CWI, Amsterdam

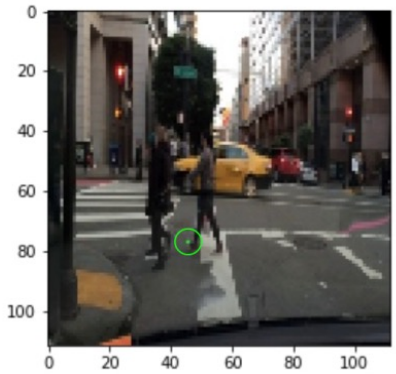
20th May 2025

Provable guarantees for data-driven policy synthesis: a formal methods perspective

Prof. Marta Kwiatkowska
Department of Computer Science
University of Oxford

Safety and security risks in AI decision making

- Well known that neural networks are unstable to adversarial perturbations



Adversarial example

(Red light classified as green after one pixel change)

Physical attack

Patch attack

Real traffic sign

- For high-stakes applications, need provable guarantees on correctness
- Yet AI/ML community focuses on performance – formal verification to the rescue?

Feature-Guided Black-Box Safety Testing of Deep Neural Networks. Wicker *et al*, In Proc. TACAS, 2018.

Michael Passingham
Senior researcher & writer

New research from Which? reveals that more than half of drivers are turning off safety tech in their cars with many finding the tech annoying, distracting or even dangerous. We explain why, and how to get the best from your current car or your next purchase.

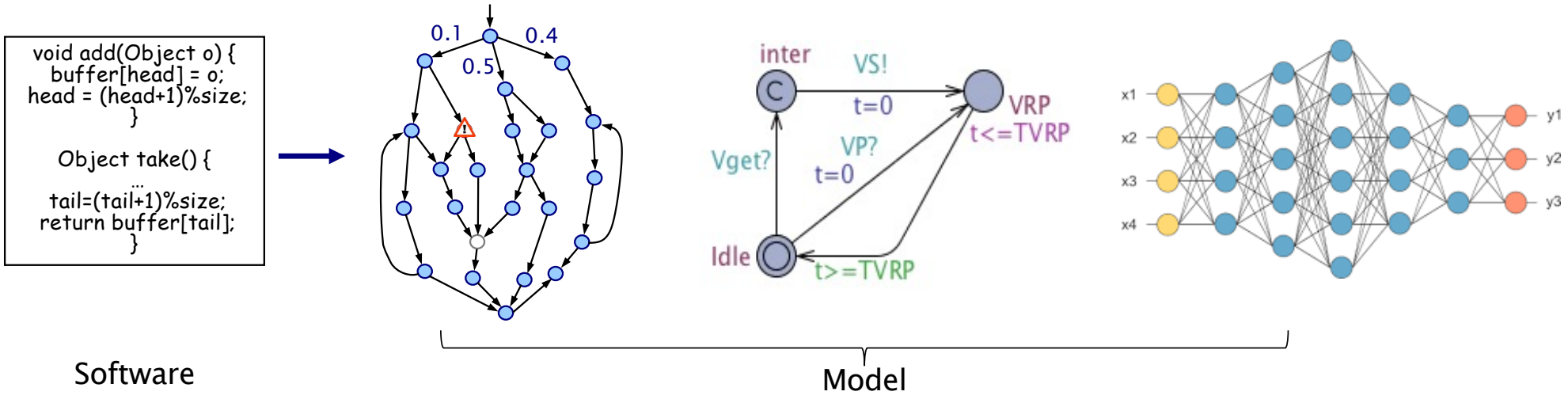
Like airbags and crumple zones, various car safety technologies are mandatory on new cars. While airbags are considered 'passive' safety tech – they only activate when you crash – Advanced Driver-Assistance Systems (ADAS) are 'active' and are intended to prevent you from having an accident in the first place.

With driver error a leading cause of road accidents in the UK, the best case scenario with ADAS features is that they prevent avoidable accidents. These include accidents where the driver unintentionally leaves their lane, is driving too fast or hasn't spotted an obstacle ahead of them.

However, Which? has found evidence that these features are being habitually turned off by drivers, with just over half of drivers who have an ADAS feature on their car reporting they turn at least one feature off at least some of the time. And when the tech is off, it isn't protecting anybody.

This highlights that there's a lot of room for improvement in the way these systems are implemented and explained.

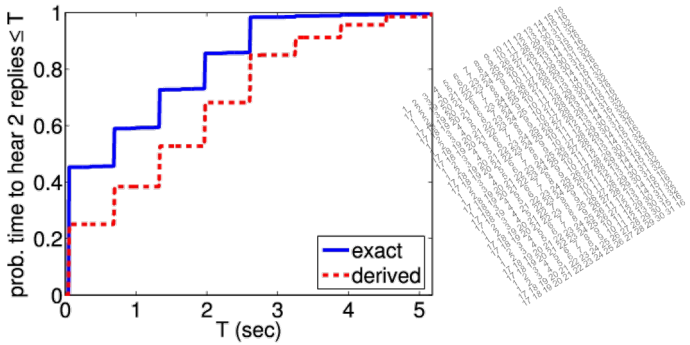
Formal verification provides provable guarantees



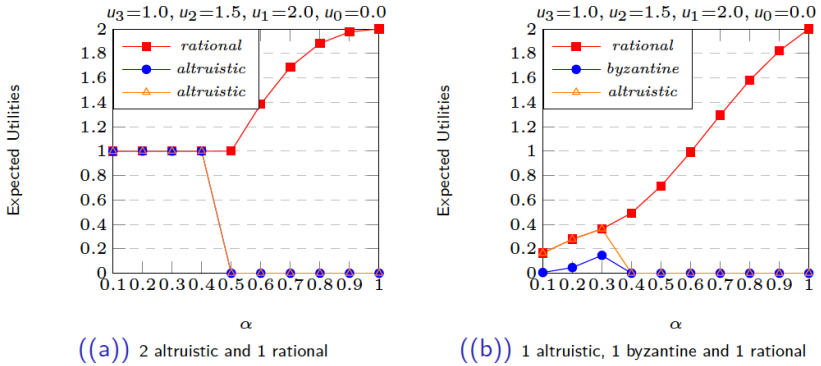
- Modelling = rigorous, mathematical abstraction
- Verification = proof that the model satisfies specification
- Synthesis = correct-by-construction model/policy from specification
- Automated = algorithmic, implemented in software

Probabilistic Model Checking in Autonomy. Kwiatkowska *et al*, Ann Rev of Control, Robotics and Aut. Sys. (2022).

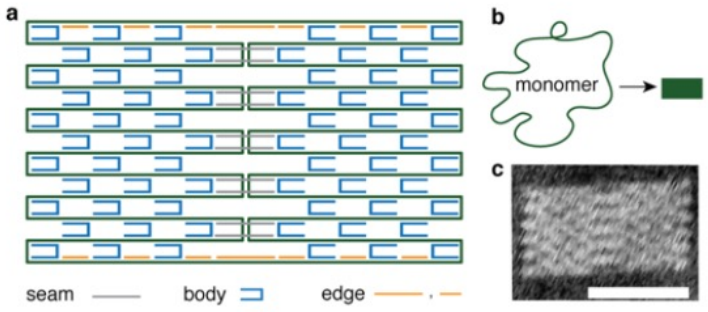
Multiple applications and use cases!



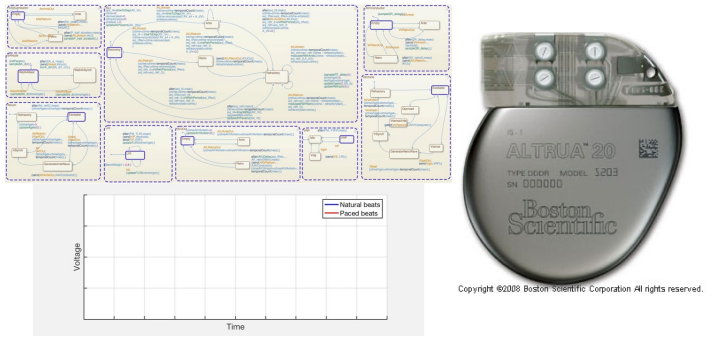
Protocol debugging



Protocol verification



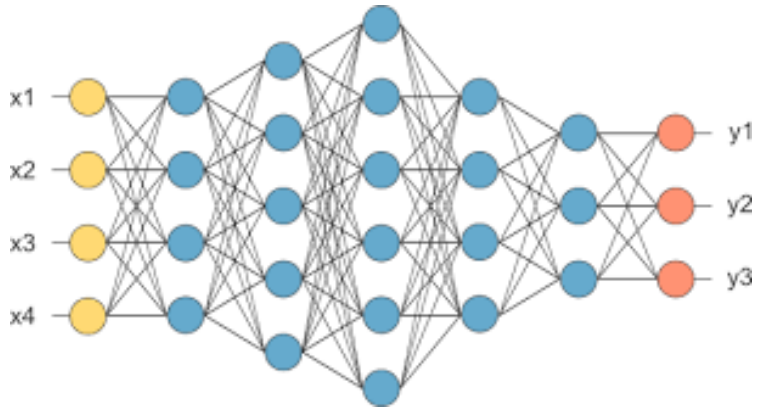
Prediction of DNA folding



Optimal controller synthesis

Formal verification for neural networks (NNs)

- Rigorous formal verification
  - can provide provable guarantees, e.g. that no adversarial examples exist
  - enables robustness certification and correct-by-construction synthesis
  - crucial part of safety assurance



- Neural network models more challenging
  - black box, lack interpretability
  - high-dimensional function
  - interplay between architecture and training (non-linear optimization)

Image classifier is a function f: $R^n \rightarrow \{c_1,\ldots c_k\}$
Learnable weights and bias

Approximates human perception from M training examples

- Much progress since 2017: Reluplex, DLV, DeepPoly, ReluVal, CROWN, …

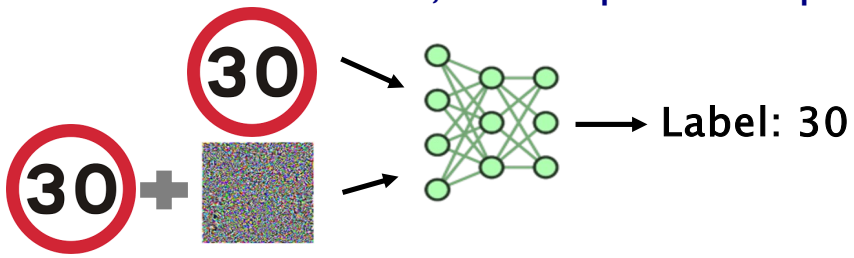
Safety Verification of Deep Neural Networks. CAV 2017 keynote

This talk: provable guarantees via formal verification

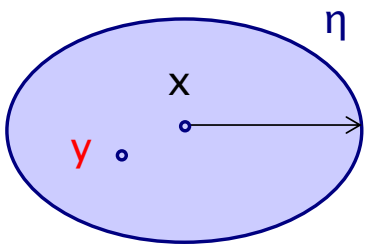
- Overview of recent research
- Focus on (data-driven) NN/RL policies and components
- Brief recap of (local) adversarial robustness certification
  - crucial part of safety assurance, pre-deployment
- A selection of snapshots
  - pre-image approximation
  - quantitative verification
  - exploiting causality
  - handling uncertainty
  - neuro-symbolic models
- Conclusions and future directions

Recap of adversarial robustness

- Focus on local adversarial robustness, for a specific input

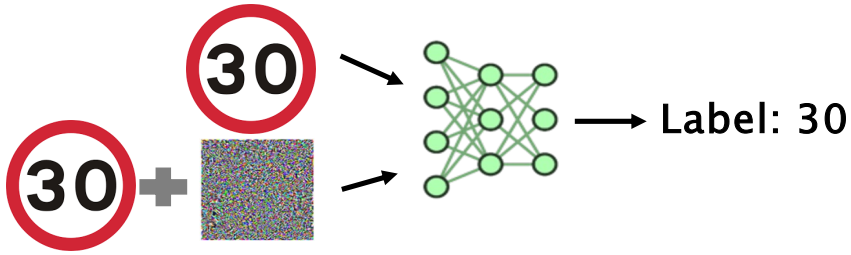


- Informally, no perturbation results in a misclassification
- More formally, assume given
  - trained neural network classifier $f : R^m \rightarrow \{c_1,\ldots c_k\}$
  - region η centred at x wrt distance function, e.g. L^2, L^∞
- Define local robustness at x wrt η by (SAT friendly)
  - $\nexists y \in \eta$ such that $f(x) \neq f(y)$
- Here, focus on computing provable guarantees on correctness, rather than constructing defences

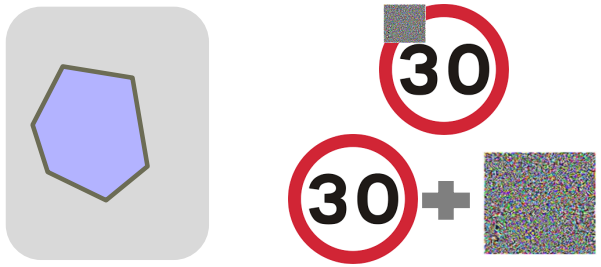


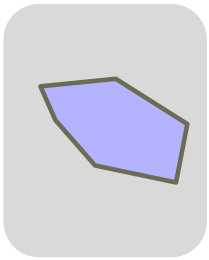
Neural network verification

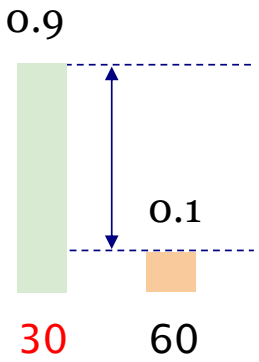
- Given a neural network $f: R^n \rightarrow R^m$, the NN verification problem is defined as $(\varphi_{pre}, \varphi_{post})$ requiring that
  - $\forall x \in R^n . x \vdash \varphi_{pre} \longrightarrow f(x) \vdash \varphi_{post}$



- Simplification to polyhedral input and output sets







$f_1(x) - f_2(x) \geq 0$

(for logits, m=2)

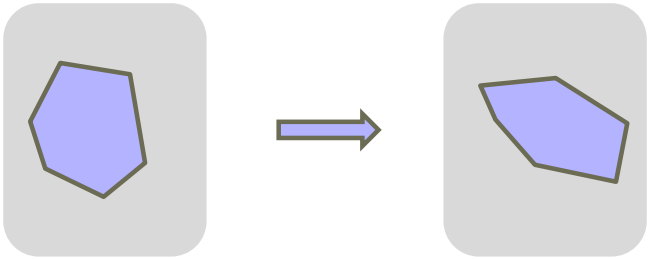
confidence gap

- Typically, exact verification intractable, focus on computing lower/upper bounds

Neural network verification: forward analysis

- Given a neural network $f: R^n \rightarrow R^m$, the NN verification problem is defined as $(\varphi_{pre}, \varphi_{post})$ requiring that
  - $\forall x \in R^n . x \vdash \varphi_{pre} \longrightarrow f(x) \vdash \varphi_{post}$

- Typical approach: forward analysis
  - start from $X = \{x \in R^n \mid x \vdash \varphi_{pre}\}$
  - bound the worst case on each layer
  - propagate bounds through layers
  - check whether the predicted labels are preserved

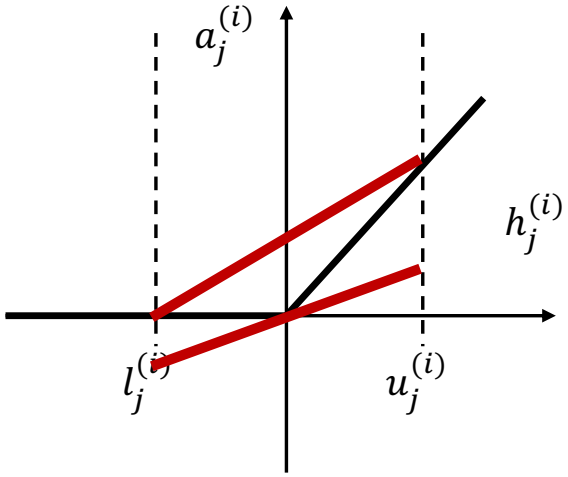


- Computes over-approximation of output set
- Note may result in loose bounds…

Progress in neural network verification

- Compute provable guarantees by lower/upper bounding the reachable values

- Methods include exact/approximate
  - search-based/Lipschitz, e.g. DLV
  - constraint solving/SMT/MIP, e.g., Reluplex
  - convex relaxation, e.g., interval/linear bound propagation, as in CROWN
  - abstract interpretation, e.g., DeepPoly
  - global optimisation, under assumption of Lipschitz continuity, e.g., DeepGO

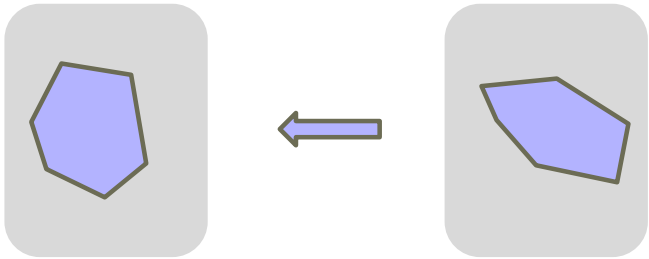


Linear bounding of ReLU activations
ReLU(x) := max(0, x)

- Hard problems, typically NP-completeness
- Convex relaxation best performers, see VNN-Comp
- Scaling, loose bounds and complex architectures an issue…

Neural network verification: backward analysis

- Given the NN verification problem $(\varphi_{pre}, \varphi_{post})$ for a neural network $f: R^n \rightarrow R^m$, requiring that
  - $\forall x \in R^n . x \vdash \varphi_{pre} \longrightarrow f(x) \vdash \varphi_{post}$
- Focus instead on backward analysis
- Characterize the inputs for output constraints $Y = \{y \in R^m \mid y \vdash \varphi_{post}\}$

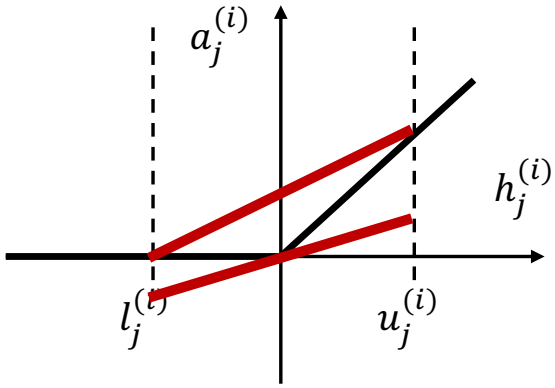


- Advantages
  - more precise correctness guarantees, particularly under-approximation
- but
  - exact preimage computation is intractable at scale, $O(2^n)$ for n unstable ReLU neurons

Provably bounding neural network preimages. Koha *et al*, In Proc. NeurIPS 2023.
Provable Preimage Under-Approximation for Neural Networks. Zhang *et al*, In Proc. TACAS 2024.

Preimage approximation

- Work backwards to generate preimage approximation via convex relaxation in terms of disjoint union of polytopes
- Given output specification $y = f(x) \geq 0$ (any polyhedral property)
- Compute symbolic lower/upper bounding functions for activations from output layer to input:
  - $\underline{A}x + \underline{b} \leq f(x) \leq \overline{A}x + \overline{b}$
- Preimage under-approximation as a polytope:
  - $\{x \mid \underline{A}x + \underline{b} \geq 0\} \longrightarrow \{x \mid f(x) \geq 0\}$
- Also over-approximation
- Method relies on
  - backward propagation
  - preimage refinement through input/ReLU splitting planes
  - heuristics and optimisations, to deal with exponential growth in constraints

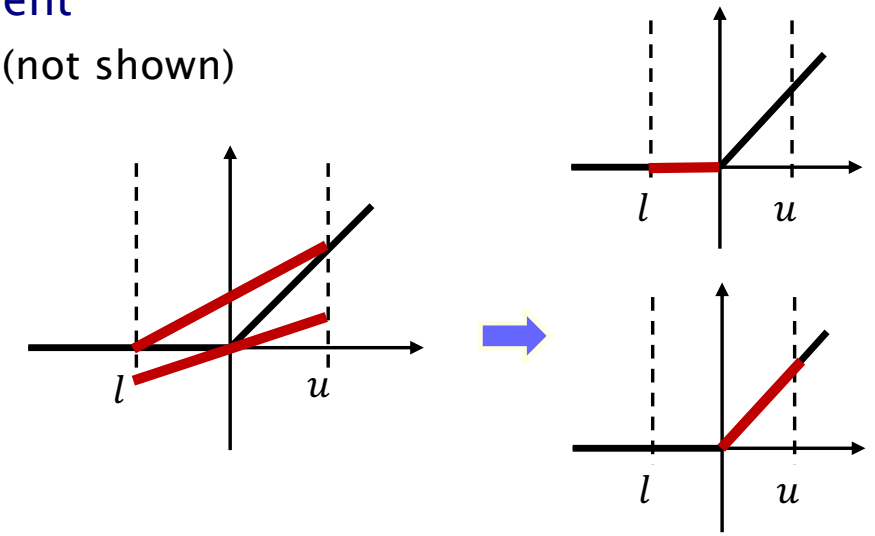


Linear bounding of ReLU activations

PREMAP: A Unifying PREiMage APproximation Framework for Neural Networks, arXiv:2408.09262

Preimage under/over-approximation

- Anytime algorithm, backward propagation via convex relaxation
- Preimage refinement to handle approximation loss
  - parallel processing of split regions
  - tightening of approximation by optimizing relaxation parameters
  - (novel differential objective)
- Two types of (sound) preimage refinement
  - via input-feature-aligned cutting plane (not shown)
  - via ReLU-aligned cutting plane
  - (unstable ReLU neuron into two stable cases: approximation becomes exact)
- Volume-estimated prioritization of splitting subregions
- Exact volume for final verification



PREMAP: A Unifying PREiMage APproximation Framework for Neural Networks, arXiv:2408.09262

Experimental results: preimage under-approximation

- Method scales to high-dimensional tasks
  - first method to scale to l_∞ attack (noise in all image pixels) and patch attack

  - evaluation on MNIST (GTSRB in progress) with varied size and position of the patch, indicating areas of vulnerability
  - provides quantitative coverage results for larger perturbation bounds

L_∞ attack	#Poly	Cov(%)	Time(s)	Patch attack	#Poly	Cov(%)	Time(s)
0.05	2	100.0	3.107	3×3(center)	1	100.0	2.611
0.07	247	75.2	121.661	4×4(center)	678	38.2	455.988
0.08	522	75.1	305.867	6×6(corner)	2	100.0	9.065
0.09	733	16.5	507.116	7×7(corner)	7	84.2	10.128

Provable Preimage Under-Approximation for Neural Networks. Zhang *et al*, In Proc. TACAS 2024.

Quantitative neural network verification

- Preimage under-approximation enables quantitative verification
  - i.e. estimating proportion of inputs that satisfy φ_{post}
  - sound and complete
- Useful in cases when verification fails
- Complementary to robustness verifiers, benchmarked against winner of VNN-Comp 2023

Task	α, β-CROWN		Our		
	Result	Time(s)	Cov(%)	#Poly	Time(s)
Cartpole ($\dot{\theta} \in [-1.642, -1.546]$)	yes	3.349	100.0	1	1.137
Cartpole ($\dot{\theta} \in [-1.642, 0]$)	no	6.927	94.9	2	3.632
MNIST (L_∞ 0.026)	yes	3.415	100.0	1	2.649
MNIST (L_∞ 0.04)	unknown	267.139	100.0	2	3.019

Provable Preimage Under-Approximation for Neural Networks. Zhang *et al*, In Proc. TACAS 2024.

Reachability for RL controllers

- Backward reachability analysis, with quantitative guarantees

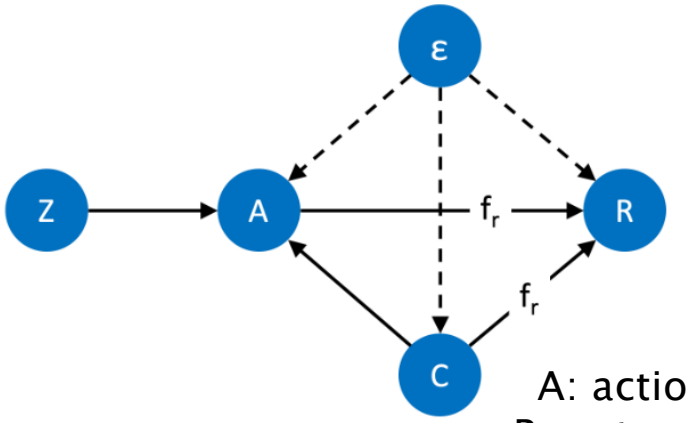
Task	Property	Config	#Poly		Cov		Time(s)	
			ux	ox	ux	ox	ux	ox
Cartpole (FNN 2×64)	$\{y \in \mathbb{R}^2 \mid y_1 \geq y_2\}$	$\dot{\theta} \in [-2, -1]$	25	1	0.766	1.213	13.337	2.149
		$\dot{\theta} \in [-2, -0.5]$	42	8	0.750	1.242	19.732	5.778
		$\dot{\theta} \in [-2, 0]$	66	22	0.755	1.246	30.563	11.476
Lunarlander (FNN 2×64)	$\{y \in \mathbb{R}^4 \mid \wedge_{i \in \{1,3,4\}}\ y_2 \geq y_i\}$	$\dot{v} \in [-1, 0]$	18	1	0.754	1.068	14.453	2.381
		$\dot{v} \in [-2, 0]$	67	23	0.751	1.246	48.455	19.210
		$\dot{v} \in [-4, 0]$	97	90	0.751	1.249	76.234	72.285
Dubinsrejoin (FNN 2×256)	$\{y \in \mathbb{R}^8 \mid \wedge_{i \in [2,4]}\ y_1 \geq y_i \bigwedge \wedge_{i \in [6,8]}\ y_5 \geq y_i\}$	$x_v \in [-0.1, 0.1]$	211	20	0.751	1.242	182.821	18.666
		$x_v \in [-0.2, 0.2]$	409	23	0.750	1.241	323.839	24.788
		$x_v \in [-0.3, 0.3]$	677	43	0.750	1.244	589.939	41.502

- Efficient, often bounding with few polytopes
- Over- and under-approximation

PREMAP: A Unifying PREiMage APproximation Framework for Neural Networks, arXiv:2408.09262

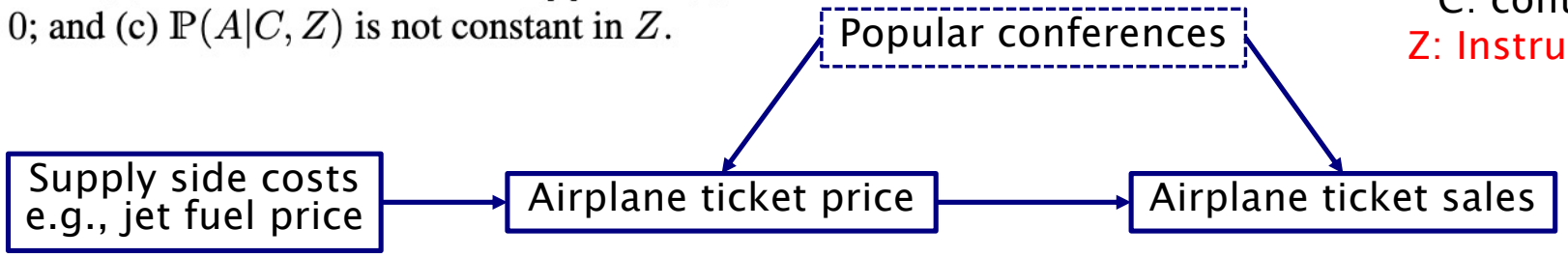
Decision policies with optimality guarantees

- The data generating model can be expressed as follows:
  $R := f_r(C, A) + \epsilon, \quad \mathbb{E}[\epsilon] = 0, \quad \mathbb{E}[\epsilon|A, C] \neq 0,$
- It has been shown that we cannot learn the causal effect of actions in the presence of hidden confounders without structural assumptions.
- Thus, we assume access to instrumental variables (IVs)



A: action
R: outcome
C: context
Z: Instrument

Assumption 2.1. (a) ϵ is additive to R and $\mathbb{E}[\epsilon] = 0$; (b) $\mathbb{E}[\epsilon|C, Z] = 0$; and (c) $\mathbb{P}(A|C, Z)$ is not constant in Z.



- Geographical distance from a medical facility can be an instrument for medical treatment

Learning Decision Policies With Instrumental Variables Through Double Machine Learning. Shao *et al*, In Proc. ICML 2024.

DML-IV

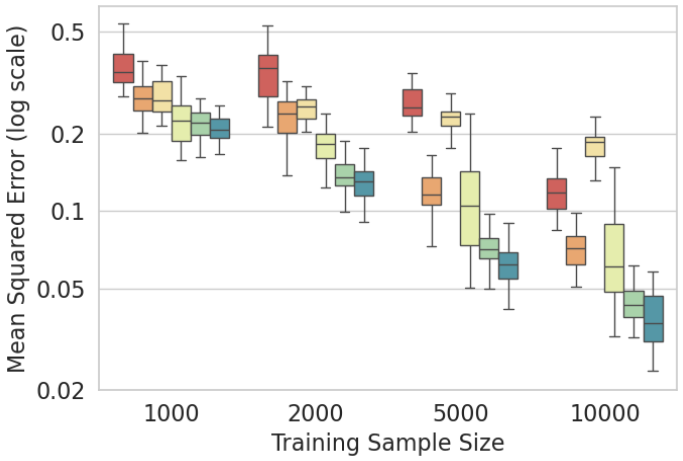
- Double Machine Learning (DML) is a statistical technique that debiases two-stage estimators and provides fast convergence rate guarantees of for general two-stage regressions, where N is the sample size
- We propose DML-IV, a novel IV regression algorithm utilising the DML framework
  - Derive a Neyman Orthogonal score function that describes the IV regression problem: $\psi(\mathcal{D}; h, (s, g)) = (s(c, z) - g(h, c, z))^2$
  - Design a k-fold cross-fitting learning algorithm.
- We are able to prove that the DML-IV estimator converges to the true causal effect function at rate $O(N^{-1/2})$ under mild regularity assumptions.
- Furthermore, the suboptimality of the induced decision policy is also $O(N^{-1/2})$

Learning Decision Policies With Instrumental Variables Through Double Machine Learning. Shao *et al*, In Proc. ICML 2024.

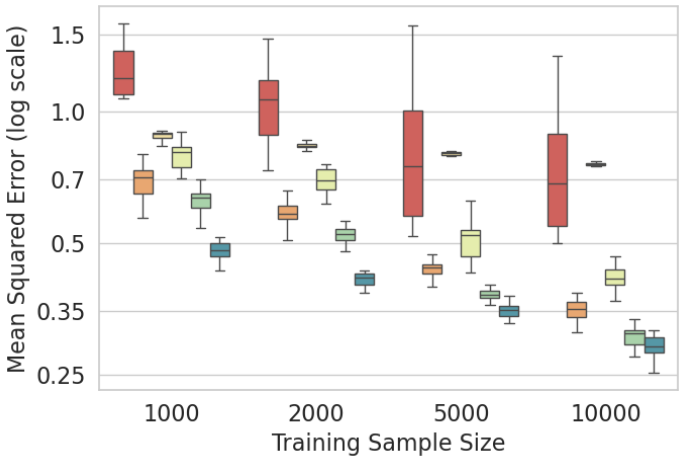
Experiments

- Evaluate DML-IV on benchmarks and semi-synthetic real-world datasets (Infant development and cardiovascular mortality rate datasets).
- Compare the error of the learned causal effect of actions (lower is better):

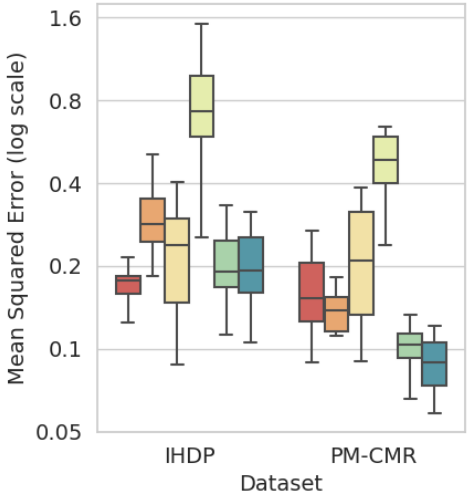
Low-dimensional



High-dimensional



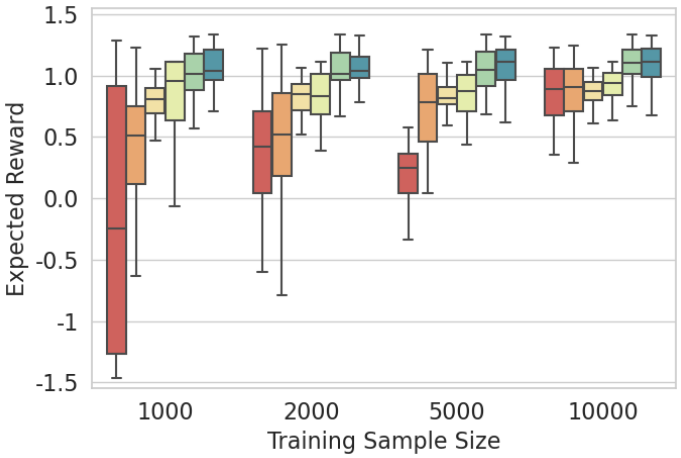
Real-World



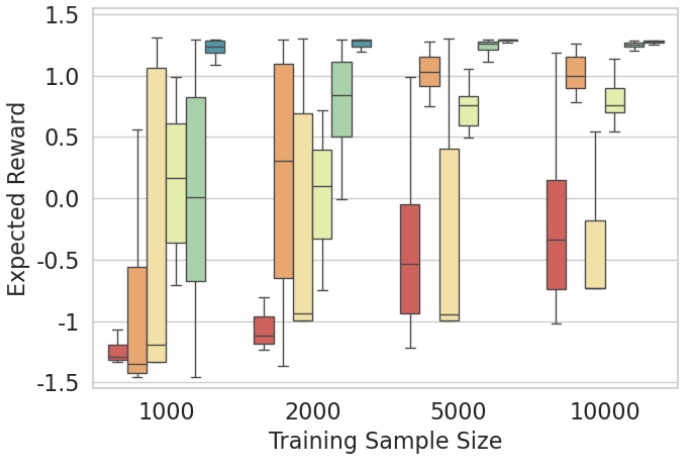
Experiments

- Compare the expected reward of the induced decision policy from the learned causal effect function (higher is better).

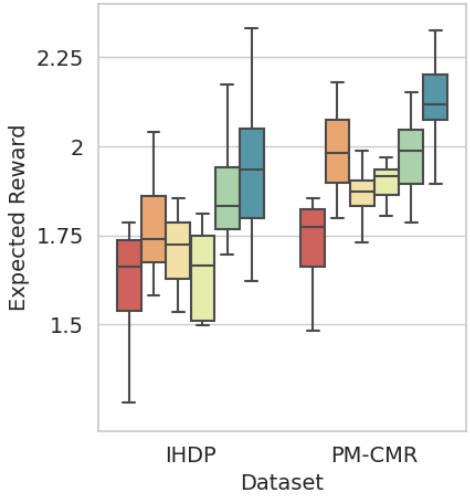
Low-dimensional



High-dimensional



Real-World



Extensions

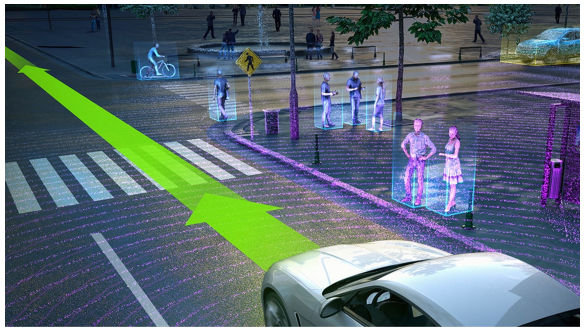
- Extension to imitation learning incorporating causal inference
  - A Unifying Framework for Causal Imitation Learning with Hidden Confounders. Shao *et al*, In *Workshop on Spurious Correlation and Shortcut Learning: Foundations and Solutions, Workshop at The International Conference on Learning Representations (ICLR) 2025*

- Learning of optimal policies from temporal logic specifications
  - Sample Efficient Model-free Reinforcement Learning from LTL Specifications with Optimality Guarantees. Shao *et al*, Proc. IJCAI 2023
  - Converts LTL to limit-deterministic Buchi automata

- Learning temporal logic specifications to debug/explain RL policies
  - Learning Probabilistic Temporal Logic Specifications for Stochastic Systems. Roy *et al*, Proc. IJCAI 2025
  - Learns concise probabilistic LTL from positive and negative examples

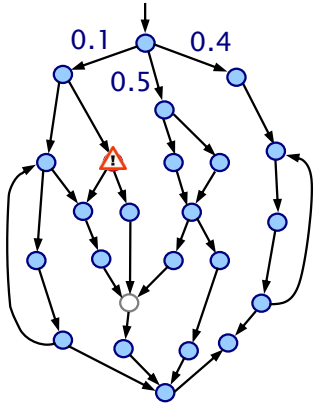
But what about uncertainty?

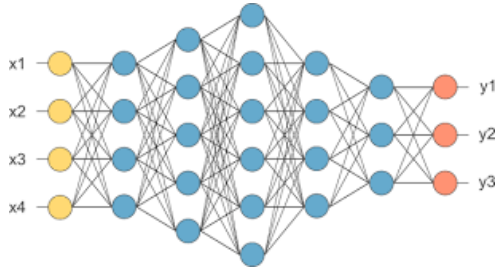
- Autonomy, yet highly uncertain scenarios!



- Main focus so far on deterministic neural networks
  - deterministic outcomes, fixed (trained) weights
  - potentially overconfident predictions

- Probabilistic verification
  - enables reasoning, planning, etc, in the presence of uncertainty
  - many tools (PRISM, Storm, etc)
  - limited to state-based models





- Can we extend probabilistic verification to neural network settings?

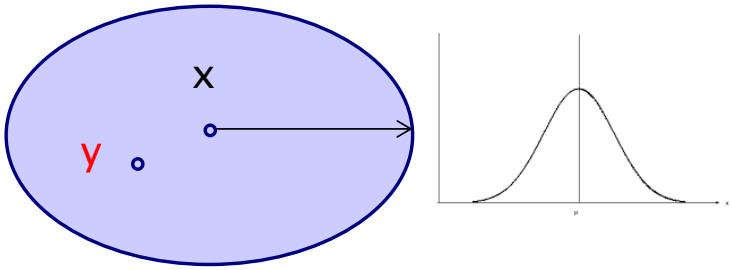
Probabilistic Model Checking for Strategic Equilibria-based Decision Making: Advances and Challenges, Kwiatkowska et al., In *Proc* MFCS 2022

Uncertainty in decision making

- Requiring that no adversarial examples exist too strict!
- Uncertainty as a first-class citizen: Bayesian neural networks (BNNs)
  - allow for 'don't know' answers, can increase trust in decisions
  - pass distributions through softmax

- Define safety with prob $1-\varepsilon$

  $Prob(\exists y \in \eta \text{ s.t. } f(x) \neq f(y) \mid D) \leq \varepsilon$
- i.e., conditioned on training data D

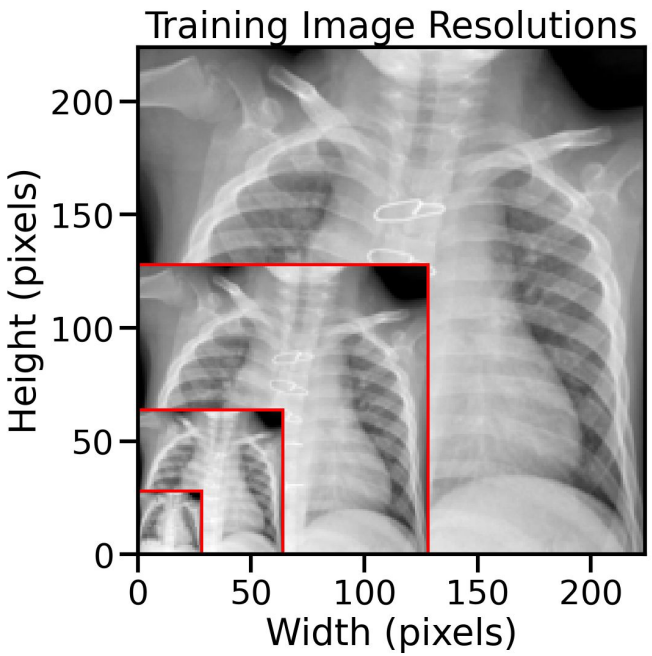


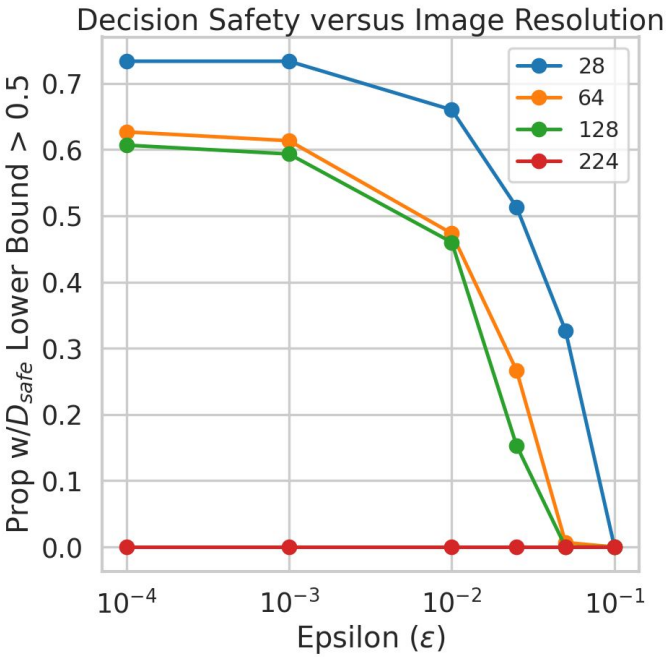
- Aim to provide provable probabilistic guarantees for BNNs
  - certified bounds on decision probability
  - also under adverse conditions
  - (certifiable adversarial robustness training)

Statistical Guarantees for the Robustness of Bayesian Neural Networks. Cardelli *et al*, In Proc. IJCAI 2019.
Probabilistic Safety for Bayesian Neural Networks, Wicker et al., In *Proc* UAI 2020

Certification of Bayesian CNN on medical images

- MedMNIST (PneumoniaMNIST)
- Decision safety
- Certified robust (lower bound on decision safety), vs image resolution



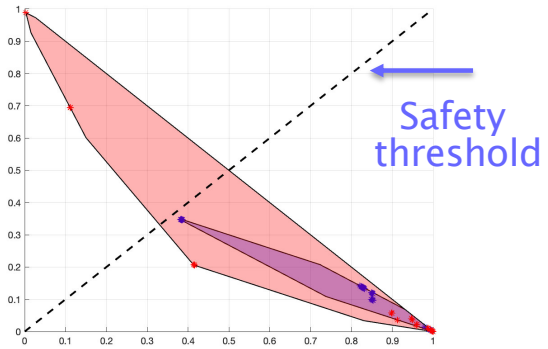


- Limited scalability, more progress needed!
- Also need stronger methodologies beyond certification: correctness by design

Adversarial Robustness Certification for Bayesian Neural Networks. FM 2024

Risk-averse certification for BNNs

- Averaging over full input distributions does not suffice
- Risk-averse certification: robustness under adverse conditions, say 5% of most adversarially unstable cases
- Principled approach incorporating CVar (Conditional Value at Risk)
- Compute certified CVar bounds with probabilistic guarantees



Output support set (empirical)

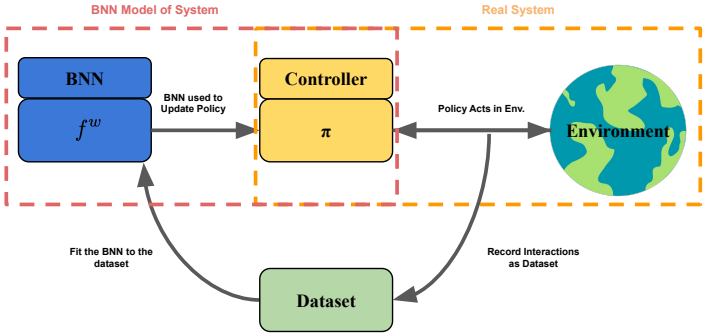
Table 2: Certified CVaR bounds for different attacks on classification tasks.

Tasks	CVaR Level	L_∞ noise	Rotation	Contrast
MNIST	$\alpha = 1$	-0.999 $\pm$ 0.1	-0.998 $\pm$ 0.1	-0.997 $\pm$ 0.1
	$\alpha = 0.5$	-0.999 $\pm$ 0.1	-0.998 $\pm$ 0.1	-0.993 $\pm$ 0.1
	$\alpha = 0.25$	-0.999 $\pm$ 0.1	-0.997 $\pm$ 0.1	-0.986 $\pm$ 0.1
FASHION	$\alpha = 1$	-0.191 $\pm$ 0.1	-0.119 $\pm$ 0.1	-0.152 $\pm$ 0.1
	$\alpha = 0.5$	0.443 $\pm$ 0.1	0.525 $\pm$ 0.1	0.562 $\pm$ 0.1
	$\alpha = 0.25$	0.885 $\pm$ 0.1	0.964 $\pm$ 0.1	0.849 $\pm$ 0.1

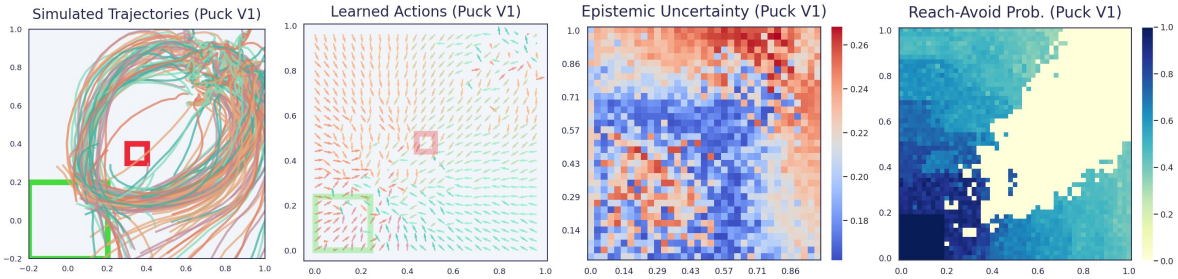
Risk-Averse Certification of Bayesian Neural Networks. arXiv:2411.19729

Probabilistic guarantees for autonomous controllers

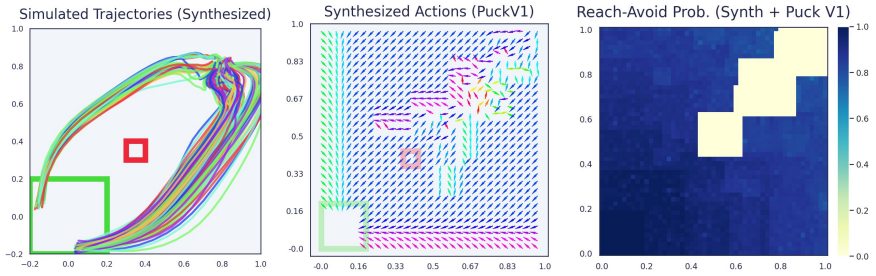
- So far only one-step predictions; now consider iterated prediction steps
- Here, environment specified as a Bayesian neural network



Variational Posterior



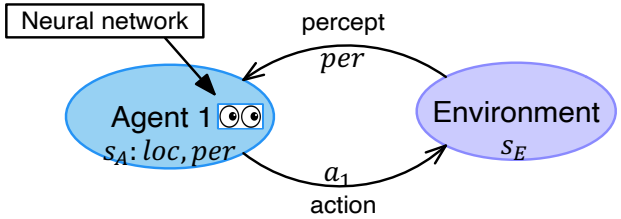
- Synthesised controllers and reach-avoid probabilities
  - controller benchmarks
  - obstacles



Probabilistic reach-avoid for Bayesian Neural Networks. Wicker *et al*, Artificial Intelligence, 2024.

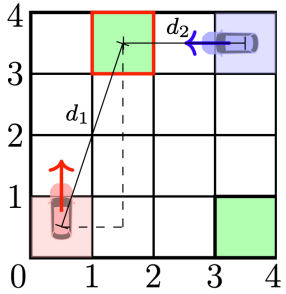
Neuro-symbolic games

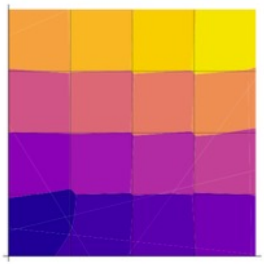
- Agents endowed with neural perception and symbolic decision making
  - here: NN classifiers (or other machine learning) for perception tasks
  - constrained interface: convert inputs such as images to symbolic percepts
  - plus: local strategies for control decisions



- Neuro-symbolic games (two players/coalitions)
  - finite-state agents + continuous-state environment E
    - $S = (Loc_1 \times Per_1) \times (Loc_2 \times Per_2) \times S_E$
  - agents only use a (learnt) perception function to observe E
    - $obs_i : (Loc_1 \times Loc_2) \times S_E \rightarrow Per_i$
  - joint actions update state probabilistically

- Example: dynamic vehicle parking
  - NN maps exact vehicle position to perceived grid cell
  - stochasticity from, e.g., motion imprecision

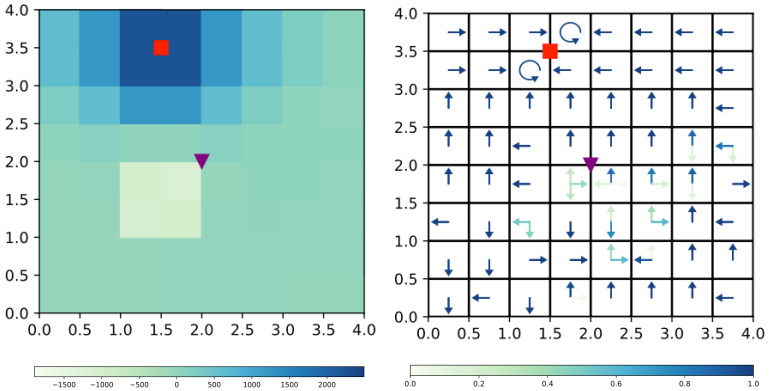




Strategy synthesis for neuro-symbolic games

- Consider zero-sum (discounted) expected reward over infinite horizon
  - for now, we assume full observability
  - value exists under Borel assumptions, fixed point of minimax
  - but optimal value may not be finitely representable
- Value iteration (VI) approach, exploit structure
  - continuous state-space decomposed into regions with the same percept (and reward)
  - further subdivision at each iteration via Borel decomposition, under assumptions
  - abstraction based on piecewise-continuous value functions, preserved by NNs and VI
- Implementation
  - pre-image computations of NNs
  - polytope representations of regions (ReLU)
  - LPs to solve zero-sum games at each step

Dynamic vehicle parking with larger (8x8) grid and simpler (regression) perception



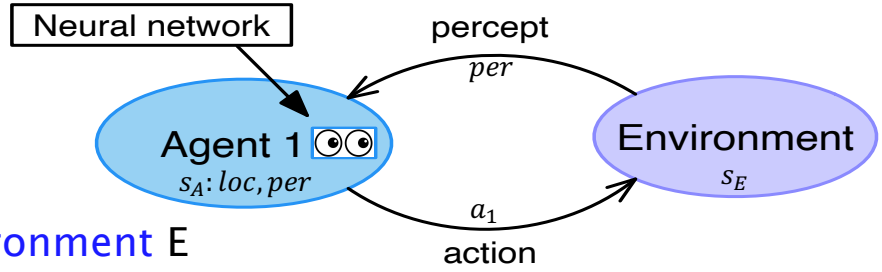
Value function (fragment)

Optimal strategy (fragment)

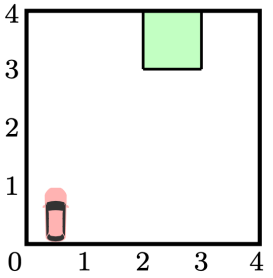
Strategy synthesis for zero-sum neuro-symbolic concurrent stochastic games, Inf & Comp, 2024

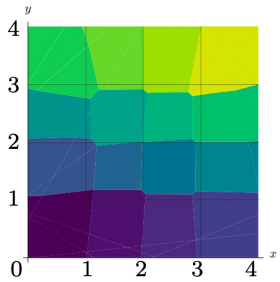
Neuro-symbolic POMDPs

- Need partial observability for neural perception, not just continuous environment!



- NS-POMDP (partially observable MDP)
  - finite-state agent + continuous-state environment E
    - S = (Loc×Per) × S_E
  - agent uses a (trained) perception function to observe E
    - obs : (Loc×S_E) → Per
  - and transitions based on local state and percept
    - δ : (Loc×Per×A) → Dist(Loc)
- Work in belief space over S (probability distributions)

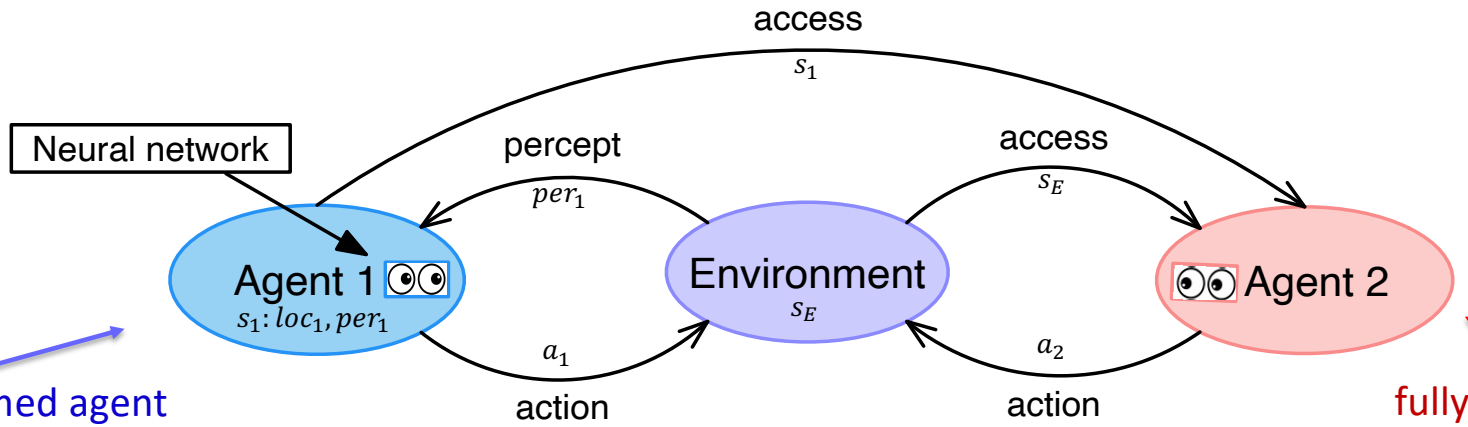




Car parking (4x4)
Localisation NN

Neuro-symbolic POSGs

- Restrict to one-sided variant, new subclass of hybrid-state POSGs



Partially informed agent
maximize

fully informed agent
minimize

NS-POSGs (same syntax as NS-CSGs)

- finite-state agents + continuous-state environment E
- Agent 1 uses a (learnt) perception function to observe E
  - $obs_1 : (Loc_1 \times S_E) \rightarrow Per_1$
- and transitions on percepts and local states $\delta : (Loc_1 \times Per_1 \times A) \rightarrow Dist(Loc)$
- Agent 2 fully informed

discounted reward:

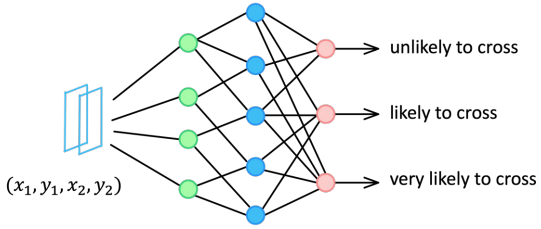
$$Y(\pi) = \sum_{k=0}^{\infty} \beta^k r(\pi(k), \pi[k])$$

zero-sum

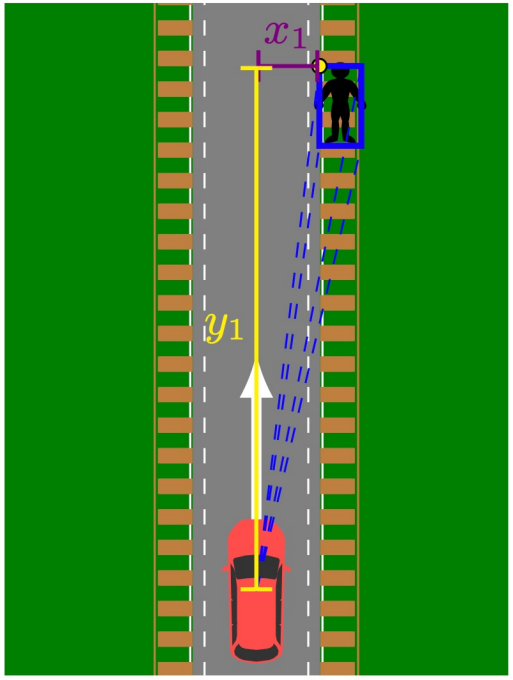
Partially observable stochastic games with neural perception mechanisms, In *Proc* FM 2024

Example: pedestrian-vehicle interaction

- Autonomous vehicle
  - partially informed
  - aims to predict pedestrian's intention
  - using NN trained from video data

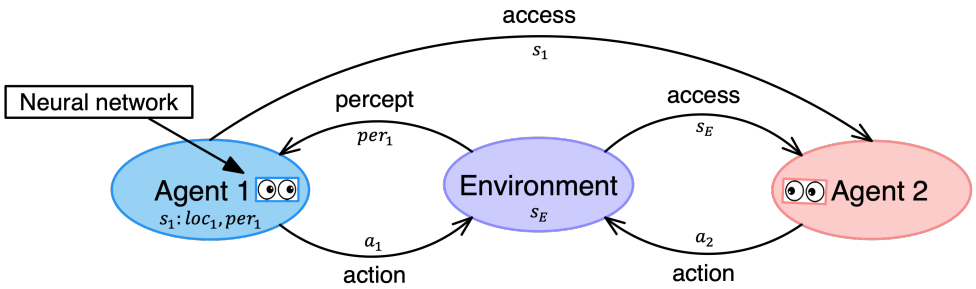


- Pedestrian
  - fully informed for worst-case analysis
  - decides whether to cross or return to sidewalk
- Goal: synthesise strategy for vehicle to minimize likelihood of crash (opposite for pedestrian)

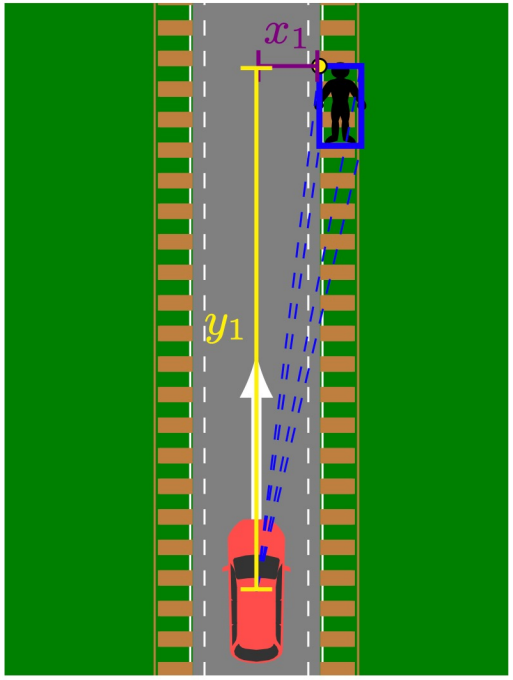


https://data.nvision2.eecs.yorku.ca/PIE_dataset/

Pedestrian–vehicle interaction as NS–POSG

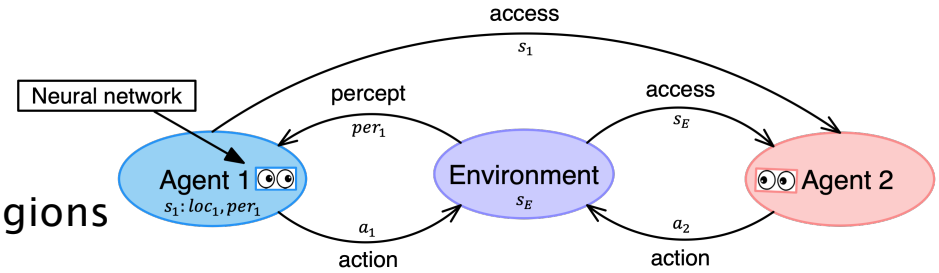


- Agent 1: vehicle
  - loc_1: speed
    per_1: pedestrian intention
  - a_1: acceleration (e.g. +3,–3)
- Agent 2: pedestrian
  - a_2: cross, back
- Environment E
  - two successive pedestrian positions
    (x_1, y_1, x_2, y_2)



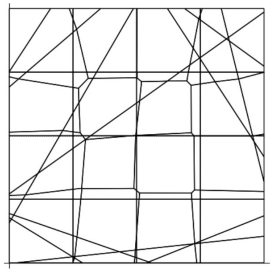
Strategy synthesis for neuro-symbolic POSGs

- Consider zero-sum (discounted) expected reward over infinite horizon
  - one sided, so Agent 2 can recover beliefs of Agent 1
  - assume determined, as value may not exist



- HSVI approach (extend Horak *et al* 2023)
  - continuous state-space decomposed into regions
  - further subdivision at each iteration
  - work with a class of piecewise-continuous α-functions, + closure properties
  - anytime

PWC α-function
polyhedra + value vector

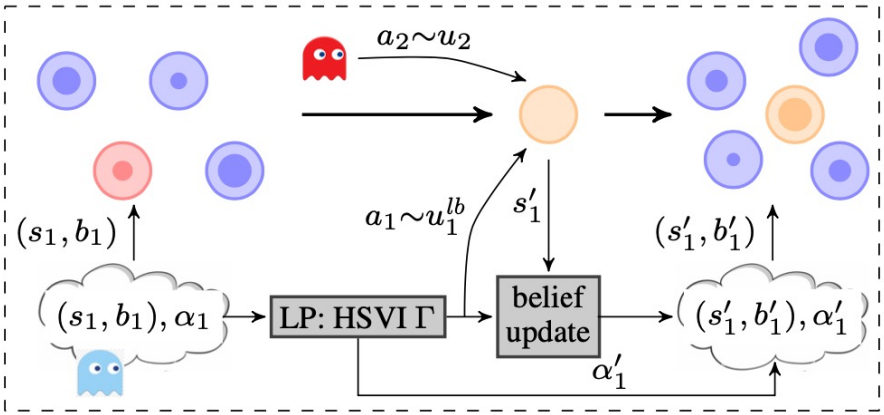


- Implementation
  - polyhedral pre-image computations of NNs
  - LPs to compute lower/upper bound and minimax values

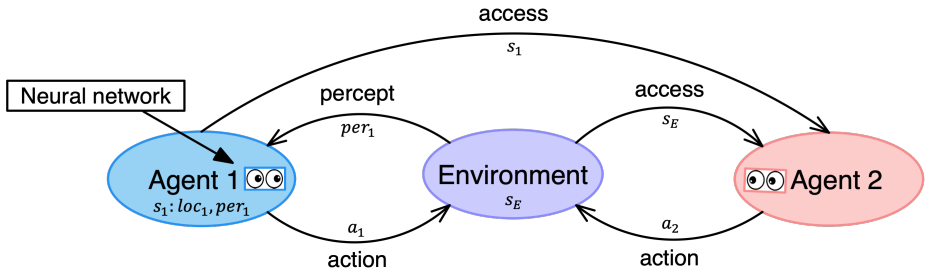
Partially observable stochastic games with neural perception mechanisms, In *Proc* FM 2024

Efficient online minimax strategies

- How to synthesize strategies based on the lower and upper bound functions



NS-HSVI continual re-solving for Ag_1



Online continual resolving

- keeps track of belief and counterfactual values
- builds and solves a game without storing complete strategy

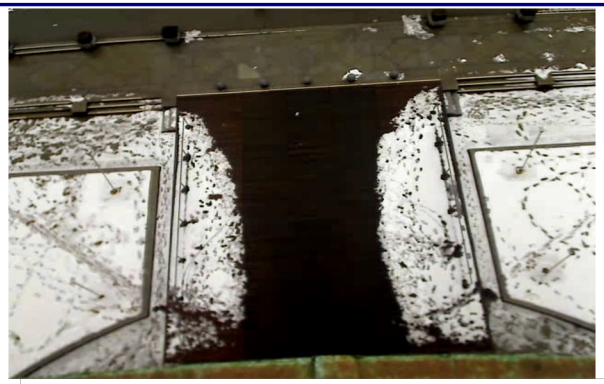
Our variant

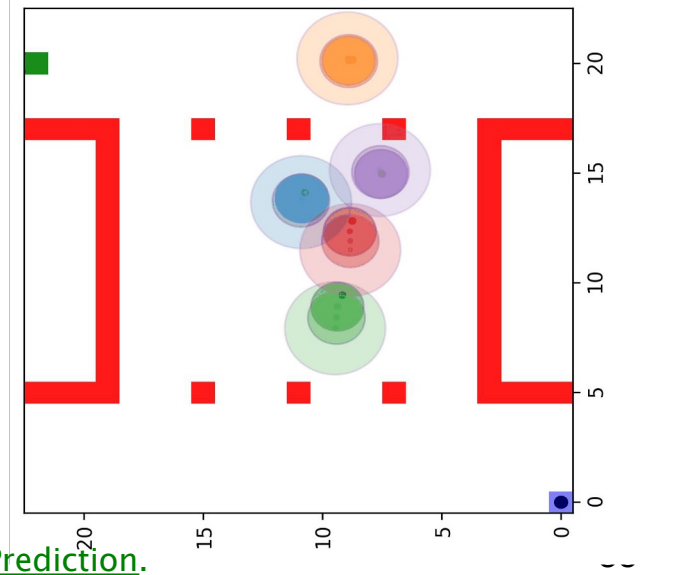
- precomputes HSVI lower bound
- keeps track of belief and PWC function α_1
- solves a single LP at each stage

HSVI-based Online Minimax Strategies for Partially Observable Stochastic Games with Neural Perception Mechanisms, In *Proc* L4DC 2024

Safe planning in a crowd

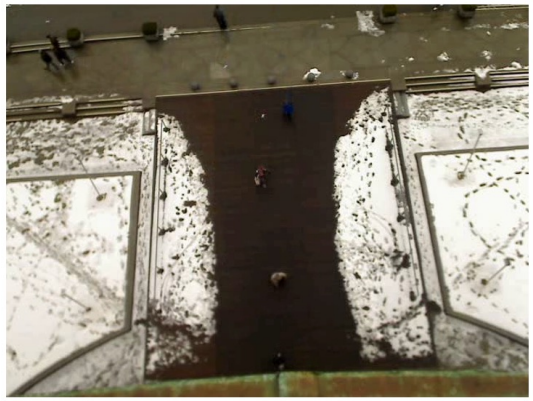
- Robotic agent modelled as a partially observable MDP (POMDP)
  - partial observability (e.g., perception inaccuracy)
- Environment is populated with pedestrians
- Pedestrian trajectory prediction
  - data-driven trajectory predictor
  - uncertainty quantification via (statistical) adaptive conformal prediction (ACP)
- Safe online planning via shielding
  - on-the-fly safety shield construction
- Safety guarantee, given any probability threshold
  - outperforms state of the art on real data





Safe POMDP Online Planning Among Dynamic Agents via Adaptive Comformal Prediction.
Sheng *et al*, RAL 2024.

Experimental evaluation using real-world data



(a) ETH

(b) Hotel

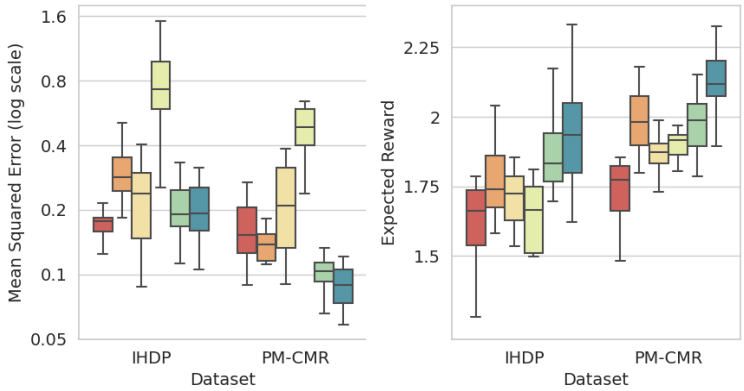
(c) GC

TABLE I: Experiment Results

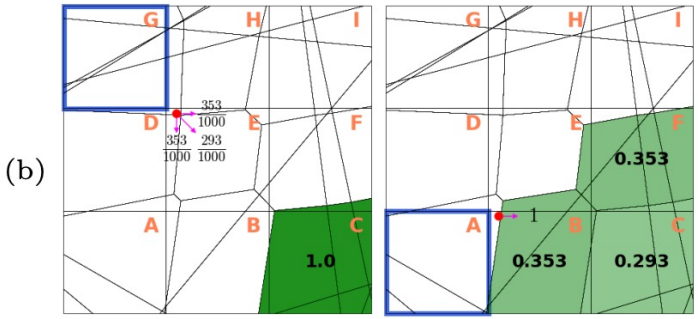
	ETH				Hotel				GC			
Method	N	Safety Rate	Time (s)	Min Distance	N	Safety Rate	Time (s)	Min Distance	N	Safety Rate	Time (s)	Min Distance
No Shield		0.893	21.1	0.28±0.19		0.944	20.1	0.42±0.21		0.91	39.3	0.22±0.13
Shielding without ACP	45	0.943	21.5	0.39±0.29	35	0.969	20.3	0.54±0.3	160	0.943	67.2	0.23±0.13
Shielding with ACP		**0.974**	22.1	0.51±0.29		**0.988**	20.6	0.8±0.49		**0.963**	71.4	0.28±0.16
No Shield		0.891	21.0	0.26±0.17		0.931	20.1	0.38±0.24		0.904	39.9	0.2±0.1
Shielding without ACP	55	0.951	21.8	0.41±0.25	45	0.959	20.1	0.48±0.24	180	0.938	66.8	0.23±0.12
Shielding with ACP		**0.975**	22.4	0.53±0.37		**0.982**	20.6	0.62±0.27		**0.953**	71.1	0.24±0.16
No Shield		0.872	21.2	0.24±0.13		0.921	20.2	0.36±0.18		0.895	39.6	0.22±0.11
Shielding without ACP	65	0.943	21.9	0.36±0.2	55	0.957	20.3	0.48±0.29	200	0.931	65.7	0.2±0.13
Shielding with ACP		**0.967**	22.6	0.42±0.26		**0.982**	20.3	0.6±0.24		**0.951**	74.3	0.25±0.15

Safe POMDP Online Planning Among Dynamic Agents via Adaptive Comformal Prediction.
Sheng *et al*, RAL 2024.

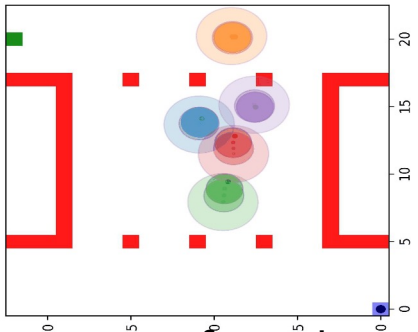
Multiple applications and NN verification use cases!



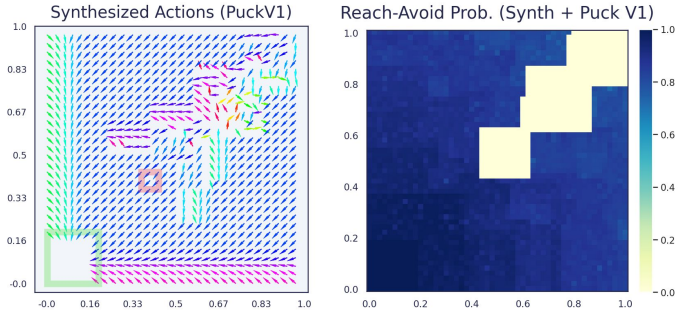
Efficient policy learning



Protocol verification



Dynamic conformal prediction



Optimal controller synthesis

http://fun2model.org/

Beyond certification: robust learning

- So far, consider trained neural networks, need to retrain if verification fails
- Can we instead efficiently robustly learn? (correct-by-construction synthesis)

- Similar to PAC framework (polynomial sample complexity), except
  - for concept c and hypothesis h, use robust risk $ProbR(\exists z \in \eta(x,\varrho)$ s.t. $c(z) \neq h(z))$
  - instead of standard risk $Prob(c(x) \neq h(x))$
  - NB learning is exact in region η, different from previously required invariance/stability over η (constant)
- Show that no non-trivial concept can be learnt in the distribution-free setting

- For simple concepts, can efficiently ϱ-robustly learn under classes of distributions (e.g., uniform)

On the Hardness of Robust Classification. Gourdeau *et al*, In Proc. *NeurIPS 2019*, extended *JMLR, 22(273)* 2021
When are Local Queries Useful for Robust Learning? Gourdeau *et al*, In Proc. *NeurIPS 2022*

Beyond adversaries: strategyproof robustness

- So far, consider only adversarial robustness to individual perturbations, but AI agents can behave strategically
- Can we instead strategyproof policy learning? (correctness by design)

- Consider RLHF (reinforcement learning from human feedback)
  - multiple agents, diverse preferences, leading to potential bias in learnt policy decisions
  - but agents can also strategically manipulate the decisions in their favour by misreporting their preferences
  - existing RLHF methods not strategyproof…

- Aim to devise strategyproof RLHF through mechanism design
  - how? incentivise truthful reporting
  - can provide an algorithm that is approximately strategyproof and converges to the optimal policy as the number of individuals and samples increases

Strategyproof Reinforcement Learning from Human Feedback. Kleine Buening *et al*, arXiv:2503.09561v1

Concluding remarks

- Range of techniques developed in the AI/ML and formal methods communities
  - robustness guarantees needed for high-stakes decisions
  - optimality, explainability of policies desirable
  - but likely to need human involvement in decisions and act as assistants
  - ML models increasing in complexity, take up of certification lagging behind
- Despite progress, major challenges remain
  - scalability to complex architectures and properties
  - foundational understanding needed
  - ideally, semantic methods, not pixel-based perturbations
  - need support for interactions with human decision makers
  - robust learning for correct-by-construction models and policies
- Need integrated processes for validation and safety assurance, not just (probabilistic) verification

Acknowledgements

- My group and collaborators in this work

- Project funding
  - ERC Advanced Grant fun2model
  - EPSRC project FAIR: Framework for responsible adoption of artificial intelligence in the financial services industry, https://www.turing.ac.uk/research/research-projects/project-fair-framework-responsible-adoption-artificial-intelligence
  - ELSA European Lighthouse on Secure and Safe AI, https://www.elsa-ai.eu/
  - UKRI AI Hub on Mathematical foundations of intelligence: an 'Erlangen Programme' for AI

- See also
  - PRISM www.prismmodelchecker.org