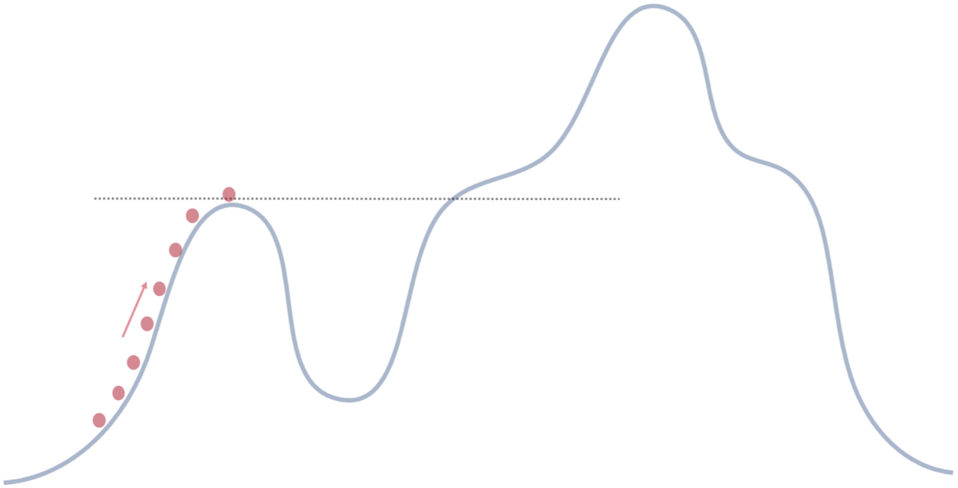




A local optimum

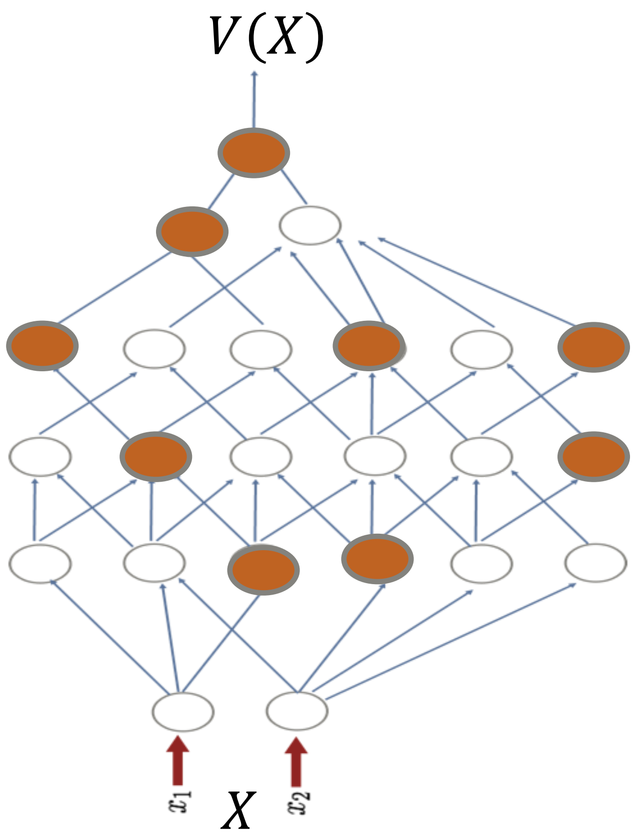
Combining local search and MILP

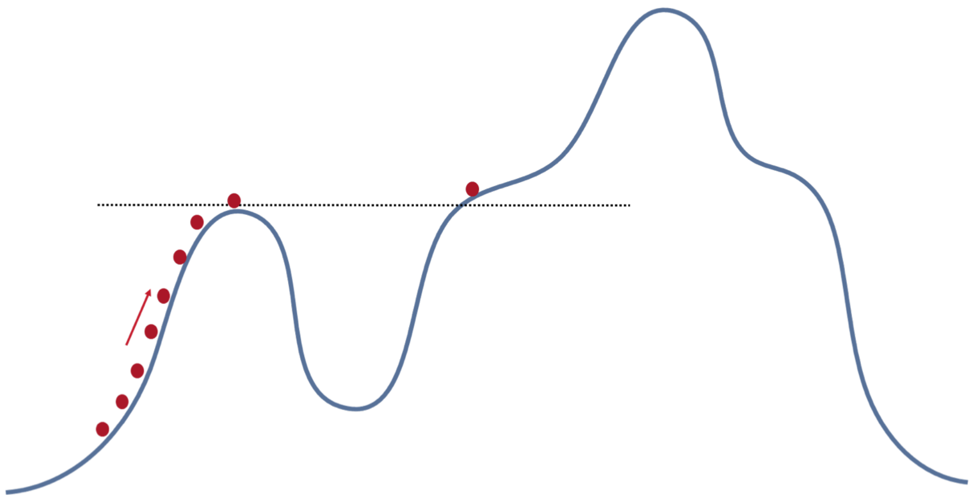




A local optimum

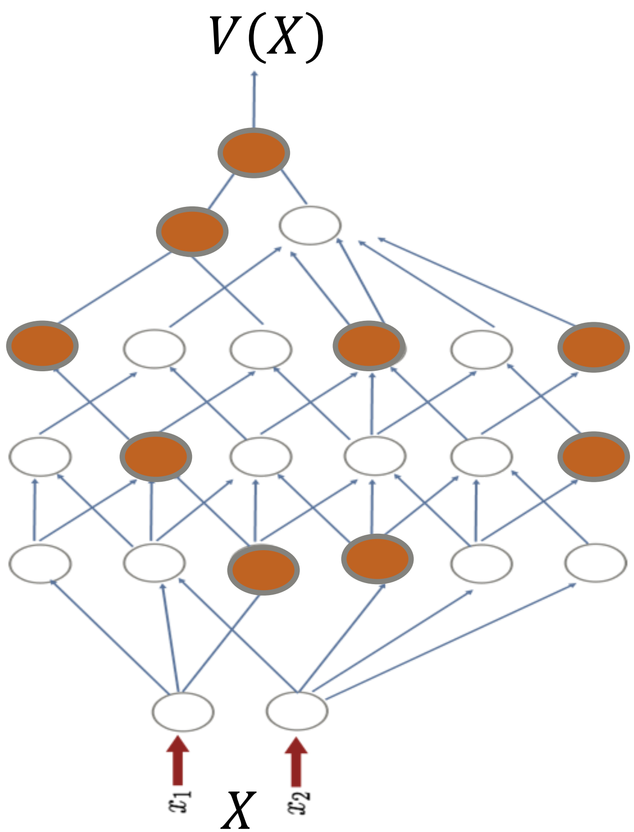
Combining local search and MILP

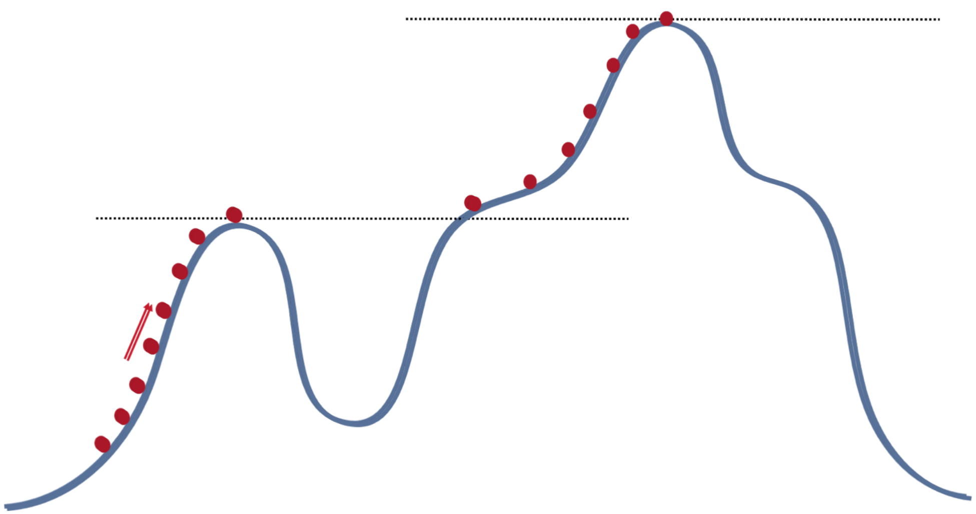




Call to MILP verifier

Combining local search and MILP





Again local search
followed by MILP verifier

Performance compared to Reluplex and MILP

ID	x	k	N	23 cores T	23 cores Nc	23 cores Monolithic T	23 cores Monolithic Nc	single core T	single core Nc	single core Monolithic T	single core Monolithic Nc	Reluplex T
N_0	2	1	100	1s	94	2.3s	24	0.4s	44	0.3s	25	9.0
N_1	2	1	200	2.2s	166	3.6s	29	0.9s	71	0.8s	38	1m50s
N_2	2	1	500	7.8s	961	12.6s	236	2s	138	2.9s	257	15m59s
N_3	2	1	500	1.5s	189	0.5s	43	0.6s	95	0.2s	53	12m25s
N_4	2	1	1000	3m52s	$32E3$	3m52s	$3E3$	1m20s	4.8E3	35.6s	5.3E3	1h06m
N_5	3	7	425	4s	6	6.1s	2	1.7s	2	0.9s	2	DNC
N_6	3	4	762	3m47s	3.3E3	4m41s	3.6E3	37.8s	685	56.4s	2.2E3	DNC
N_7	4	7	731	3.7s	1	7.7s	2	3.9s	1	3.1s	2	1h35m
N_8	3	8	478	6.5s	3	40.8s	2	3.6s	3	3.3s	2	DNC
N_9	3	8	778	18.3s	114	1m11s	2	12.5s	12	4.3s	73	DNC
N_{10}	3	26	2340	50m18s	$4.6E4$	1h26m	$6E4$	17m12s	2.4E4	18m58s	1.9E4	DNC
N_{11}	3	9	1527	5m44s	450	55m12s	6.4E3	56.4s	483	130.7s	560	DNC
N_{12}	3	14	2292	24m17s	1.8E3	3h46m	2.4E4	8m11s	2.3E3	1h01m	1.6E4	DNC
N_{13}	3	19	3057	4h10m	2.2E4	61h08m	6.6E4	1h7m	1.5E4	15h1m	1.5E5	DNC
N_{14}	3	24	3822	72h39m	8.4E4	111h35m	1.1E5	5h57m	3E4	timeout	-	DNC
N_{15}	3	127	6845	2m51s	1	timeout	-	3m27s	1	timeout	-	DNC

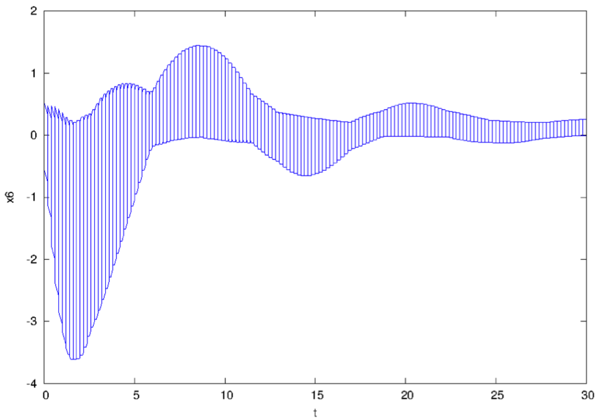
Sherlock: A Tool for Verification of Deep Neural Networks. Dutta et. al. AAAI Spring Symposium on Verification of Neural Networks, 2019.

Closed-loop validation with NN controllers

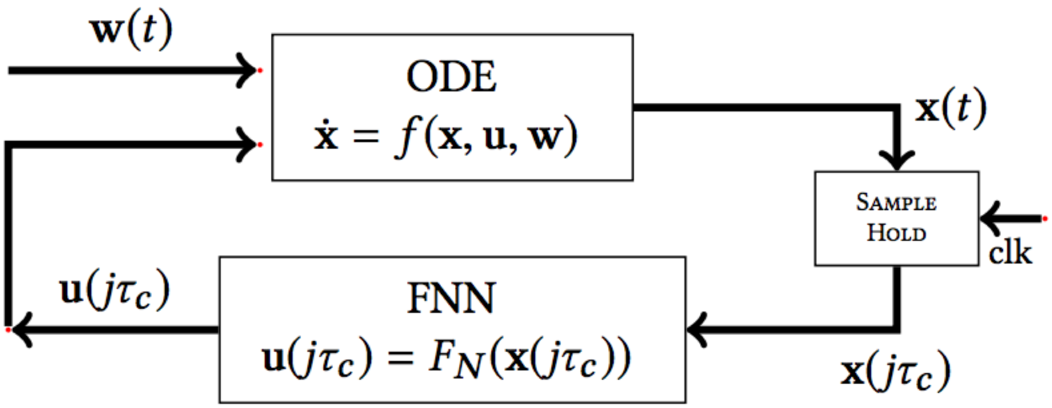
Key idea: Combine neural network range estimation with reachable set computation for dynamical systems. Dovetail between

- Estimate range of control input
- Estimate range of next state (accelerate by taking multiple steps, more approximate)

$\Diamond\Box T$ Specification: Stability



Xin Chen, Sriram Sankaranarayanan, and Erika Abraham. FLOW* 1.2: More Effective to Play with Hybrid Systems.



Learning and Verification of Feedback Control Systems using Feedforward Neural Networks. Souradeep Dutta, Susmit Jha, Sriram Sankaranarayanan, Ashish Tiwari. IFAC Conference on Analysis and Design of Hybrid Systems, 2018

Sherlock - A Tool for Verification of Neural Network Feedback Systems: Demo Abstract. (Best Demo Award) . Dutta et. al. 22nd ACM International Conference on Hybrid Systems: Computation and Control (HSCC), 2019

Closed-loop validation with NN controllers

Key idea: Combine neural network range estimation with reachable set computation for dynamical systems. Dovetail between

- Estimate range of control input
- Estimate range of next state (accelerate by taking multiple steps, more approximate)

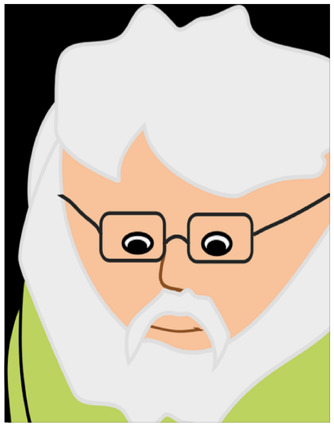
Specification: Stability $\Diamond\Box T$

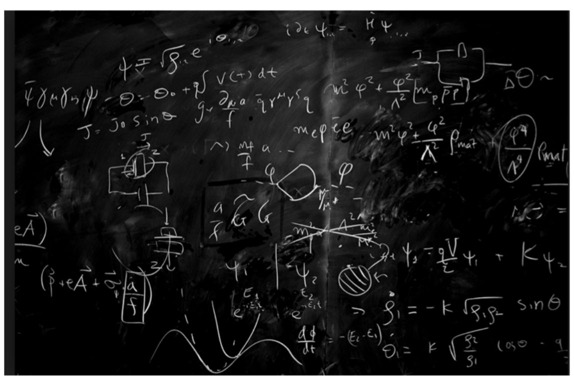
ID	NN Layer Sizes	*Acc*	$Reach_T$	Acc_T	Inv_T
1	2, 52, 3, 4, 3, 4, 3, 200, 2	1	7.53s	2	2.8s
2	2, 102, 52, 3, 4, 3, 4, 3, 250, 2	2	2m25s	2	1m3s
3	3, 103, 53, 4, 5, 4, 5, 4, 600, 3	2	2m33s	5	3m10s
4	3, 103, 53, 4, 5, 4, 5, 4, 300, 3	1	48s	3	17.89s
5	3, 103, 4, 5, 4, 5, 4, 300, 3	5	63m6.4s	16	111m45s
6	3, 303, 203, 4, 252, 3	2	16m25s	4	9m19s
7	4, 104, 5, 6, 5, 6, 5, 600, 4	3	19m42s	8	22m1s

Learning and Verification of Feedback Control Systems using Feedforward Neural Networks. Souradeep Dutta, Susmit Jha, Sriram Sankaranarayanan, Ashish Tiwari. IFAC Conference on Analysis and Design of Hybrid Systems, 2018

Sherlock - A Tool for Verification of Neural Network Feedback Systems: Demo Abstract. Dutta et. al. 22nd ACM International Conference on Hybrid Systems: Computation and Control (HSCC), 2019

Where do we get specifications from?





Extracting Safety Property from Data: Mining Safe Driving Patterns

Safe Driving is more than adherence to traffic rules.

If we observe how 'safe' human drivers drive, can we transfer these habits/patterns to an autonomous car?

220GB of driving data: Instrumented car (2016 Lincoln MKZ) driving along El Camino Real (San Francisco Bay Area). A mixture of turns and straight driving.

timestamp,angle,torque,speed,throttle,brake

```
1479424215873184606,0.00035957866912,0.375,23.0033496401
1479424215877284689,0.000717059237998,0.375,23.0039188398
1479424215880976321,0.00103893906341,0.375,23.0044313542
1479424215922817911,0.00349065847695,0.394737094117,23.0073100866
1479424215927281227,0.00349065847695,0.380823288686,23.0060736059
1479424215930775951,0.00349065847695,0.369925094231,23.0060070992
1479424215972815517,0.00349065847695,0.36386123579,23.0083332062
1479424215977378050,0.00349065847695,0.349617492104,23.0083332062
1479424215980916687,0.00349065847695,0.33856983926,23.0083332062
1479424216022820692,0.00466956372537,0.479716434815,23.0064573783
1479424216027309701,0.00506125153205,0.49374272130 4,23.0058341399
1479424216030737492,0.00523598771542,0.491111238869,23.0051610113
1479424216073039735,0.00555371235876,0.522755351565,22.996972343
1479424216077185305,0.00591556852827,0.548671390895,22.9998519908
1479424216080827865,0.00623351662237,0.571442745942,23.0023822186
1479424216122856036,0.012229296689,0.708619621716,23.0157470729
1479424216127297778,0.0133895580247,0.73631882096,23.0163624604
1479424216130778849,0.0142974665448,0.742006460967,23.015955927
1479424216172794699,0.0217718389734,0.677617815283,23.0115497083
1479424216177269210,0.0229428882819,0.663639446207,23.0121712441
1479424216180687241,0.0238374389065,0.652961536493,23.0126460279
1479424216222706305,0.0319459443015,0.627780242013,23.0204110546
1479424216227198494,0.0327277701736,0.585784626578,23.019788622
1479424216230735818,0.0331612564623,0.5625,23.0190077045
1479424216272749590,0.0328959117777,0.399492153054,23.0147337254
1479424216277232525,0.0325053001336,0.343541126177,23.015976778
1479424216280736012,0.0322000511415,0.29981740514,23.016948179
1479424216322795445,0.0302828893188,0.20942605674 3,23.0333328247
1479424216327284030,0.0298919886907,0.195427966481,23.0333328247
1479424216330780633,0.029587548536,0.1875,23.0335972054
1479424216372784939,0.0279252678156,0.19647137980 4,23.0464384649
```

How does acceleration and speed change during initiation, continuation and termination of a turn for a safe driver?

Temporal Logic

- Temporal logics specify patterns that timed behaviors of systems may or may not satisfy.

- Linear Temporal Logic (LTL) specify property of discrete sequences of states.
  - Based on logic operators (¬, ∧, ∨), and
  - temporal operators: “next”, “always” (G), “eventually” (F) and “until” (U)

- Extension of LTL with continuous time and real-valued signals
  - Reasoning about continuous signals: steering angle of a car

LTL : G (torque applied → F (turn complete))

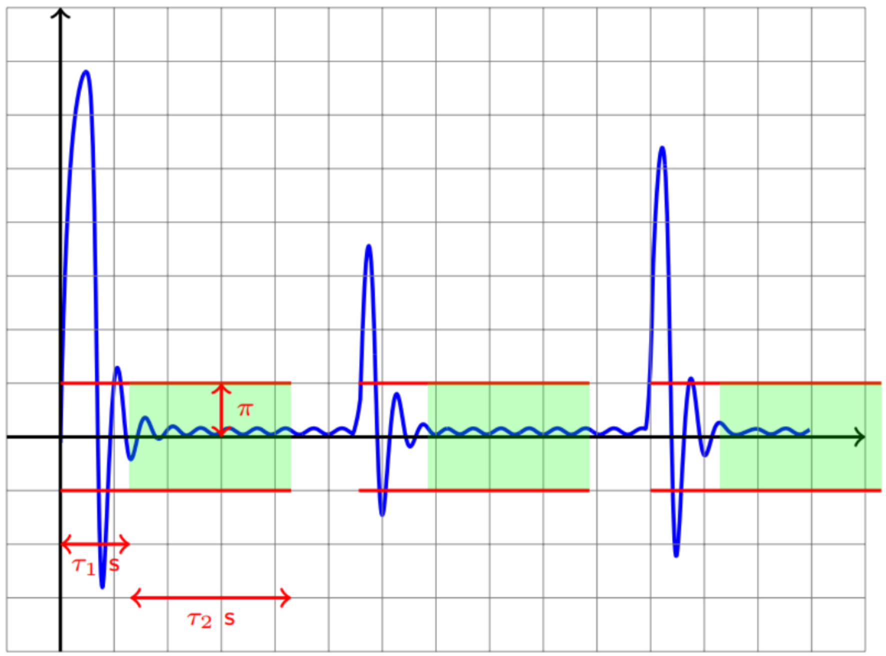
MTL : G (torque applied → $F_{[0,10]}$ (turn complete)) [real time]

STL : G (torque ≥ 0 → $F_{[0,10]}$ (turn angle ≥ 90)) [real valued + real time]

Learning Signal Temporal Logic

$$\varphi := \mathsf{G}\left(x[t] > \pi \rightarrow \mathsf{F}_{[0,\tau_1]}\ \left(\mathsf{G}_{[0,\tau_2]}\ x[t] < \pi\right)\right)$$

- Valuation 1: $\pi \leftarrow 1.5$, $\tau_1 \leftarrow 1\ s$, $\tau_2 \leftarrow 1.15\ s$
- Valuation 2 (tight): $\pi \leftarrow .5$, $\tau_1 \leftarrow 0.65\ s$, $\tau_2 \leftarrow 2\ s$



<u>Challenge</u>

Multiple possible values for the same parameter.
<u>Select tightest parameter!</u>

Given a set of traces, learn parameter values for the template STL formula that is consistent with all the examples.

Learning Using Tightness Metric

Constrained Multiobjective Optimization Problem

$$\texttt{minimize}\ \{|\epsilon_1|, |\epsilon_2|, \ldots, |\epsilon_k|\}\ \ s.t.$$
$$\epsilon_1 = p_1 - p'_1, \epsilon_2 = p_2 - p'_2, \ldots, \epsilon_k = p_k - p'_k$$
$$\forall \tau \in \mathcal{T}\ \tau \models \phi(p_1, p_2, \ldots, p_k),\ \ \exists \tau' \in \mathcal{T}\ \tau' \not\models \phi(p'_1, p'_2, \ldots, p'_k)$$

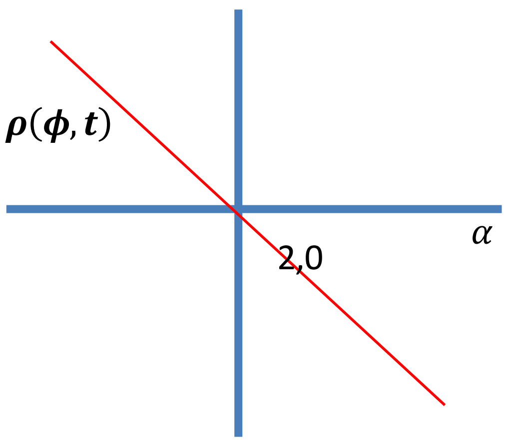
Satisfaction of STL

Qualitative			Robustness
$(\mathbf{x}, t) \models \mu$	$\Leftrightarrow$	$f(x_1[t], \ldots, x_n[t]) > 0$	$f(x_1[t], \ldots, x_n[t]) > 0$
$(\mathbf{x}, t) \models \varphi \wedge \psi$	$\Leftrightarrow$	$(x, t) \models \varphi \wedge (x, t) \models \psi$	$-\chi^{\varphi}(x, t)$
$(\mathbf{x}, t) \models \neg\varphi$	$\Leftrightarrow$	$\neg((x, t) \models \varphi)$	$\min(\chi^{\varphi_1}(x, t), \chi^{\varphi_2}(w, t))$
$(\mathbf{x}, t) \models \varphi\, \mathcal{U}_{[a,b]}\, \psi$	$\Leftrightarrow$	$\exists t' \in [t + a, t + b]$ such that $(x, t') \models \psi \wedge$ $\forall t'' \in [t, t'],\ (x, t'') \models \varphi\}$	$\max\limits_{\tau \in t+[a,b]}(\min(\chi^{\varphi_2}(x, \tau), \min\limits_{s \in [t,\tau]} \chi^{\varphi_1}(x, s))$

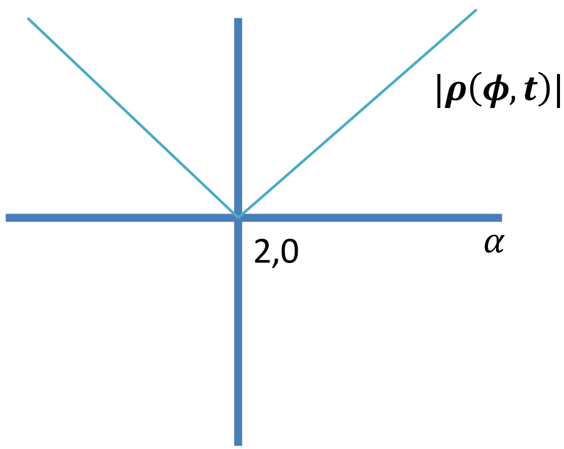
Learning STL

Formula to learn $\boldsymbol{\phi}$: $F_{[0,0.5]}\,(x \geq \alpha)$ from set of traces example T
Let us assume that $\alpha = 2$ is the tightest parameter for T

Robustness metric



Absolute value of robustness metric



Find
α *that minimizes* $|\boldsymbol{\rho(\phi(\alpha), t)}|$

Problems:
- Non-differential close to optimum
- Could learn false property even when close to optimum

Learning STL with Tightness Metric

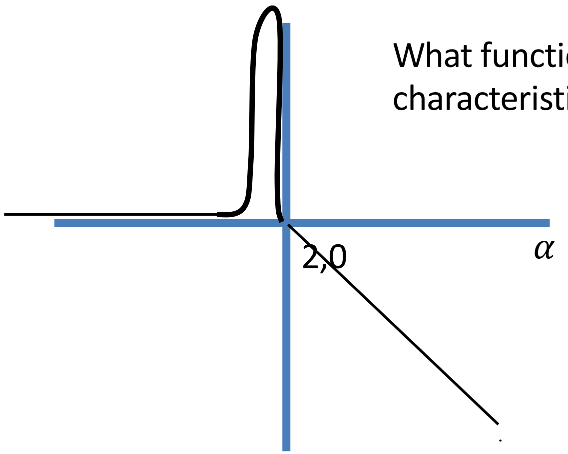
TeLEx: Passive STL Learning Using Only Positive Examples. Susmit Jha, Ashish Tiwari, Sanjit A. Seshia, Natarajan Shankar, and Tuhin Sahai. 17th International Conference on Runtime Verification (RV), 2017

Formula to learn $\boldsymbol{\phi}$: $F_{[0,0.5]}\,(x \geq \alpha)$ from set of traces example T

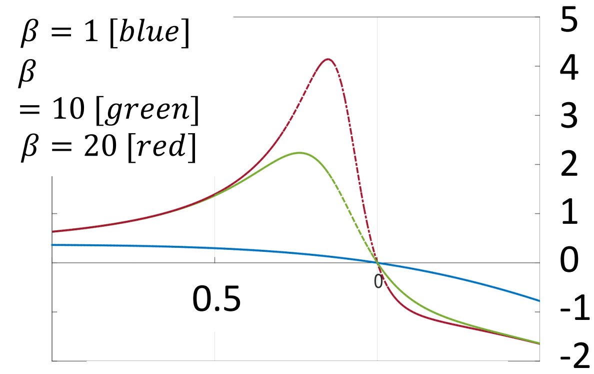
Let us assume that $\alpha = 2$ is the tightest parameter for T

$$\boxed{\frac{1}{r + e^{-\beta r}} - e^{-r}}$$

Practical Metric (Correctness + differentiability for optimization)



What function $\boldsymbol{\theta}$ would have this characteristic ?



<u>Tightness Metric:</u> *The tightness metric* $\theta : \mathcal{F} \times \mathcal{T} \times ST \mapsto \mathbb{R} \cup \{-\infty, \infty\}$ *maps an STL formula* $\phi \in \mathcal{F}$*, a trace* $\tau \in \mathcal{T}$*, and a sampled time instance* $t \in ST$ *to a real value s.t.:*

- $\theta(\top, \tau, t) = \infty, \quad \theta(\bot, \tau, t) = -\infty$
- $\theta(\mu, \tau, t) = \mathtt{P}(g(\tau(t)) - \alpha)$ *where* $\mu(\mathbf{x}) := (g(\mathbf{x}) \geq \alpha)$
- $\theta(\phi_1 \wedge \phi_2, \tau, t) = \min(\theta(\phi_1, \tau, t), \theta(\phi_2, \tau, t))$
- $\theta(\phi_1 \vee \phi_2, \tau, t) = \max(\theta(\phi_1, \tau, t), \theta(\phi_2, \tau, t))$
- $\theta(\mathbf{F}_{[t_1,t_2]}\phi, \tau, t) = \mathtt{C}(\gamma, t_1, t_2) \sup_{t' \in [t+t_1, t+t_2]} \theta(\phi, \tau, t')$
- $\theta(\mathbf{G}_{[t_1,t_2]}\phi, \tau, t) = \mathtt{E}(\gamma, t_1, t_2) \inf_{t' \in [t+t_1, t+t_2)} \theta(\phi, \tau, t')$
- $\theta(\phi_1 \mathbf{U}_{[t_1,t_2]}\phi_2, \tau, t) = \mathtt{E}(\gamma, t_1, t_2) \sup_{t' \in [t+t_1, t+t_2]} (\min(\theta(\phi_2, \tau, t'), \inf_{t'' \in [t,t')} \theta(\phi_1, \tau, t'')))$

Learning Using Tightness Metric

Constrained Multiobjective Optimization Problem

$$\texttt{minimize}\ \{|\epsilon_1|, |\epsilon_2|, \ldots, |\epsilon_k|\}\ \ s.t.$$
$$\epsilon_1 = p_1 - p_1', \epsilon_2 = p_2 - p_2', \ldots, \epsilon_k = p_k - p_k'$$
$$\forall \tau \in \mathcal{T}\ \tau \models \phi(p_1, p_2, \ldots, p_k),\ \ \exists \tau' \in \mathcal{T}\ \tau' \not\models \phi(p_1', p_2', \ldots, p_k')$$

Unconstrained Scalar Optimization Problem

$$(v_1^*, v_2^*, \ldots, v_k^*) = \arg \max_{p_1, p_2, \ldots, p_k}\ [\min_{\tau \in \mathcal{T}}\ \theta(\phi(p_1, p_2, \ldots, p_k), \tau, 0)]$$

Example Results

Safe Driving is more than adherence to traffic rules.

220GB of driving data: Instrumented car (2016 Lincoln MKZ) driving along El Camino Real (San Francisco Bay Area). A mixture of turns and straight driving.

The speed of the car must be below some upper bound $a \in [15, 25]$ if the angle is larger than 0.2 or below -0.2. Intuitively, this property captures required slowing down of the car when making a significant turn.

Template STL:	$G[0, 2.2e11](((angle \geq 0.2) \vert (angle \leq -0.2)) \Rightarrow (speed \leq a?15; 25))$
Synthesized STL:	$G[0.0, 2.2e11](((angle \geq 0.2) \vert (angle \leq -0.2)) \Rightarrow (speed \leq 22.01))$
Performance:	Tightness Metric $= 0.067$, Robustness Metric $= 0.004$
Runtime:	8.64 seconds

Example Results

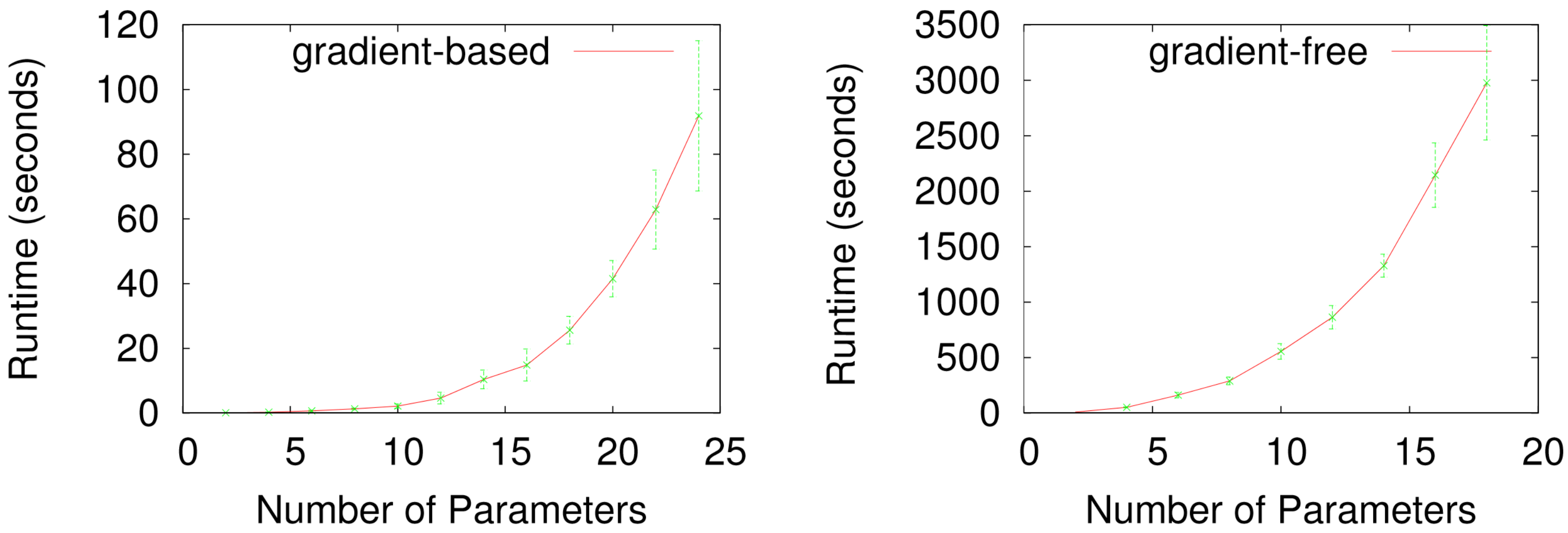
Safe Driving is more than adherence to traffic rules.

220GB of driving data: Instrumented car (2016 Lincoln MKZ) driving along El Camino Real (San Francisco Bay Area). A mixture of turns and straight driving.

Another property of interest is to ensure that when the turn angle is high (say, above 0.06), the magnitude of negative torque applied is below a threshold. This avoids unsafe driving behavior of making late sharp compensation torques to avoid wide turns.

Template STL:	$G[0, 2.2e11]((angle \geq 0.06) \Rightarrow (torque \geq b? - 2; -0.5))$
Synthesized STL:	$G[0.0, 2.2e11]((angle \geq 0.06) \Rightarrow (torque \geq -1.06))$
Performance:	Tightness Metric $= 0.113$, Robustness Metric $= 0.003$
Runtime:	7.30 seconds

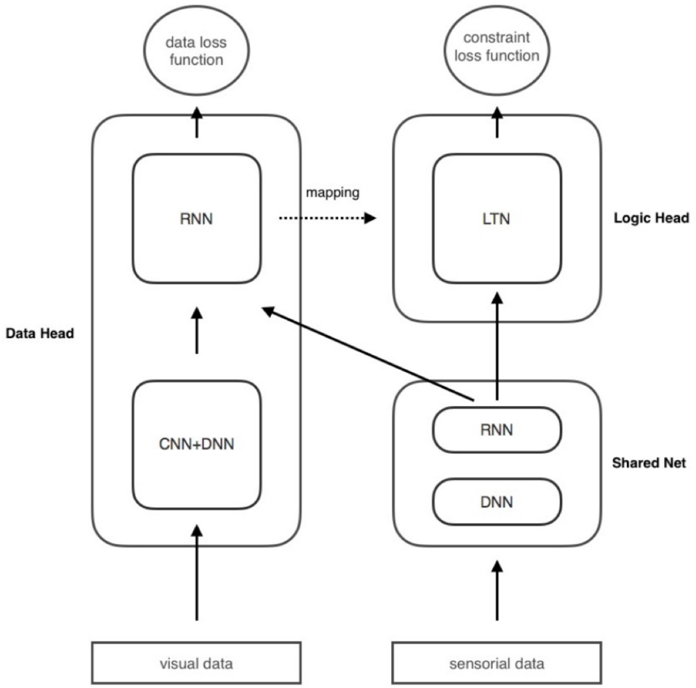
Impact of Smoothness of θ



TeLEx: Passive STL Learning Using Only Positive Examples.
Susmit Jha, Ashish Tiwari, Sanjit A. Seshia, Natarajan Shankar, and Tuhin Sahai.
17th International Conference on Runtime Verification (RV), 2017
https://github.com/susmitjha/TeLEX

Bombara, Giuseppe, Cristian-Ioan Vasile, Francisco Penedo, Hirotoshi Yasuoka, and Calin Belta. "A decision tree approach to data classification using signal temporal logic."
In Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control, pp. 1-10. ACM, 2016.

Application to Safe Autonomous Control

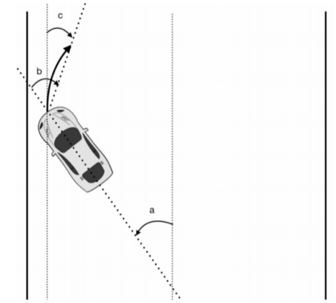


Proximal gradient-descent

$$\min_{w} \ell(o, o') + \lambda g(o') \quad \text{s.t. } o' = f(w, i)$$

$$w_{t+1/2} = w_t - \eta_t \partial \ell(w_t)$$

$$w_{t+1} = \arg\min_{w} \left(\frac{1}{2} \left\| w - w_{t+1/2} \right\|^2 + \eta_t \lambda g(w) \right)$$



Geometric variables in the TORCS

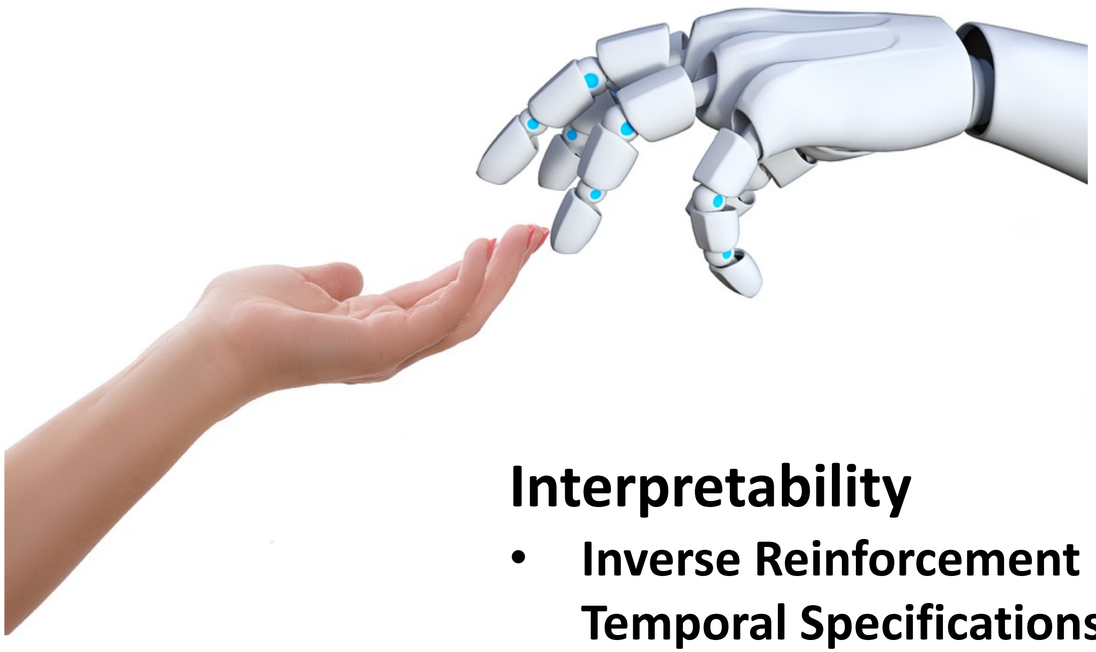
Trusted Neural Networks for Safety-Constrained Autonomous Control.
Shalini Ghosh, Amaury Mercier, Dheeraj Pichapati, Susmit Jha, Vinod Yegneswaran, Patrick Lincoln. SCA/ICML, May, 2018

Verma, A., Murali, V., Singh, R., Kohli, P., & Chaudhuri, S. Programmatically interpretable reinforcement learning. ICML, 2018

Rest of the Talk

Trust

- Global Assume/Guarantee Contracts on DNNs
- Extracting and Integrating Temporal Logic into Learned Control



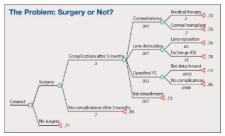
Interpretability

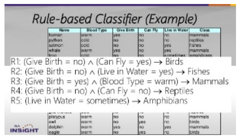
- **Inverse Reinforcement Learning of Temporal Specifications**

Resilience

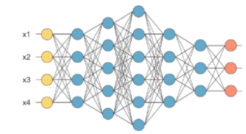
- Adversarial Robustness

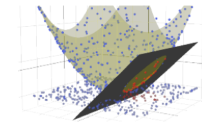
Need for explanation



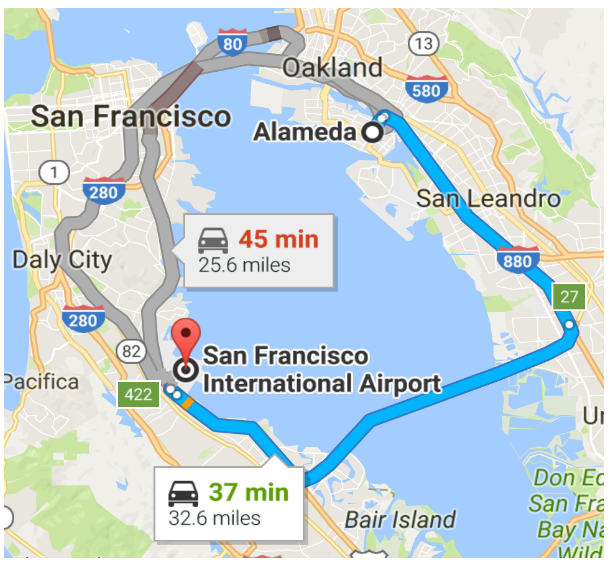


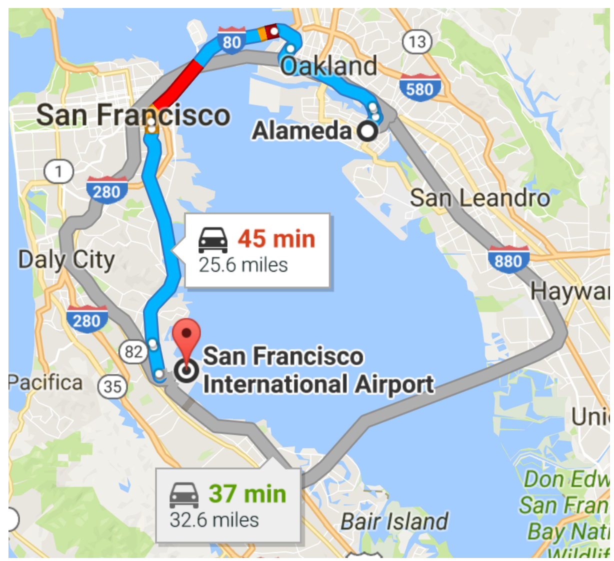
Interpretable but less scalable:
Decision Trees, Linear Regression





Scalable but less interpretable :
Neural Networks, Support Vector Machines



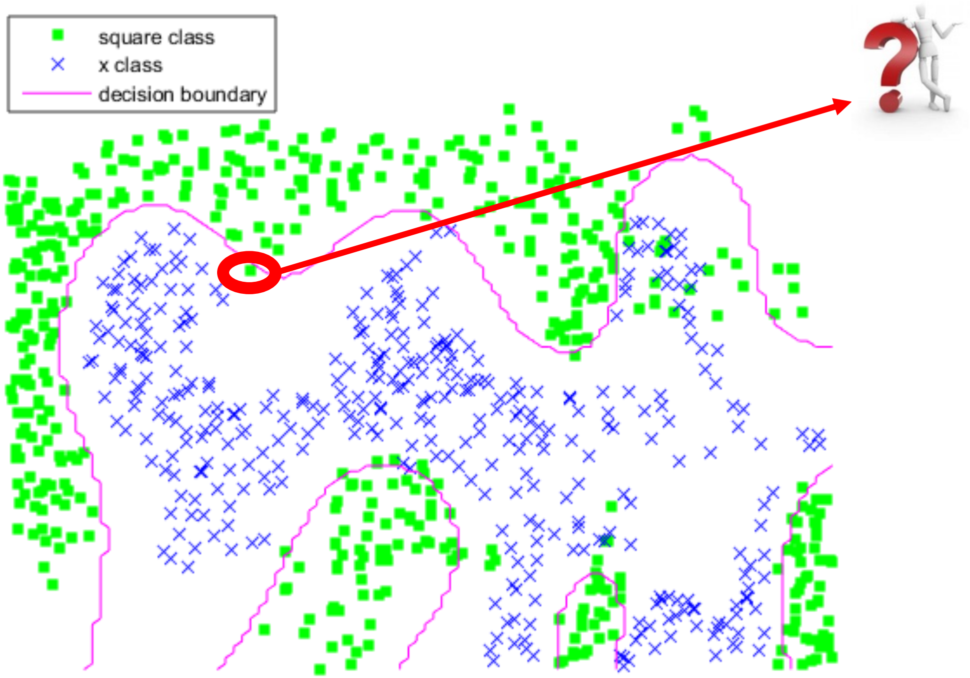


- This route is faster.
- There is traffic on Bay Bridge.
- There is an accident just after Bay Bridge backing up traffic.

Why did we take the San Mateo bridge instead of the Bay Bridge ?

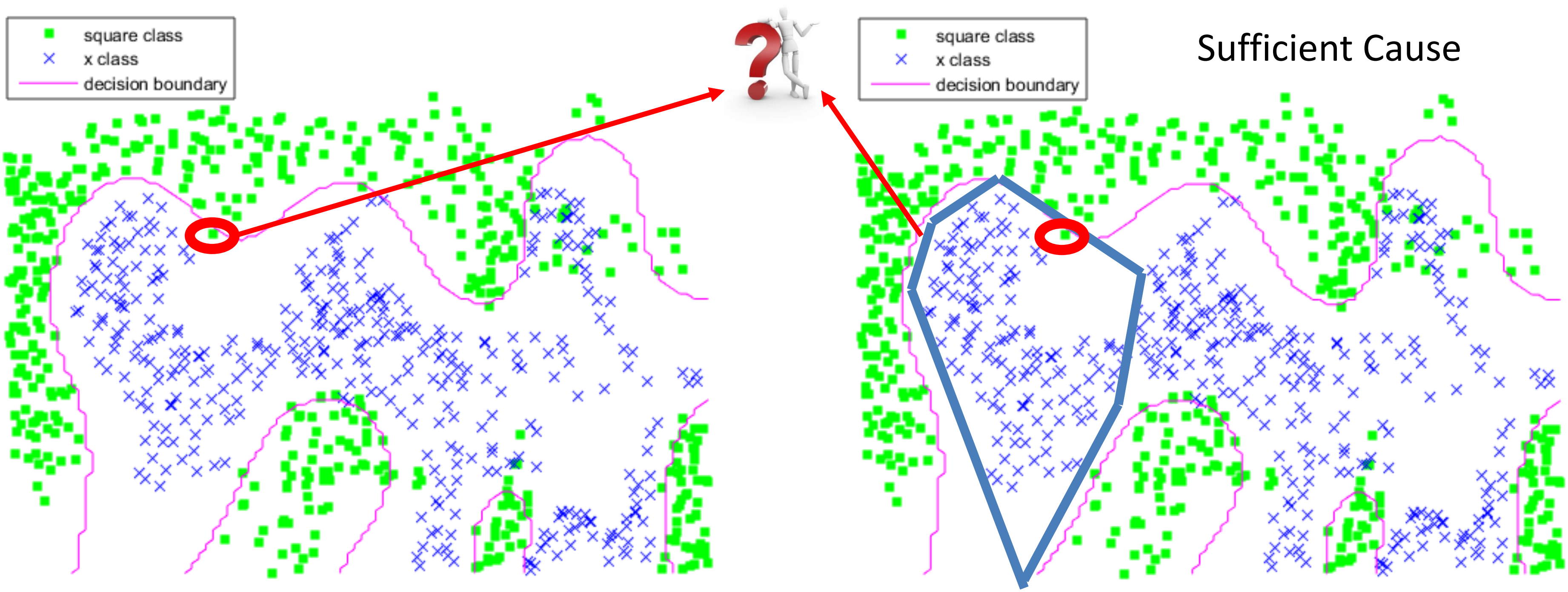
Local Explanations of Complex Models

Not reverse engineering an ML model but finding explanation locally for one decision.



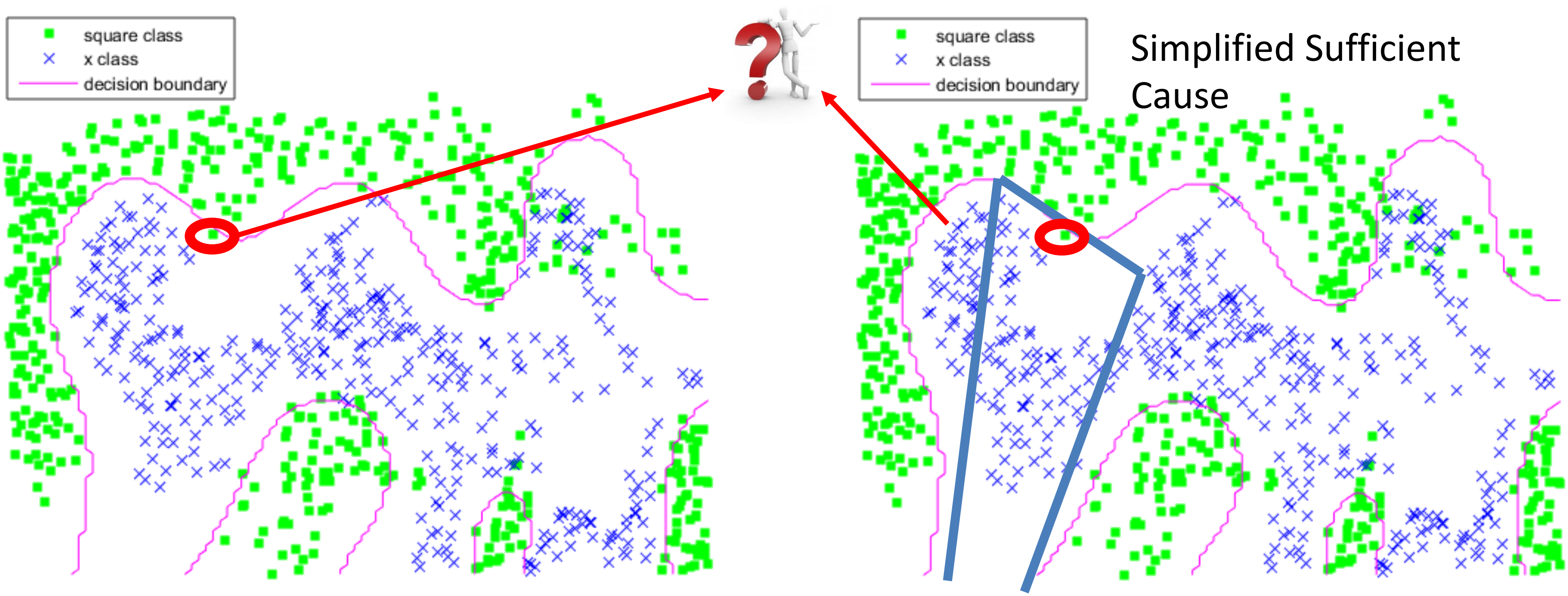
Local Explanations of Complex Models

Not reverse engineering an ML model but finding explanation locally for one decision.



Local Explanations of Complex Models

Not reverse engineering an ML model but finding explanation locally for one decision.



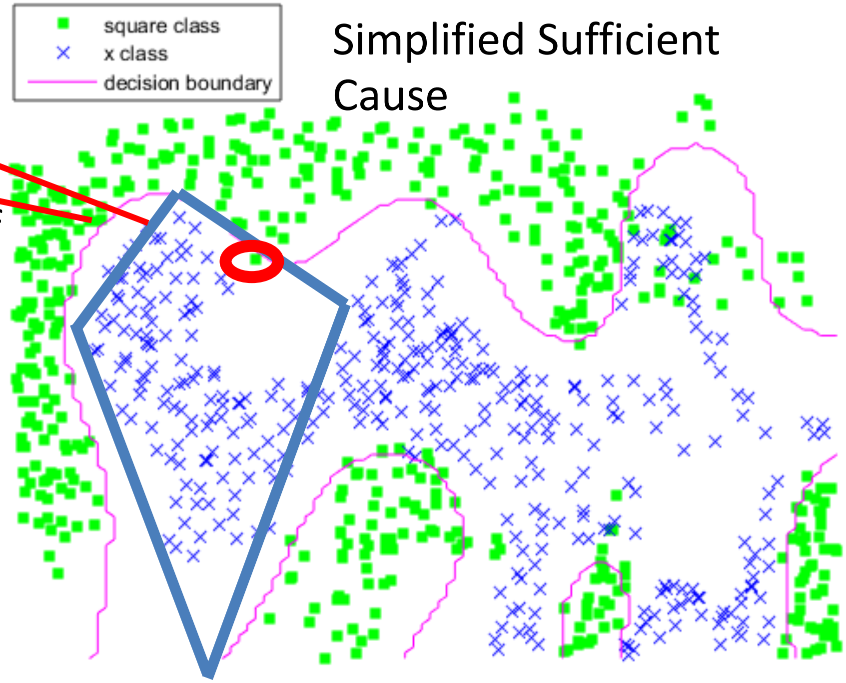
Local Explanations in AI

Not reverse engineering an ML model but finding explanation locally for one decision.

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \quad \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

$\mathcal{L}(f, g, \pi_x)$ Measure of how well g approximates f

$\Omega(g)$ Measure of complexity of g

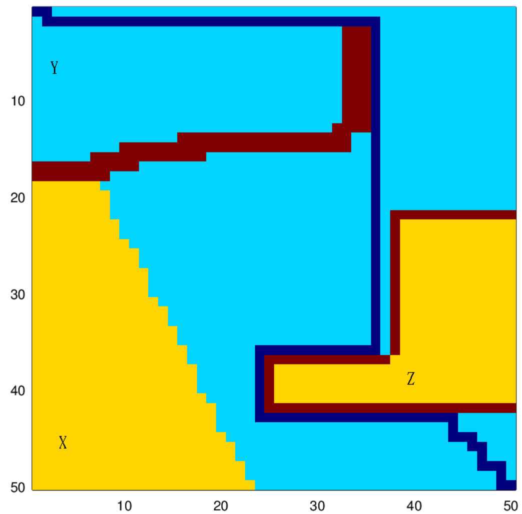


Simplified Sufficient Cause

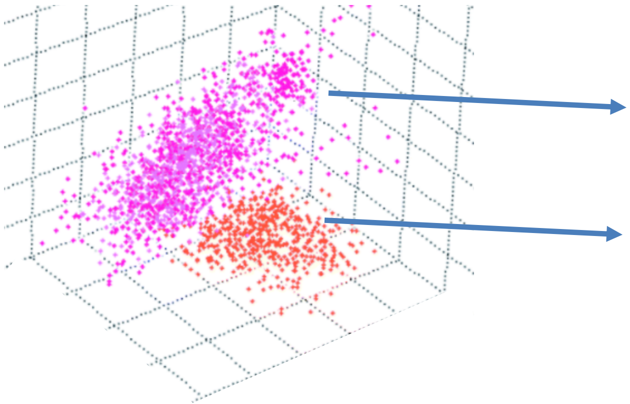
Formulation in AI:

- Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?: Explaining the Predictions of Any Classifier." *International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.
- Hayes, Bradley, and Julie A. Shah. "Improving Robot Controller Transparency Through Autonomous Policy Explanation." *International Conference on Human-Robot Interaction*. ACM, 2017.

Model Agnostic Explanation through Boolean Learning



Why does the path not go through Green?



Maps in which optimum path goes via green

Maps in which optimum path does not go via green

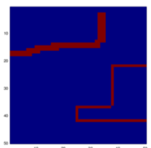
Let each point in k-dimensions (for some k) correspond to a map.

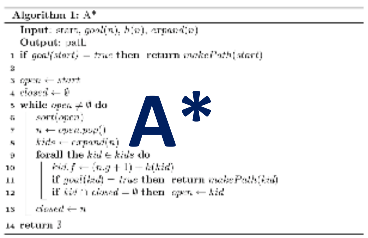
Find a Boolean formula ϕ such that

$\phi \Leftrightarrow Path\ contain\ z$

$\phi \Rightarrow Path\ contain\ z$

Explanations as Learning Boolean Formula

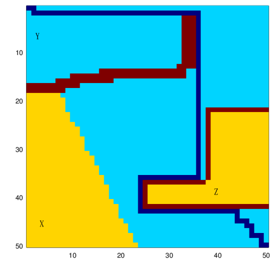




Algorithm 1: A*

```
Input: start, goal(n), h(n), expand(n)
Output: path
if goal(start) = true then return makePath(start)

open ← start
closed ← ∅
while open ≠ ∅ do
    sort(open)
    n ← open.pop()
    kids ← expand(n)
    forall the kid ∈ kids do
        kid.f ← (n.g + 1) + h(kid)
        if goal(kid) = true then return makePath(kid)
        if kid ∩ closed = ∅ then open ← kid
    closed ← n
return ∅
```

$\phi_{explain}$:
Using explanation vocabulary
Ex: Obstacle presence

ϕ_{query} :
Some property of the output
Ex: Some cells not selected

$$\phi_{explain} \Rightarrow \phi_{query}$$

$$\phi_{explain} \Leftrightarrow \phi_{query}$$

How difficult is it? Boolean formula learning

$\phi_{explain} \Rightarrow \phi_{query}$

$\phi_{explain} \Leftrightarrow \phi_{query}$

50x50 grid has $2^{2^{50X50}}$ possible explanations even if vocabulary only considers presence/absence of obstacles.

Scalability: Usually the feature space or vocabulary is large. For a map, its order of features in the map. For an image, it is order of the image's resolution.

Guarantee: Is the sampled space of maps enough to generate the explanation with some quantifiable probabilistic guarantee?

How difficult is it? Boolean formula learning

$$\phi_{explain} \Rightarrow \phi_{query}$$
$$\phi_{explain} \Leftrightarrow \phi_{query}$$

50x50 grid has $2^{2^{50X50}}$ possible explanations even if vocabulary only considers presence/absence of obstacles.

Scalability: Usually the feature space or vocabulary is large. For a map, its order of features in the map. For an image, it is order of the image's resolution.

Guarantee: Is the sampled space of maps enough to generate the explanation with some quantifiable probabilistic guarantee?

On PAC learning algorithms for rich Boolean function classes

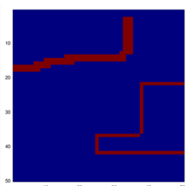
Rocco A. Servedio*

Department of Computer Science
Columbia University
New York, NY U.S.A.
rocco@cs.columbia.edu

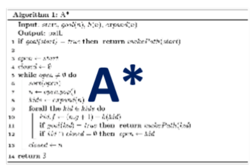
Theoretical Result:

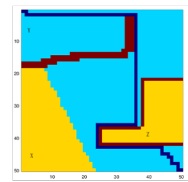
Learning Boolean formula even approximately is hard. 3-DNF is not learnable in Probably Approximately Correct framework unless RP = NP.

Two Key Ideas



$\phi_{explain}$:
Using explanation vocabulary
Ex: Obstacle presence





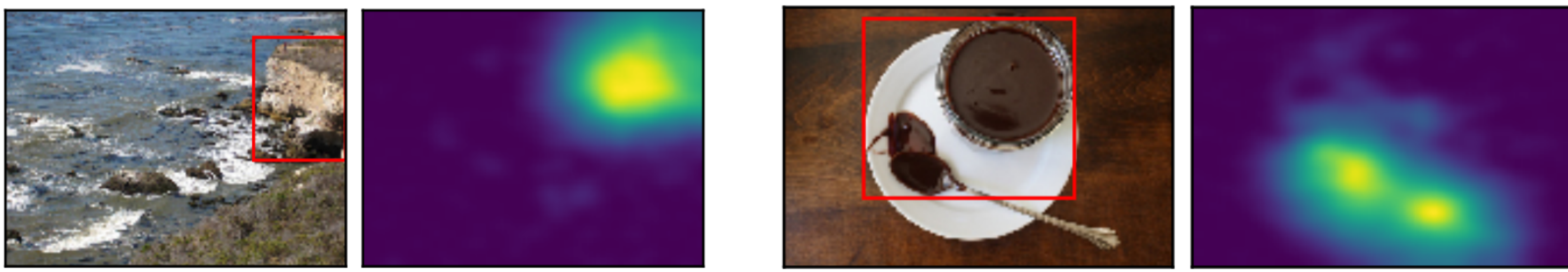
ϕ_{query} :
Some property of the output
Ex: Some cells not selected

1. Vocabulary is large.
2. How many samples (and what distribution) to consider for learning explanation ?
3. Learning Boolean formula with PAC guarantees is hard.

Active learning Boolean formula $\boldsymbol{\phi_{explain}}$ and not learning from fixed sample.

Explanations are often short and involve only few variables !

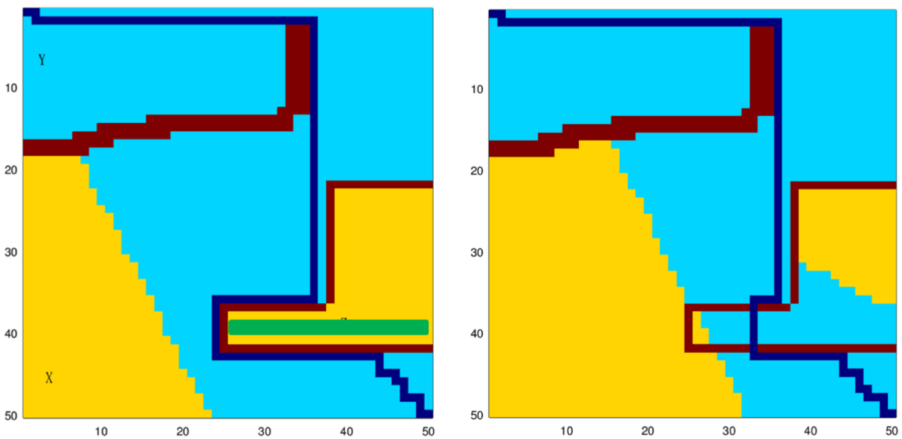
Two Key Ideas



Active learning Boolean formula $\boldsymbol{\phi_{explain}}$ and not learning from fixed sample.

Explanations are often short and involve only few variables !

Two Key Ideas



Involves only two variables.
If we knew which two, we had only $2^{2^2} = 16$ possible explanations.

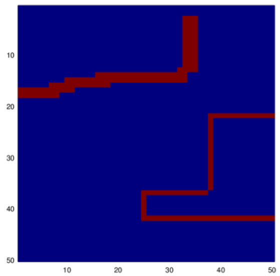
How do we find these relevant variables?

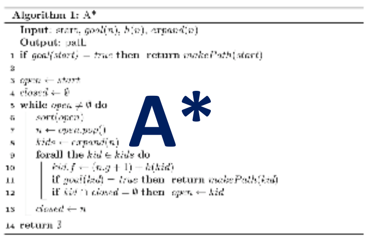
Active learning Boolean formula $\boldsymbol{\phi_{explain}}$ and not learning from fixed sample.

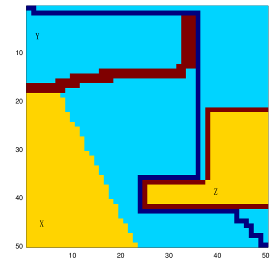
Explanations are often short and involve only few variables !

Actively Learning Boolean Formula

Oracle







ϕ_{query} :
Some property of the output
Ex: Some cells not selected

Assignments to V
m1 = (0,0,0,1,1,0,1)
m2 = (0,0,1,1,0,1,0)

$\phi_{explain}$ (V) :
Using explanation vocabulary
Ex: Obstacle presence

ϕ
Evaluates assignments and returns T,F

Actively Learning Relevant Variables

$$\boxed{Find\ U\ such\ that\ \phi_{explain}(\mathrm{V}) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|}$$

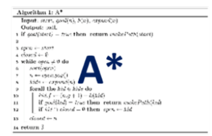
$$\phi_{explain}\ is\ sparse$$

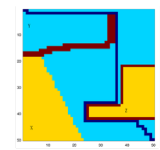
Actively Learning Relevant Variables

$$\boxed{Find\ U\ such\ that\ \phi_{explain}(\mathrm{V}) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|}$$

Assignments to V
m1 = (0,0,0,1,1,0,1)

Oracle





ϕ_{query} :
Some property of the output
Ex: Some cells not selected

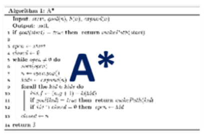
m1 : True

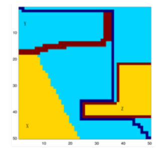
Actively Learning Relevant Variables

$$\boxed{Find\ U\ such\ that\ \phi_{explain}(\mathrm{V}) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|}$$

Assignments to V
m1 = (0,0,0,1,1,0,1)
m2 = (0,0,1,1,0,1,0)

Oracle





ϕ_{query} :
Some property of the output
Ex: Some cells not selected

m1: True, m2: False

Random Sample Till
Oracle differs

Actively Learning Relevant Variables

$$\boxed{Find\ U\ such\ that\ \phi_{explain}(V) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|}$$

Assignments to V

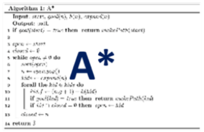
m1 = (0,0,0,1,1,0,1)

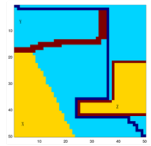
m2 = (0,0,1,1,0,1,0)

m3 = (0,0,0,1,1,1,0)

Binary Search Over Hamming Distance

Oracle





ϕ_{query} :
Some property of the output
Ex: Some cells not selected

m1: True, m2: False

Actively Learning Relevant Variables

$$\boxed{Find\ U\ such\ that\ \phi_{explain}(V) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|}$$

Assignments to V

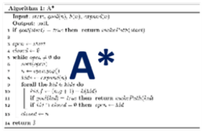
m1 = (0,0,0,1,1,0,1)

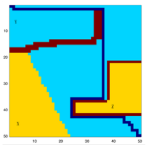
m2 = (0,0,1,1,0,1,0)

m3 = (0,0,0,1,1,1,0)

Binary Search Over Hamming Distance

Oracle





ϕ_{query} :
Some property of the output
Ex: Some cells not selected

m1: True, m2: False

m3: True

Actively Learning Relevant Variables

$$\boxed{Find\ U\ such\ that\ \phi_{explain}(V) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|}$$

Hamming Distance = 4

Assignments to V

~~m1 = (0,0,0,1,1,0,1)~~

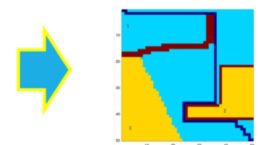
m2 = (0,0,1,1,0,1,0)

m3 = (0,0,0,1,1,1,0)

Hamming Distance = 2

Binary Search Over Hamming Distance

Oracle



ϕ_{query} :
Some property of the output
Ex: Some cells not selected

~~m1: True~~, m2: False
m3: True

Actively Learning Relevant Variables

$$\boxed{Find\ U\ such\ that\ \phi_{explain}(V) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|}$$

Hamming Distance = 2

Hamming Distance = 1

Assignments to V

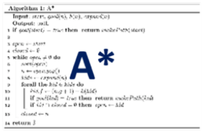
m2 = (0,0,1,1,0,1,0)

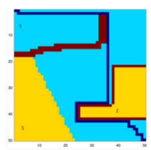
m3 = (0,0,0,1,1,1,0)

m4 = (0,0,1,1,1,1,0)

Binary Search Over Hamming Distance

Oracle





ϕ_{query} :
Some property of the output
Ex: Some cells not selected

m2: False, m3: True

m4: True

Actively Learning Relevant Variables

$$Find\ U\ such\ that\ \phi_{explain}(V) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|$$

Hamming Distance = 2

Assignments to V

m2 = (0,0,1,1,0,1,0)

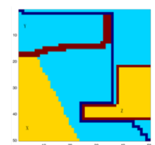
~~m3 = (0,0,0,1,1,1,0)~~

m4 = (0,0,1,1,1,1,0)

Hamming Distance = 1

Binary Search Over Hamming Distance

Oracle



ϕ_{query} :
Some property of the output
Ex: Some cells not selected

m2: False, ~~m3: True~~

m4: True

Actively Learning Relevant Variables

$$\boxed{Find\ U\ such\ that\ \phi_{explain}(V) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|}$$

Hamming Distance = 1

Assignments to V

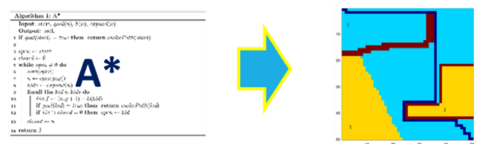
m2 = (0,0,1,1,$\boxed{0}$,1,0)

m4 = (0,0,1,1,$\boxed{1}$,1,0)

Fifth variable v_5 is relevant !!

Binary Search Over Hamming Distance

Oracle



ϕ_{query} :

Some property of the output

Ex: Some cells not selected

m2: False, m4: True

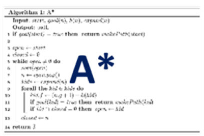
Actively Learning Relevant Variables

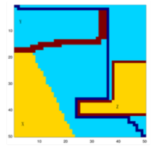
$$\boxed{Find\ U\ such\ that\ \phi_{explain}(\mathrm{V}) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|}$$

Repeat to find all relevant variables

Binary Search Over Hamming Distance

Oracle





ϕ_{query} :
Some property of the output
Ex: Some cells not selected

m2: False, m4: True

Actively Learning Relevant Variables

$$\boxed{Find\ U\ such\ that\ \phi_{explain}(\mathrm{V}) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|}$$

For each assignment to relevant variables

$2^{|U|}$

Random Sample Till Oracle differs

$ln(1/(1-\kappa))$

Binary Search Over Hamming Distance

$ln(|V|)$

Relevant variables of $\boldsymbol{\phi_{explain}}$ found with confidence $\boldsymbol{\kappa}$ in

$$\mathbf{2^{|U|}}\boldsymbol{ln}(|\boldsymbol{V}|/(\mathbf{1}-\boldsymbol{\kappa}))$$

Actively Learning Boolean Formula

$$\boxed{Find\ U\ such\ that\ \phi_{explain}(V) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|}$$

Used distinguishing example based approach from ICSE'10

Susmit Jha, Sumit Gulwani, Sanjit A Seshia, and Ashish Tiwari. Oracle-guided component-based program synthesis. In *2010 ACM/IEEE 32nd International Conference on Software Engineering*, volume 1, pages 215–224. IEEE, 2010.

Scales to ~200 variables

Build Truth Table for the relevant variables U

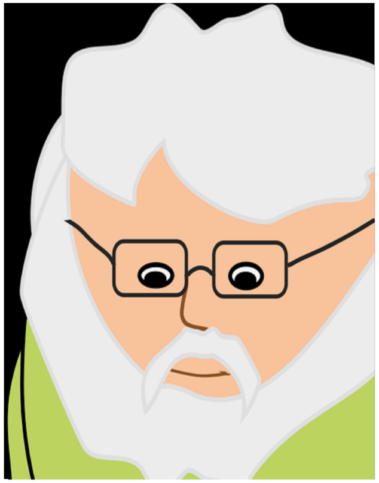
$Worst\ Case$: $2^{|U|}$

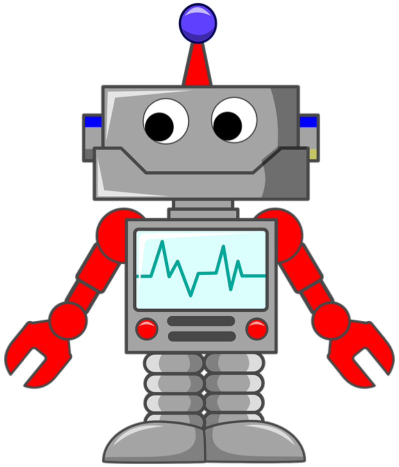
$\boldsymbol{\phi_{explain}}$ **found with confidence $\boldsymbol{\kappa}$ in**

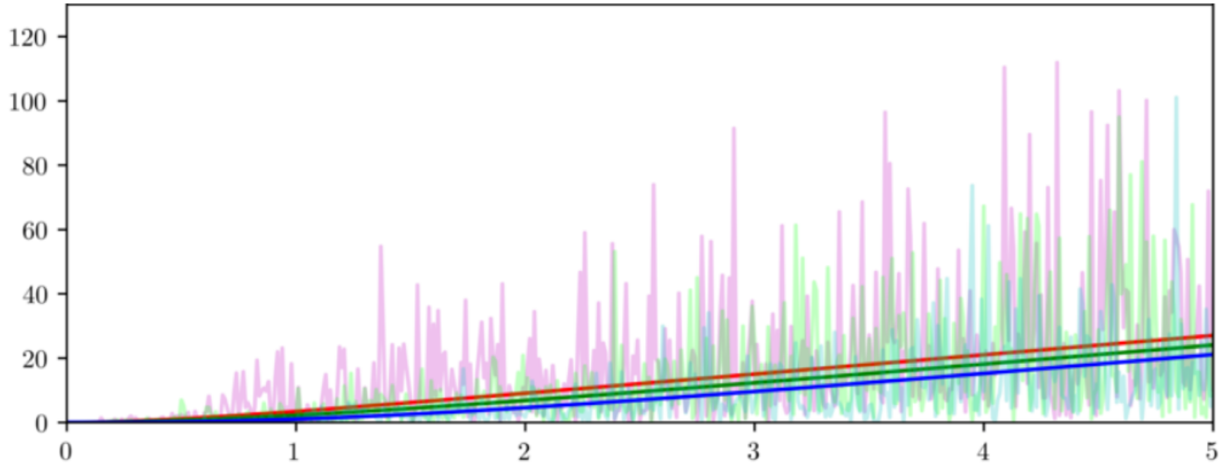
$$\mathbf{O}(2^{|\mathrm{U}|} \boldsymbol{ln}(|\boldsymbol{V}|/(\mathbf{1}-\boldsymbol{\kappa})))$$

A PAC Learning Framework

Interpretability: Observed Time Traces



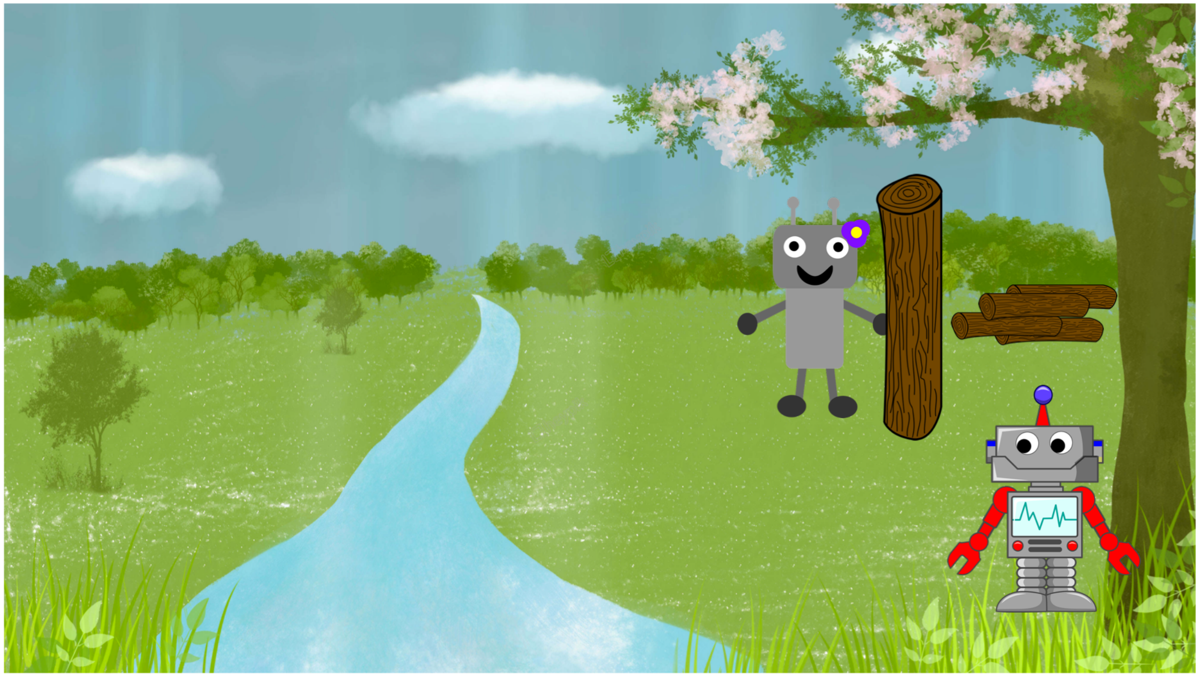


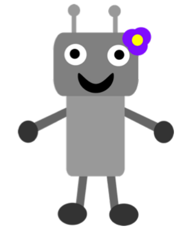


1. Noisy 2. Large corpus and not selected examples 3. Opportunity to query

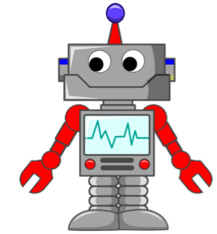
Interpretable Learning for Shared Intentionality

Inferring and Conveying Intentionality: Beyond Numerical Rewards to Logical Intentions. Susmit Jha and John Rushby. AAAI Spring Symposium on Conscious AI Systems, 2019





Alice



Bob

Humans can undertake novel, collective behavior, or teamwork
Capability to communicate goals, plans and ideas to create shared intentionality

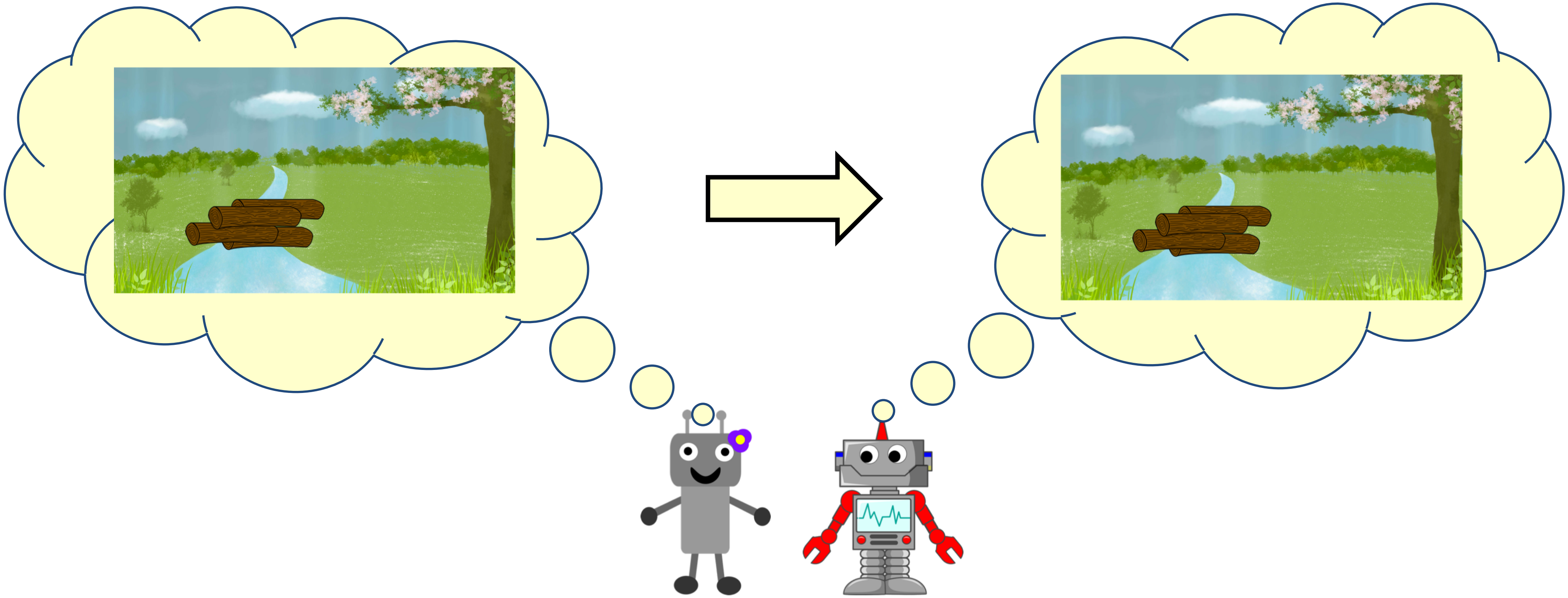
Consider two autonomous agents Alice and Bob with cognition capability.

Alice can invent a novel behavior – use tree logs to try and build a bridge.

How will Bob, who is watching Alice, understand Alice's goal and assist her ?

Alice's mental state needs to be recreated in Bob's brain for Bob to collaborate with Alice.

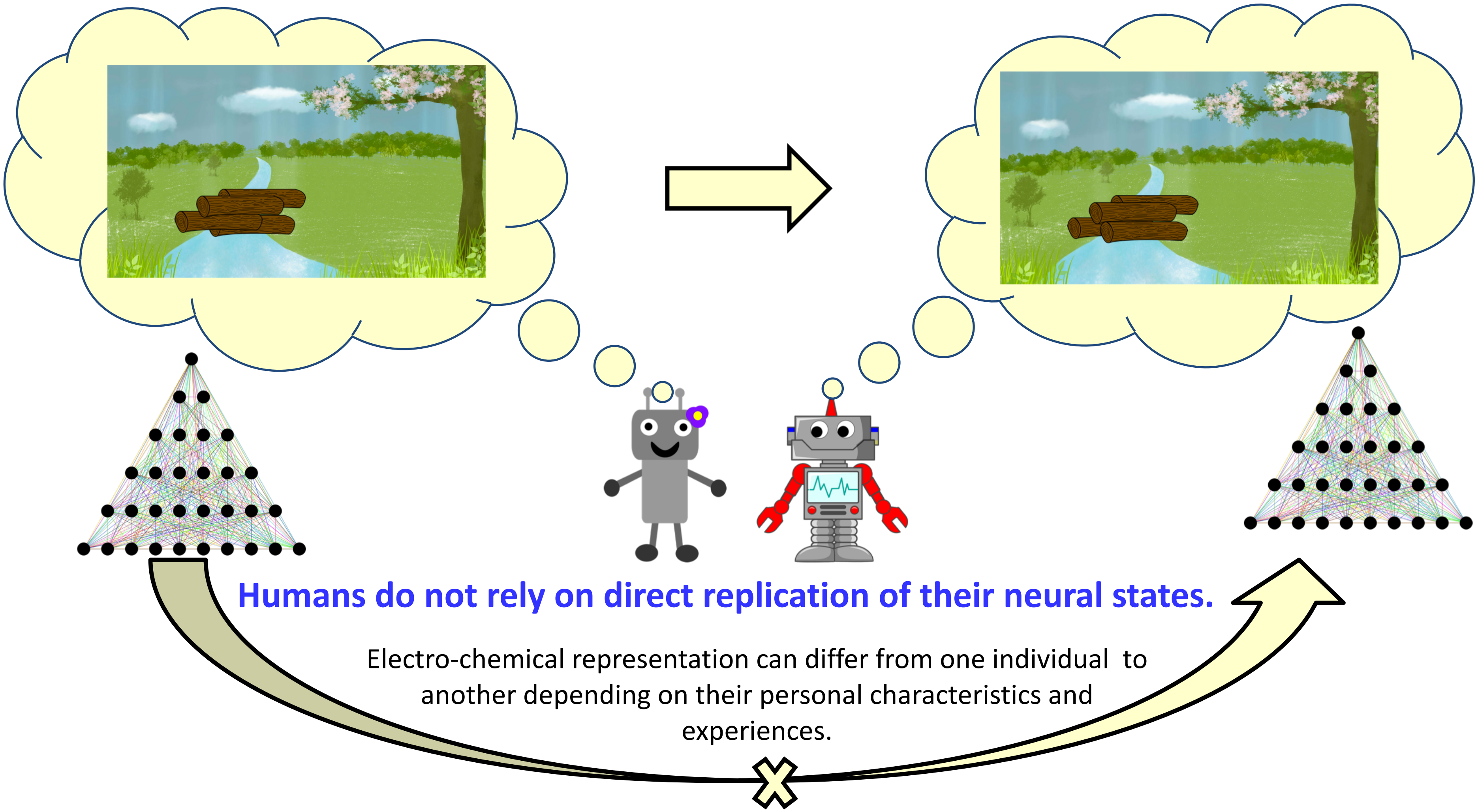
Interpretable Learning for Shared Intentionality



Alice’s mental state needs to be recreated in Bob’s brain for Bob to collaborate with Alice.

Shared Intentionality: Mental Cloning?

Gweon, H., Saxe, R. (2013). Developmental cognitive neuroscience of Theory of Mind. *Neural Circuit Development and Function in the Brain: Comprehensive Developmental Neuroscience.*

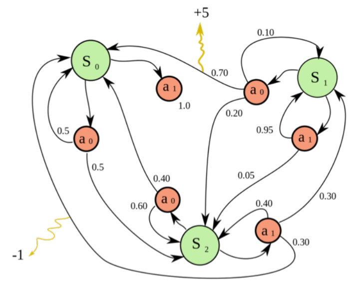


Humans do not rely on direct replication of their neural states.

Electro-chemical representation can differ from one individual to another depending on their personal characteristics and experiences.

Alice’s mental state needs to be recreated in Bob’s brain for Bob to collaborate with Alice.

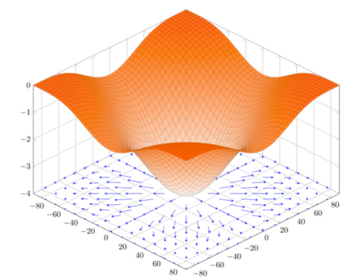
Communicating Using Demonstrations: Non-Markovian IRL



Environment Markov Decision Process

Noisy Expert Demonstrations

Inverse Reinforcement Learning



Numerical Reward Function

- Demonstrations and rewards are often non-Markovian due to mental state of the actor not directly modeled by environment MDP.

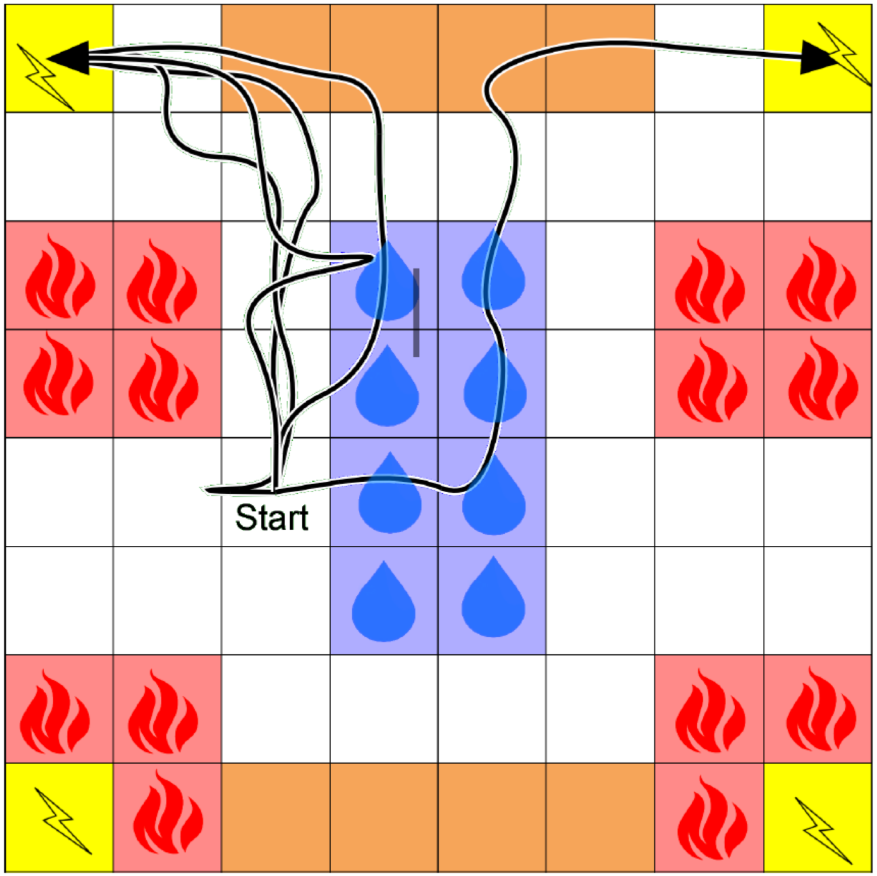
- Composability? , Resilience to changes in task context? Interpretability?

Communicating Using Demonstrations: More involved example

1. Avoid fire (red).

2. Eventually Recharge (yellow).

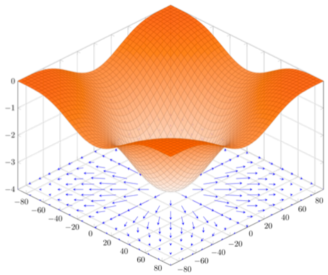
3. If you touch the water (blue) then dry off (brown) before recharging (yellow).



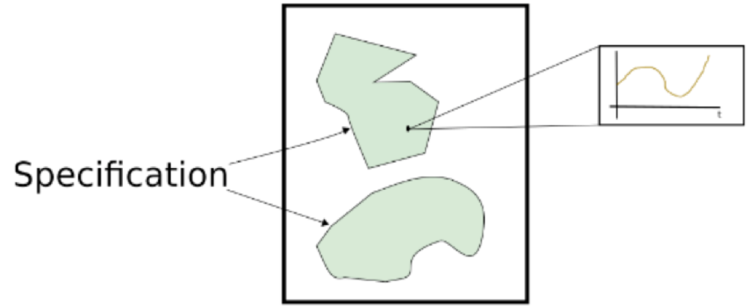
Explicit reduction to non-Markovean representation suffers from the curse of history.

a. (4 colors)^(10 time steps) = 2^20 traces ≈ 1048576
b. #specifications = 2^(2^20) ≈ 10^315652

Communicating Using Demonstrations: Temporal logic specifications



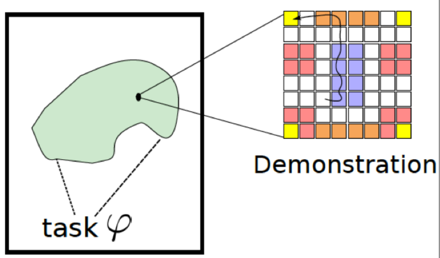
Numerical Reward Function



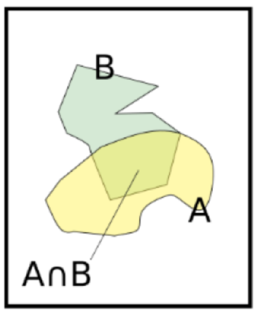
Traces

Trace Properties as Task Specifications

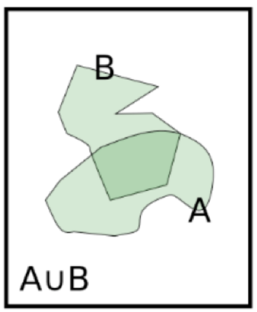
$$r_\varphi(\xi) = \begin{cases} 1 & \text{if } \xi \in \varphi \\ 0 & \text{otherwise} \end{cases}$$



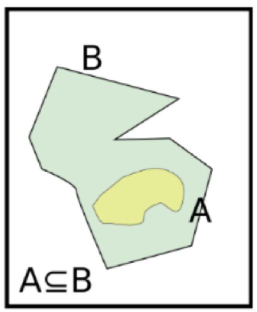
Traces



Traces



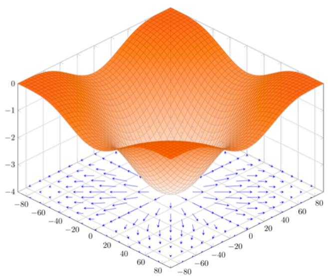
Traces



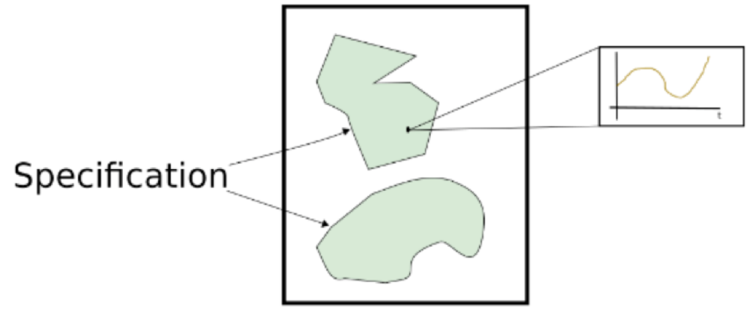
Traces

- Composable
- Resilient to changes in task context
- Interpretable
- Can leverage formal methods tools

Communicating Using Demonstrations: Temporal logic specifications



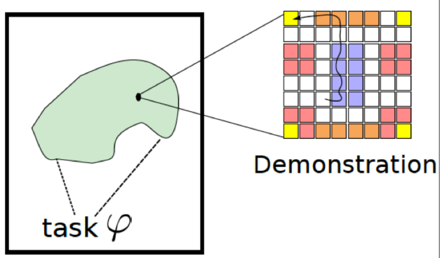
Numerical Reward Function



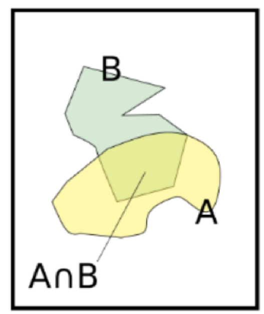
Traces

Trace Properties as Task Specifications

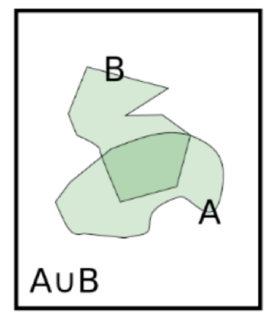
$$r_\varphi(\xi) = \begin{cases} 1 & \text{if } \xi \in \varphi \\ 0 & \text{otherwise} \end{cases}$$



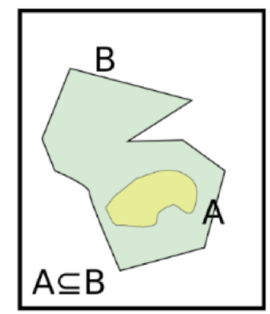
Traces



Traces



Traces

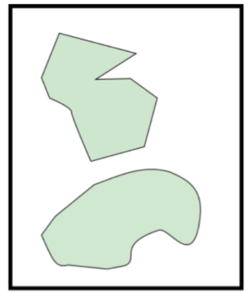
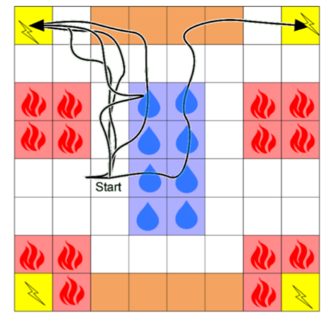


Traces

- Pnueli, Amir. "The temporal logic of programs." IEEE, 1977.
- Donzé, Alexandre, and Oded Maler. "Robust satisfaction of temporal logic over real-valued signals." *FORMATS*, 2010.
- Jha, Susmit, Vasumathi Raman, Dorsa Sadigh, and Sanjit A. Seshia. "Safe autonomy under perception uncertainty using chance-constrained temporal logic." *Journal of Automated Reasoning* 60, 2018.

Communicating Using Demonstrations: Specification Inference Problem

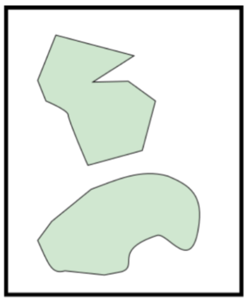
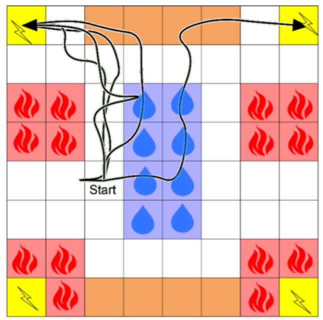
Like most inverse problems, this problem is underspecified.

What is Pr( | ) ?

Specification

- Intent satisfaction is Boolean. Either Alice/Bob did the task or didn't.
- Assuming Alice is at least better at performing the task than a random action policy.
- Applying the principle of maximum entropy select the the distribution.
  - Inspired by Maximum Entropy Principle (also used in Inverse Reinforcement Learning)

Communicating Using Demonstrations: KL Divergence

Pr( | 

) $\propto e^{D_{KL}\left(\mathcal{B}(\overline{\varphi}) \,\|\, \mathcal{B}(\widehat{\varphi})\right)}$

Specification

Demonstrations

Bernoulli Distribution

Satisfaction probability for Alice given dynamics

Satisfaction probability given uniformly random actions

Marcell Vazquez-Chanlatte, Susmit Jha , Ashish Tiwari, Mark K. Ho and Sanjit A. Seshia. Learning Task Specifications from Demonstrations. NeurIPS, 2018

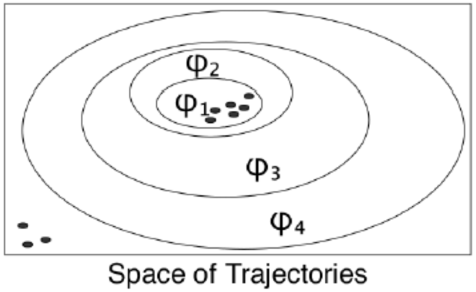
Communicating Using Demonstrations: Computing posterior

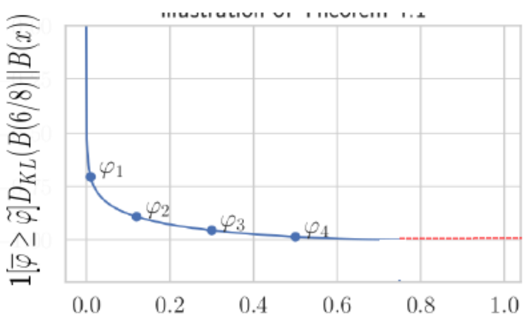
Maximum a Posteriori

$$\max_{\varphi} D_{KL}\left(\mathcal{B}(\overline{\varphi}) \parallel \mathcal{B}(\widehat{\varphi})\right)$$

Algorithm Sketch

If one fixes the measured sat probability, the KL-divergence term in the model is convex in the random satisfaction rate. This enables an efficient lattice based search for the most probable specification.





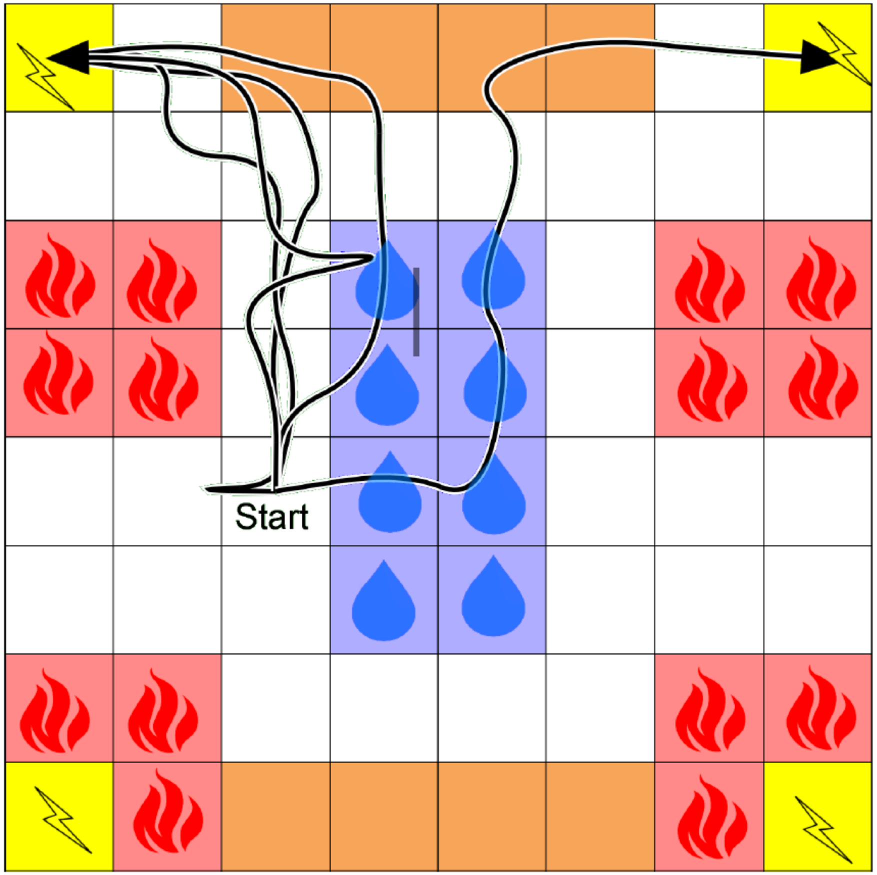
Marcell Vazquez-Chanlatte, Susmit Jha , Ashish Tiwari, Mark K. Ho and Sanjit A. Seshia. Learning Task Specifications from Demonstrations. NeurIPS, 2018

Communicating Using Demonstrations: More involved example

1. Avoid fire (red).

2. Eventually Recharge (yellow).

3. If you touch the water (blue) then dry off (brown) before recharging (yellow).



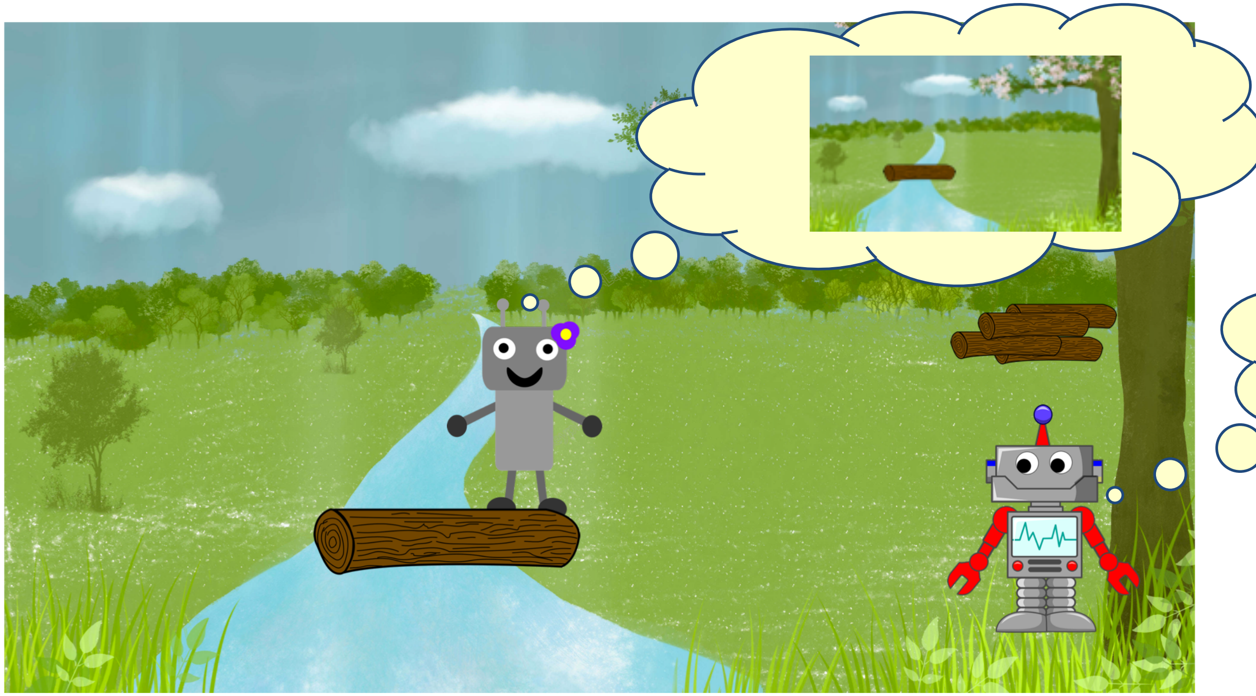
Temporal Logic Specification

H: Historically
O: Once
S: Since

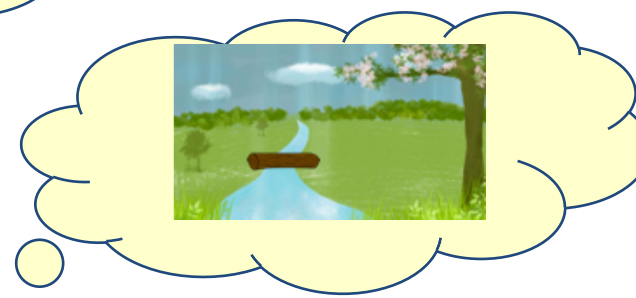
$$(H \neg red \wedge O\ yellow) \wedge H\big((yellow \wedge O\ blue) \Rightarrow (\neg blue\ S\ brown)\big)$$

A Candidate Mechanism to Computationally Implement Shared Intentionality

Inferring and Conveying Intentionality: Beyond Numerical Rewards to Logical Intentions. Susmit Jha and John Rushby. AAAI Spring Symposium, Towards Conscious AI Systems, 2019



Marcell Chanlatte, Susmit Jha , Ashish Tiwari, Mark K. Ho and Sanj A. Seshia. Learning Task Specifications from Demonstrations. NeurIPS, 2018



Jha, Susmit et al. "Safe autonomy under perception uncertainty using chance-constrained temporal logic." *Journal of Automated Reasoning* 60, 2018

Find Specification as Maximum a Posteriori

$$\max_{\varphi} D_{KL}\left(\mathcal{B}(\overline{\varphi}) \parallel \mathcal{B}(\widehat{\varphi})\right)$$

Interpretability / Explanation Generation in TRINITY

- **Inferring and Conveying Intentionality: Beyond Numerical Rewards to Logical Intentions.** Susmit Jha and John Rushby.
  AAAI Spring Symposium, Towards Conscious AI Systems, 2019

- **Learning Task Specifications from Demonstrations.** Marcell Vazquez-Chanlatte, Susmit Jha, Ashish Tiwari, Mark K. Ho and Sanjit A. Seshia.
  Neural Information Processing Systems (NeurIPS), 2018

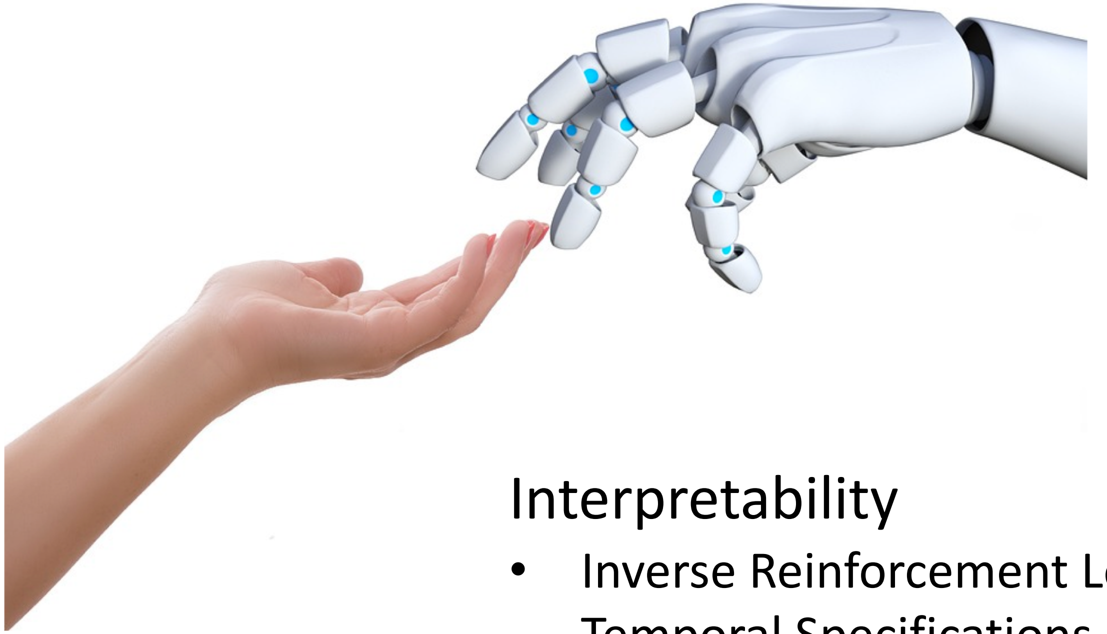
- **Explaining AI Decisions Using Efficient Methods for Learning Sparse Boolean Formulae.** Susmit Jha, Tuhin Sahai, Vasumathi Raman, Alessandro Pinto and Michael Francis.
  Journal of Automated Reasoning, 2018

- **On Learning Sparse Boolean Formulae For Explaining AI Decisions.** Susmit Jha, Vasumathi Raman, Alessandro Pinto, Tuhin Sahai, and Michael Francis.
  NASA Formal Methods (NFM), 2017

Rest of the Talk

Trust

- Global Assume/Guarantee Contracts on DNNs
- Extracting and Integrating Temporal Logic into Learned Control



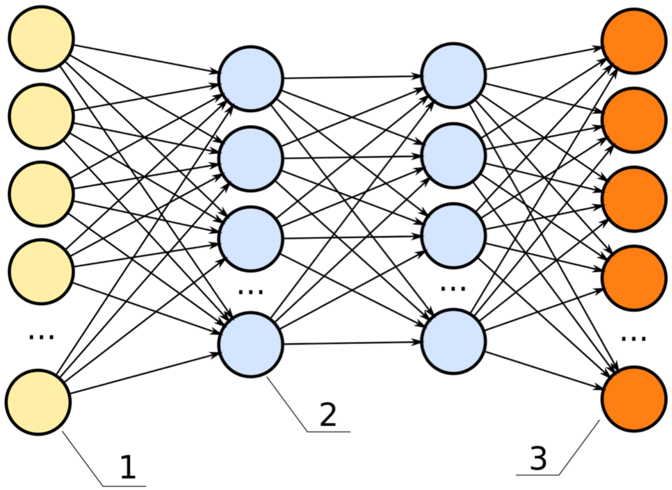
Interpretability

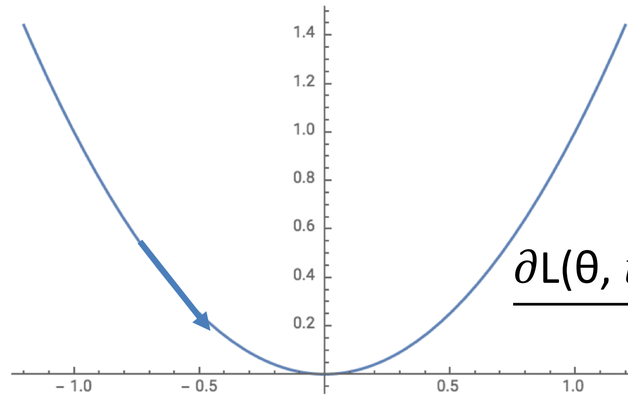
- Inverse Reinforcement Learning of Temporal Specifications

Resilience

- **Adversarial Robustness**

Adversarial Examples in Deep Learning



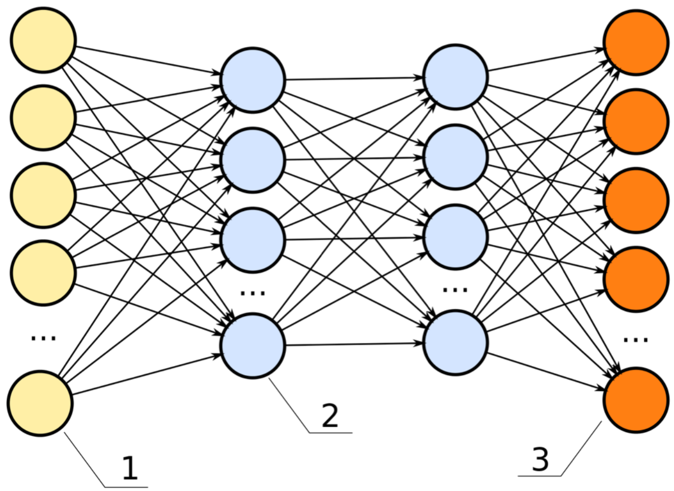


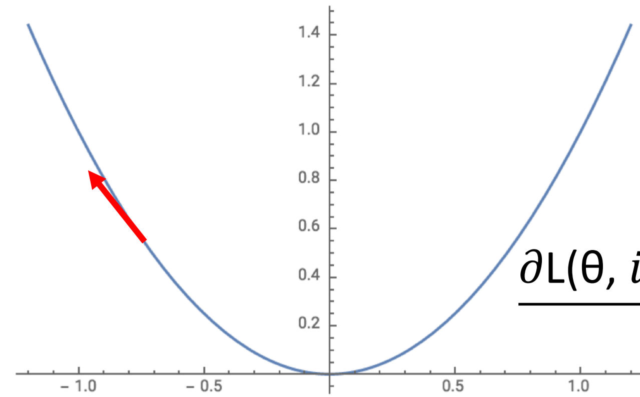
$$\frac{\partial \mathrm{L}(\theta, input_i \text{ , output})}{\partial \theta}$$

Loss function L(θ, $input_i$, output) with θ the parameters of the models.
Measures how good the prediction of the model is on a specific example.

To train a neural network we compute the derivative of L according to the weights θ and update θ in order to decrease the loss value.

Adversarial Examples in Deep Learning





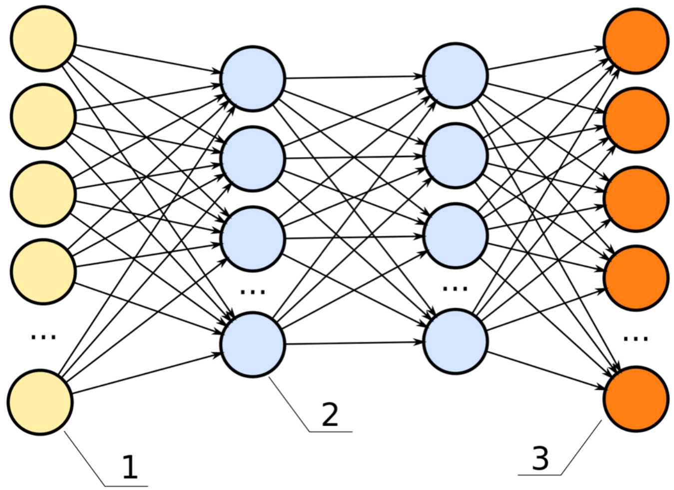
$$\frac{\partial L(\theta, input_i, \text{output})}{\partial input_i}$$

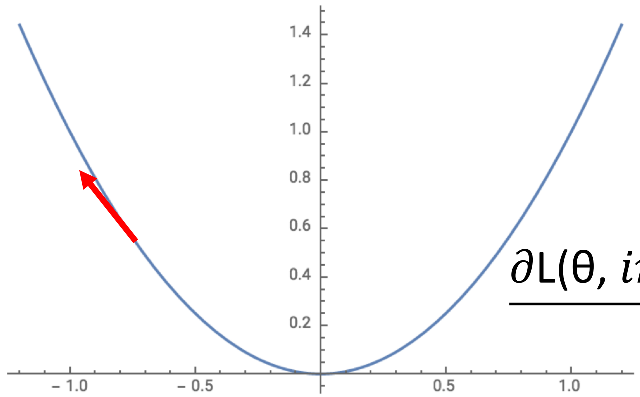
Loss function L(θ, $input_i$, output) with θ the parameters of the models.
Measures how good the prediction of the model is on a specific example.

To train a neural network we compute the derivative of L according to the weights θ and update θ in order to decrease the loss value.

To create an adversarial sample, we compute the derivative of L according to the input and use the result to update the pixel values in order to increase the loss value.

Adversarial Examples in Deep Learning





$$\frac{\partial L(\theta, input_i \text{ , output})}{\partial input_i}$$

Loss function L(θ, $input_i$, output) with θ the parameters of the models.
Measures how good the prediction of the model is on a specific example.

To train a neural network we compute the derivative of L according to the weights θ and update θ in order to decrease the loss value.

$$input \ = input + \epsilon \ \ sign \ (\frac{\partial L(\theta, input, \text{output})}{\partial input})$$

Fast Gradient Sign Method

Adversarial Defense by Irrelevant Factor Identification

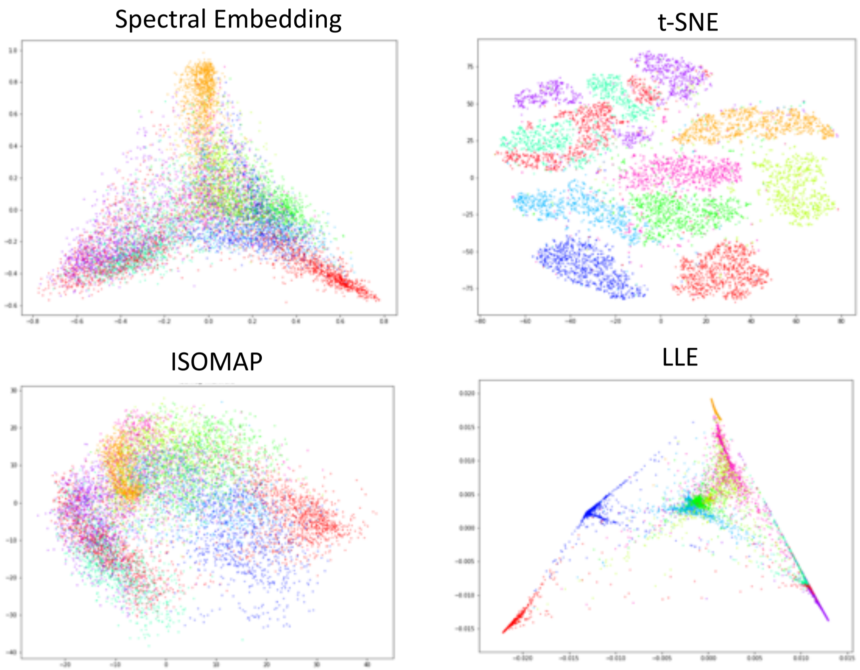
Causal Modeling

Attribution-driven Causal Analysis for Detection of Adversarial Examples. Susmit Jha et. al. SafeML/ICLR, 2019

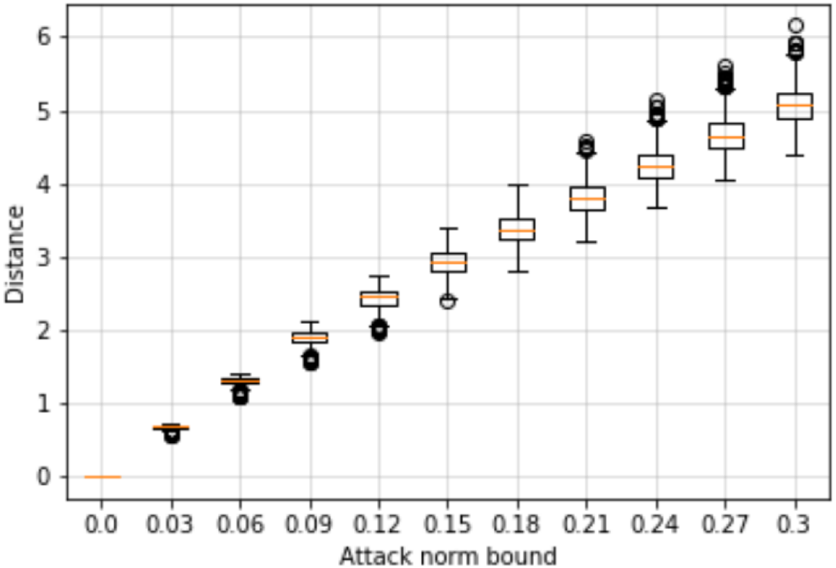
Geometric Invariants

Detecting Adversarial Examples Using Data Manifolds. Susmit Jha, Uyeong Jang, Somesh Jha and Brian Jalaian. IEEE Military Communications Conference (MILCOM), 2018

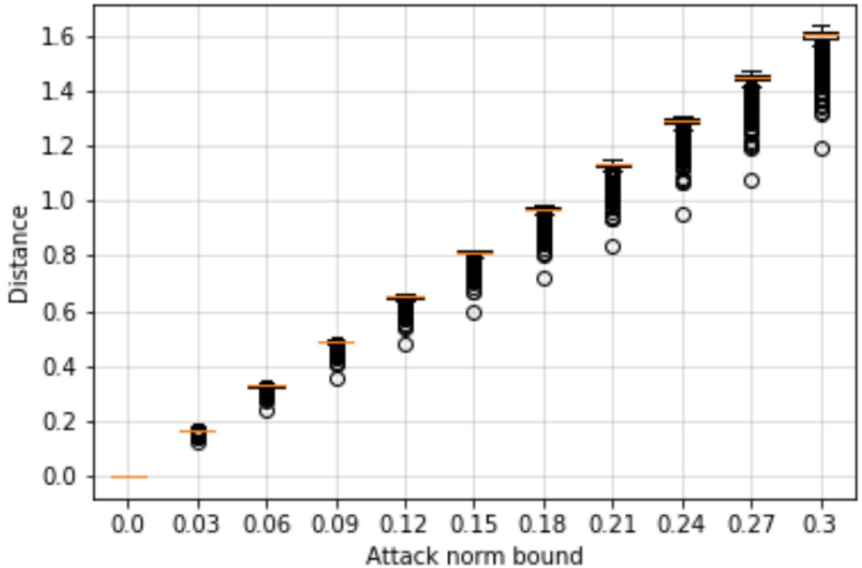
Manifold-based Robust Learning. Susmit Jha, Uyeong Jang, Somesh Jha and Brian Jalaian. NATO SET 262, 2018



MNIST and CFAR: FGSM Attack and Manifold Distance



MNIST



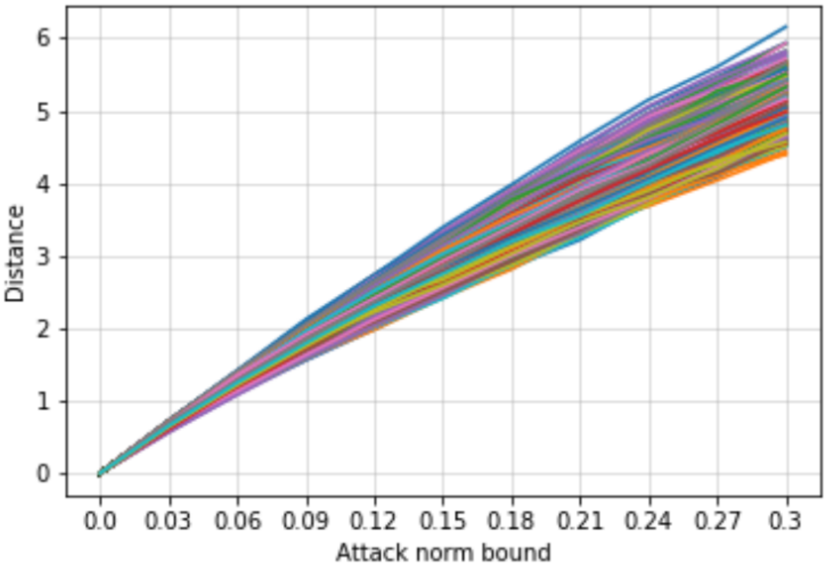
CFAR

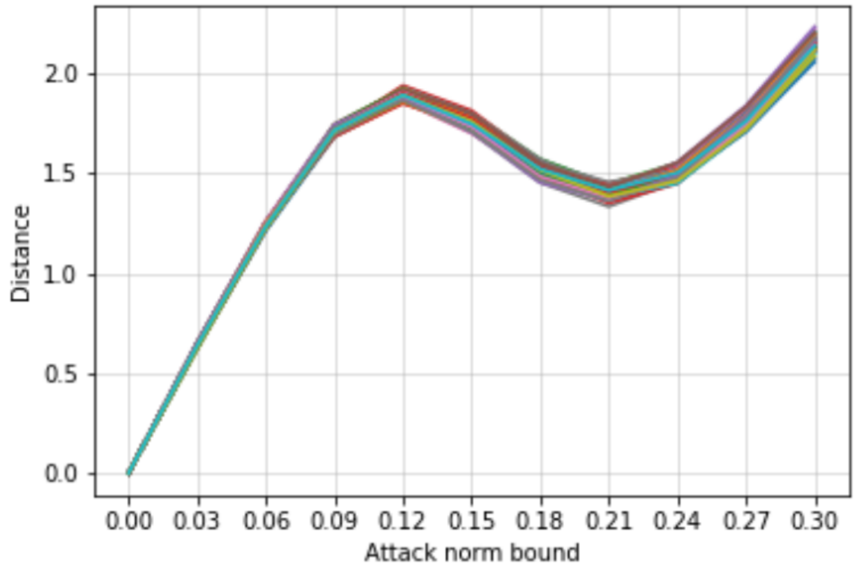
Used CleverHans system for generating attacks.

$$\max_{||x^{adv}-x||_\infty \leq \epsilon} Loss(x^{adv}, l_x)$$

Nicolas Papernot et. al.

Manifold Distance in Input Space and Logit Space





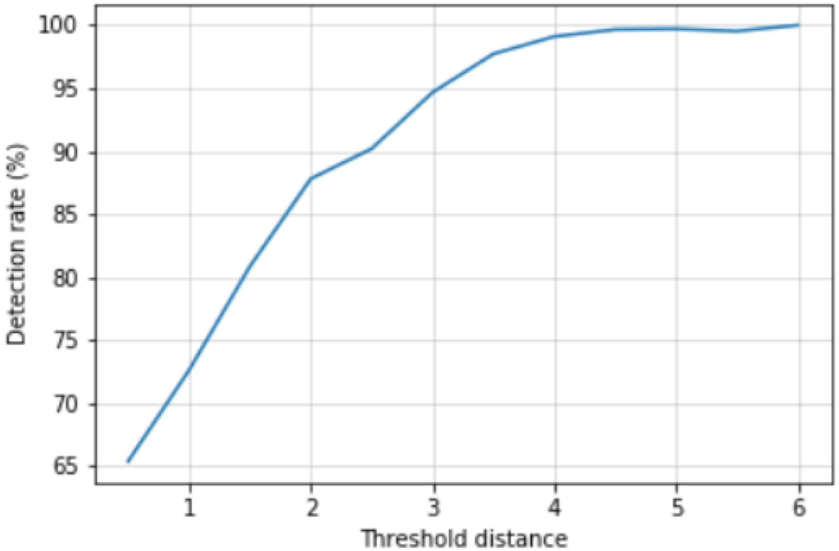
Hypothesized in literature that the deeper layers of a deep neural network provide more linear and unwrapped manifolds in comparison to the input space. Thus, the task of identifying the manifold becomes easier as we progress from the input space to the more abstract feature spaces all the way to the logit space.

Yoshua Bengio, Gregoire Mesnil, Yann Dauphin, and Salah Rifai. ´ Better mixing via deep representations. In International Conference on Machine Learning, pages 552–560, 2013.

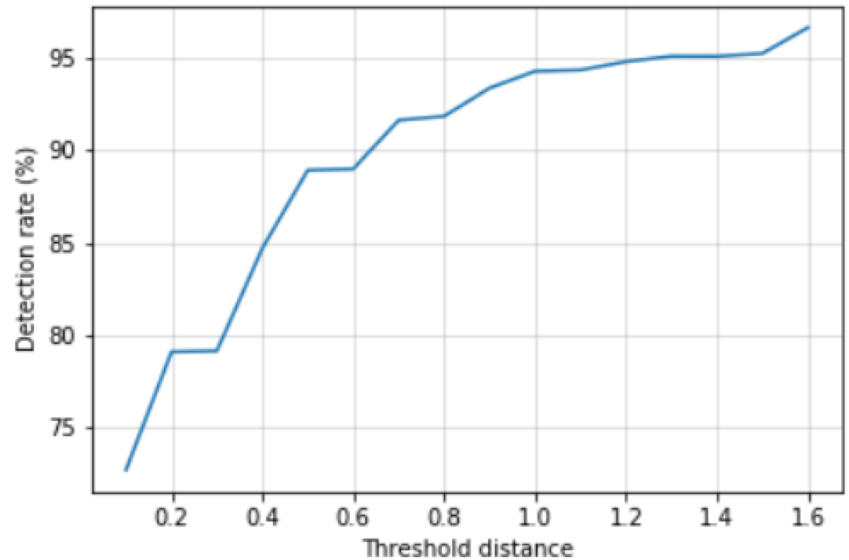
Jacob R Gardner, Paul Upchurch, Matt J Kusner, Yixuan Li, Kilian Q Weinberger, Kavita Bala, and John E Hopcroft. Deep manifold traversal: Changing labels with convolutional features. arXiv preprint arXiv:1511.06421, 2015

Detection Rate Using Manifold Distance

MNIST



CFAR



The kernel density estimation can be used to measure the distance d(x) of x from the data manifold of training set. Specifically, d(x) = $\frac{1}{|X|}\sum_{\{x_i \in X\}} k(x_i, x)$, where X is the full data set and k(·, ·) is a kernel function such as Gaussian or a simple L^∞ or L^2 norm.

Thanks!

Questions?