DSN, Naples 25th June 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

- AI decisions rely on neural network components
- Well known that neural networks are unstable to adversarial perturbations

Physical attack

Lightbeam 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?

Safety Verification of Deep Neural Networks. CAV 2017 keynote

Michael Passingham
Senior researcher & writer

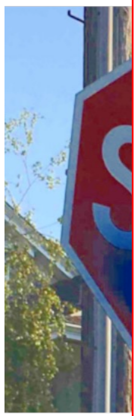
New research from Which? reveals that more than half of drivers are turning off safety tech in their cars with many finding the tech annoying, distracting or even dangerous. We explain why, and how to get the best from your current car or your next purchase.

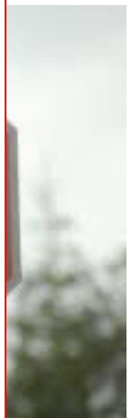
Like airbags and crumple zones, various car safety technologies are mandatory on new cars. While airbags are considered 'passive' safety tech – they only activate when you crash – Advanced Driver-Assistance Systems (ADAS) are 'active' and are intended to prevent you from having an accident in the first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





Phy

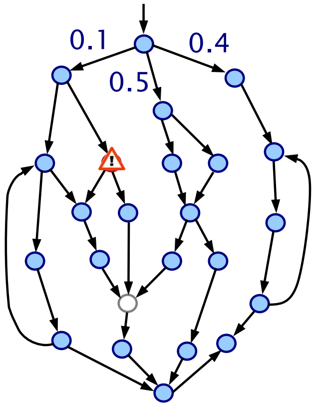
Safe

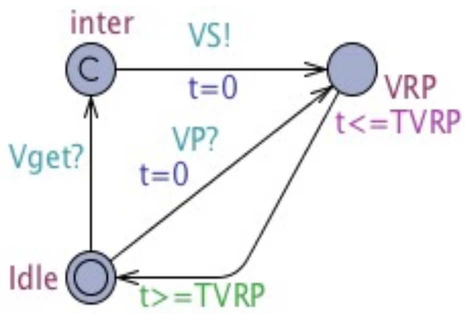
Formal verification provides provable guarantees

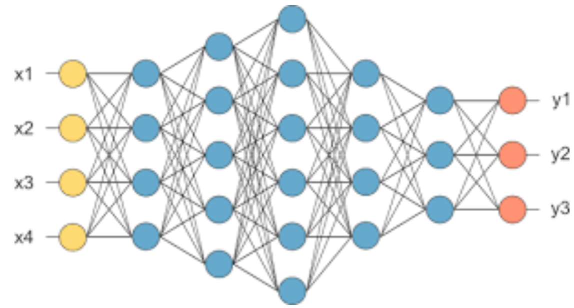
```
void add(Object o) {
  buffer[head] = o;
 head = (head+1)%size;
}

Object take() {
 …
 tail=(tail+1)%size;
 return buffer[tail];
}
```

Software





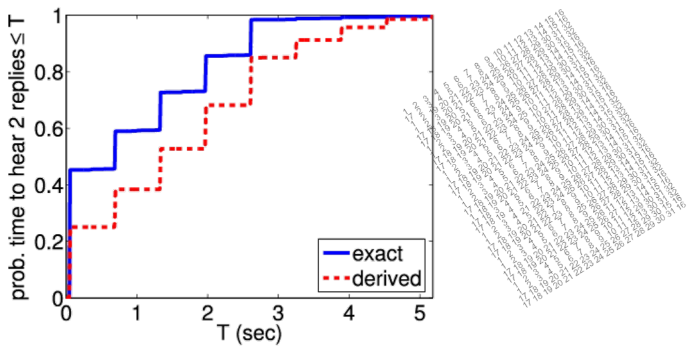


Model

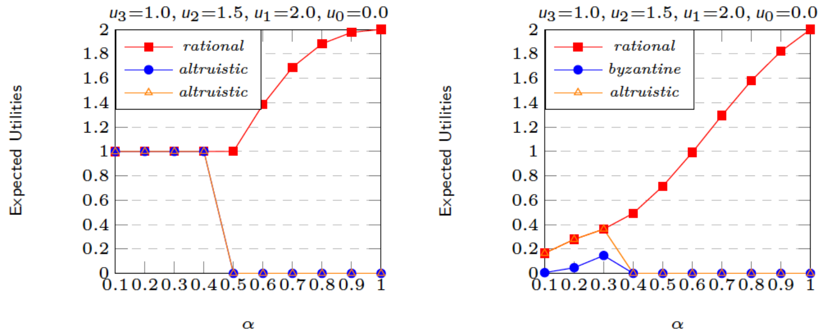
- Modelling = rigorous, mathematical abstraction
- Verification = proof that the model satisfies specification
- Synthesis = correct-by-construction model/policy from specification
- Automated = algorithmic, implemented in software

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

Multiple applications and use cases!



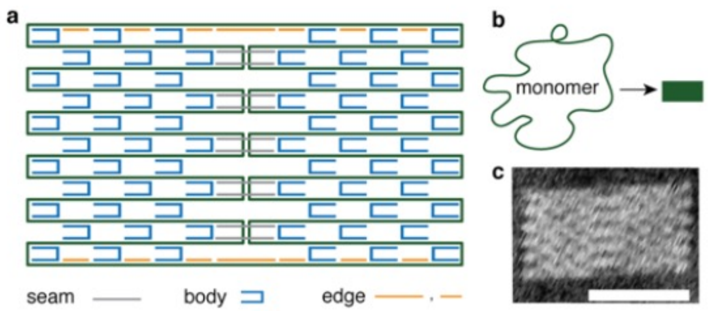
Protocol debugging



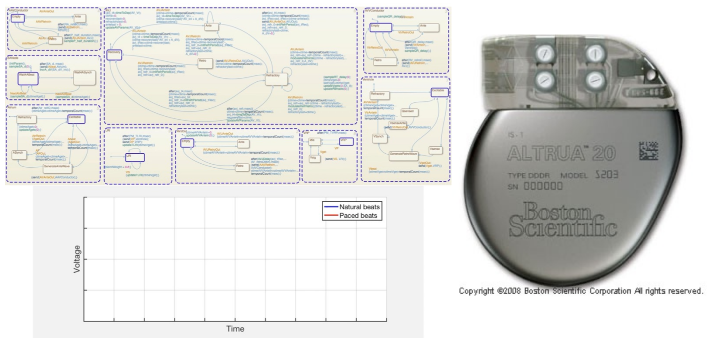
((a)) 2 altruistic and 1 rational

((b)) 1 altruistic, 1 byzantine and 1 rational

Protocol verification



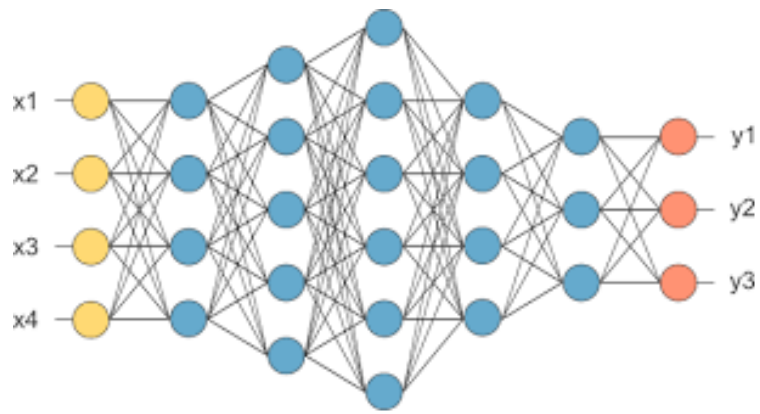
Prediction of DNA folding



Optimal controller synthesis

Formal verification for neural networks (NNs)

- Rigorous formal verification
  - can provide provable guarantees, e.g. that no adversarial examples exist
  - enables safety/security 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, DeepPoly, ReluVal, CROWN, …

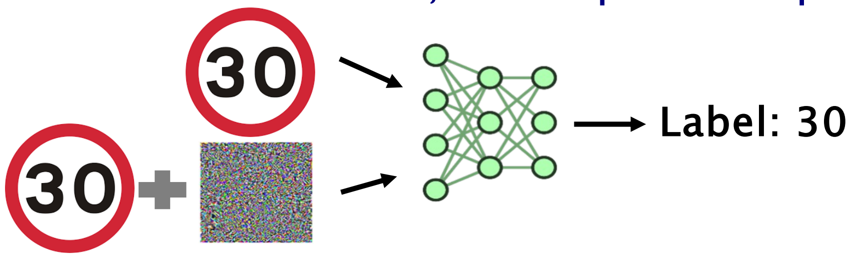
Safety Verification of Deep Neural Networks. CAV 2017 keynote

This talk: provable guarantees via formal verification

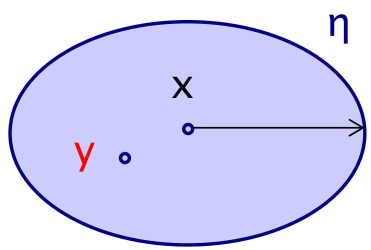
- Focus on (data-driven) neural network policies and components
- Brief recap of (local) adversarial robustness certification
  - crucial part of safety assurance, pre-deployment
  - and many more use cases
- A selection of snapshots, with a common thread
  - pre-image approximation
  - quantitative verification
  - backward reachability for controllers
  - neuro-symbolic models
- Conclusions and future directions

Recap of adversarial robustness

- Consider local adversarial robustness, for a specific input



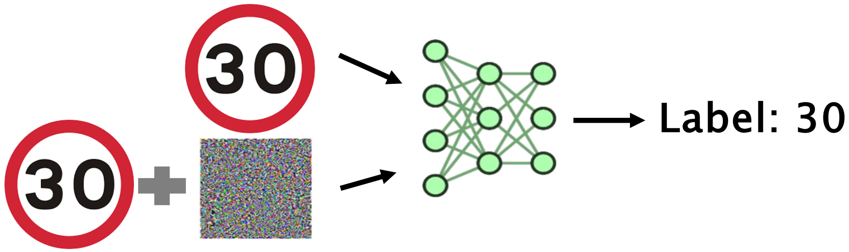
- Informally, no perturbation results in a misclassification



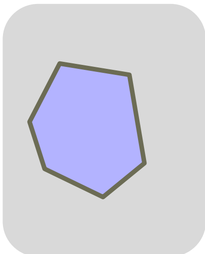
- More formally, assume given
  - trained neural network classifier $f : R^m \rightarrow \{c_1, \ldots c_k\}$
  - region η centred at x wrt distance function, e.g. L^2, L^∞
- Define local robustness at x wrt η by (SAT friendly)
  - $\nexists y \in \eta$ such that $f(x) \neq f(y)$
- Here, focus on computing provable guarantees on correctness, rather than constructing defences

Neural network verification

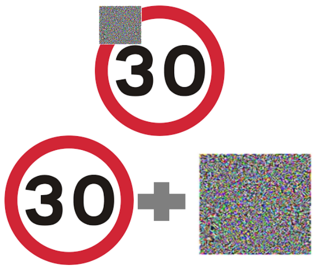
- Given a neural network $f: R^n \rightarrow R^m$, the NN verification problem is defined as $(\varphi_{pre}, \varphi_{post})$ requiring that
  - $\forall x \in R^n . x \vdash \varphi_{pre} \longrightarrow f(x) \vdash \varphi_{post}$

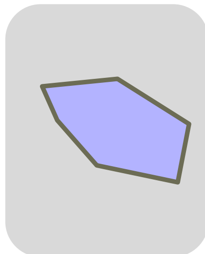


- Simplification to polyhedral input and output sets

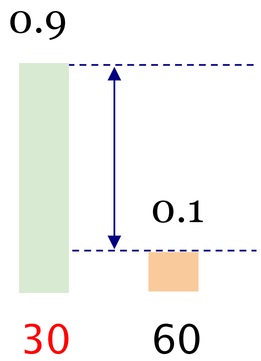


Input





Output



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

(for logits, m=2)

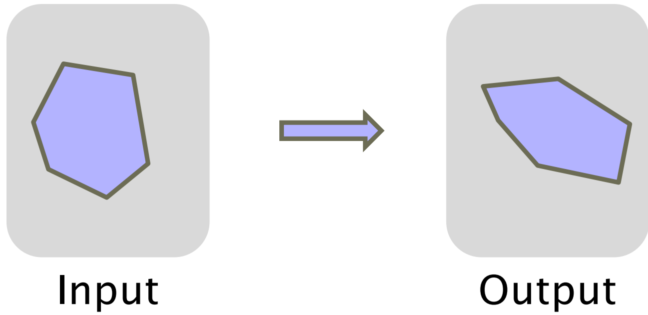
confidence gap

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

Neural network verification: forward analysis

- Given a neural network $f\colon R^n \rightarrow R^m$, 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

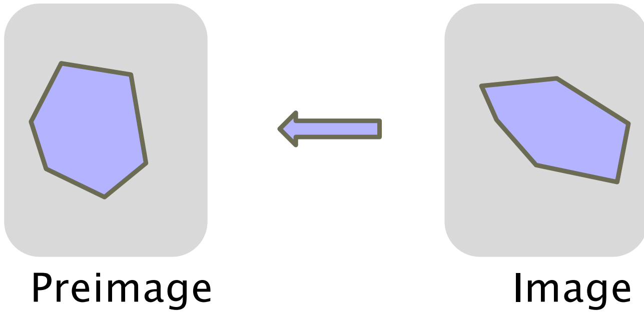


- Computes over-approximation of output set
- Note may result in loose bounds…

Neural network verification: backward analysis

- Given the NN verification problem $(\varphi_{pre}, \varphi_{post})$ for a neural network $f: R^n \to R^m$, 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 $\mathrm{Y} = \{y \in R^m \mid y \vdash \varphi_{post}\}$

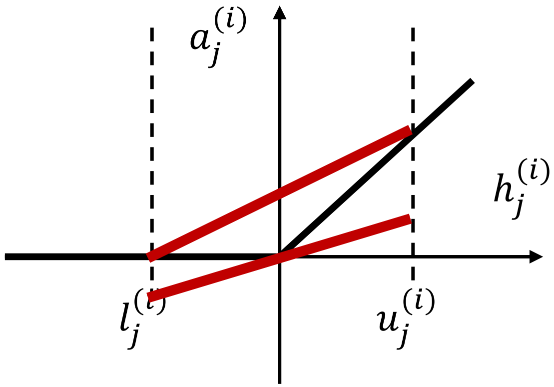


- Advantages
  - more precise correctness guarantees, particularly under-approximation
- but
  - exact preimage computation is intractable at scale, $O(2^n)$ for n unstable ReLU neurons

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 preimage over-approximation
- Method relies on
  - backward propagation
  - preimage refinement through input/ReLU splitting planes
  - heuristics and optimisations, to deal with exponential growth in constraints

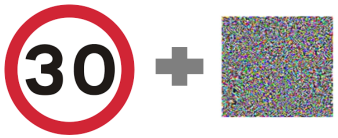


Linear bounding of ReLU activations

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

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**
  - recently improved and extended to CNNs



  - evaluated on MNIST, GTSRB and SVHN 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

Efficient Preimage Approximation for Neural Network Certification. Bjorklund *et al*, arxiv.org/abs/2505.22798

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.

Backward reachability for controllers

- Provable quantitative guarantees for reinforcement learning (RL) controllers
- via preimage over- and under-approximation

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

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

Provable guarantees for RL decision policies

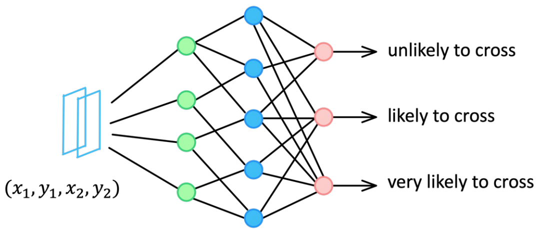
- Learning of optimal policies from temporal logic specifications
  - Sample Efficient Model-free Reinforcement Learning from LTL Specifications with Optimality Guarantees. Shao *et al*, Proc. IJCAI 2023
  - Converts LTL to limit-deterministic Buchi automata

- 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

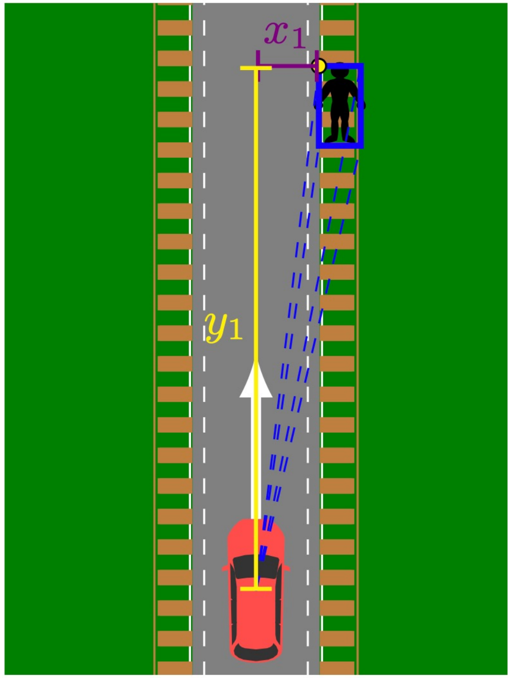
- Extension to imitation learning incorporating causal inference
  - A Unifying Framework for Causal Imitation Learning with Hidden Confounders. Shao *et al*, In *ICLR 2025 Workshop on Spurious Correlation and Shortcut Learning: Foundations and Solutions*
  - Learns causal effects using instrumental variables

Motivating 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

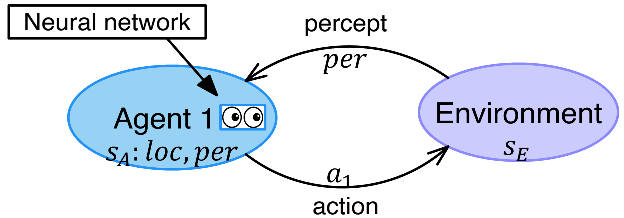


- Goal: synthesise strategy for vehicle to minimize likelihood of crash (opposite for pedestrian)

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

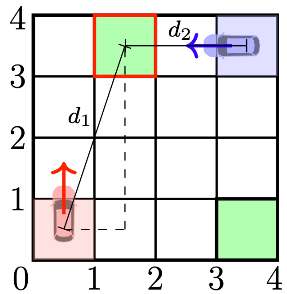
Neuro-symbolic games (NS-POSGs)

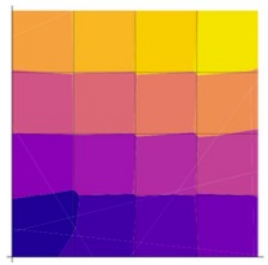
- Agents endowed with neural perception and symbolic decision making
  - here: NN classifiers (or other machine learning) for perception tasks
  - constrained interface: convert inputs such as images to symbolic percepts
  - plus: local strategies for control decisions



- 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

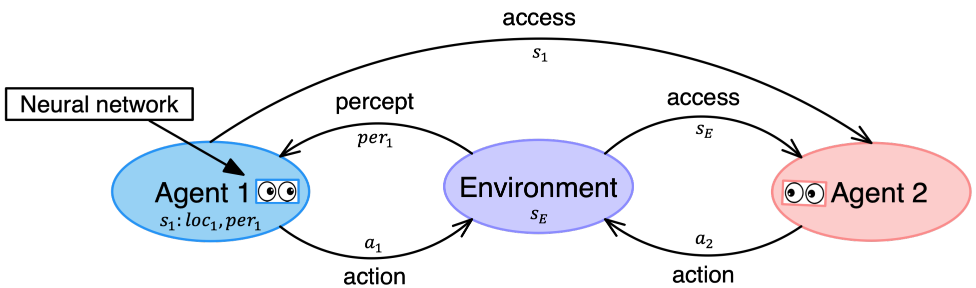
- Example: dynamic vehicle parking
  - NN maps exact vehicle position to perceived grid cell



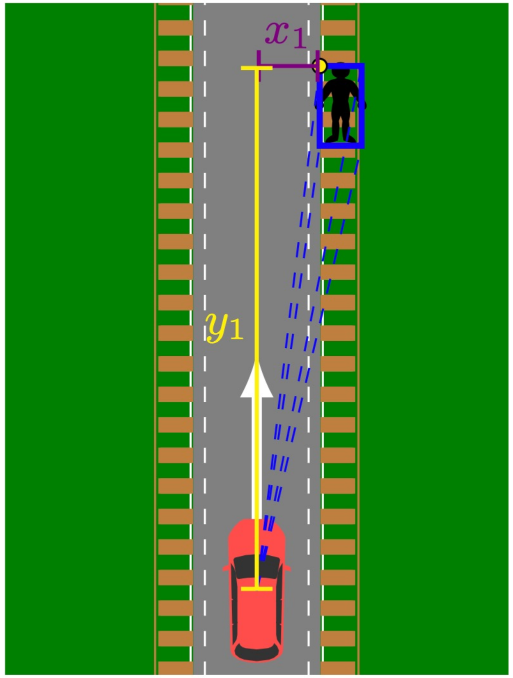


Strategy synthesis for zero-sum neuro-symbolic concurrent stochastic games, Kwiatkowska *et al*, *Information & Computation*, 2025

Pedestrian–vehicle interaction as NS-POSG

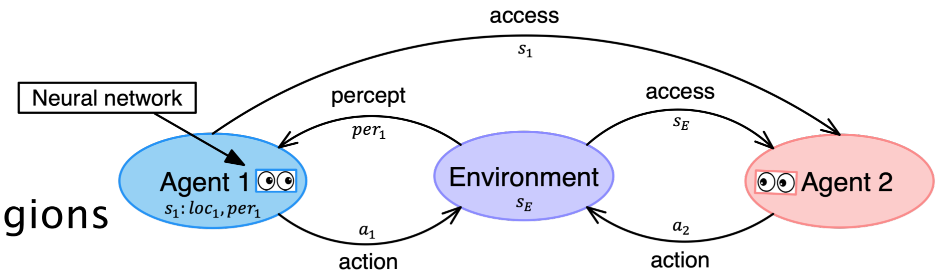


- Agent 1: vehicle
  - loc_1: speed
    per_1: pedestrian intention
  - a_1: acceleration (e.g. +3,-3)
- Agent 2: pedestrian
  - a_2: cross, back
- Environment E
  - two successive pedestrian positions (x_1,y_1,x_2,y_2)



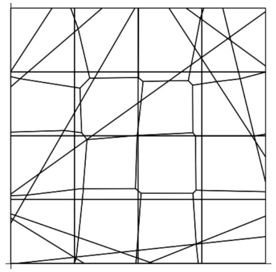
Strategy synthesis for neuro-symbolic POSGs

- 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

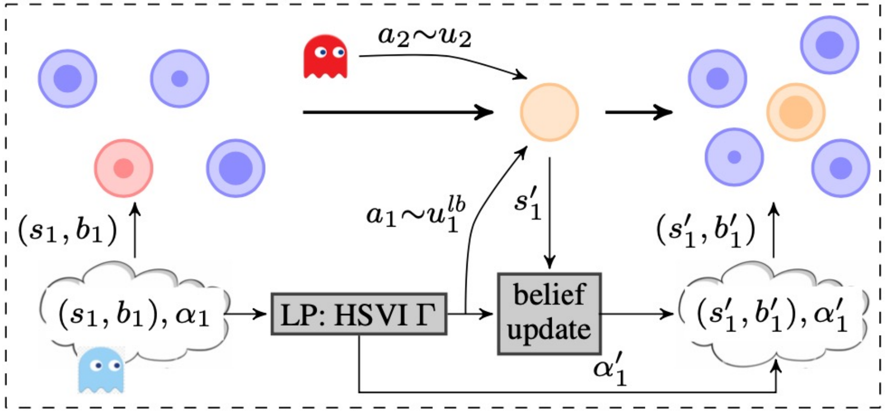


- Implementation
  - polyhedral pre-image computations of NNs
  - LPs to compute lower/upper bound and minimax values

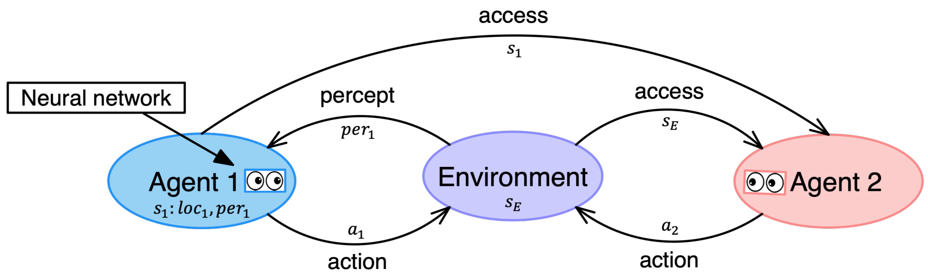
Partially observable stochastic games with neural perception mechanisms, In *Proc* FM 2024

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

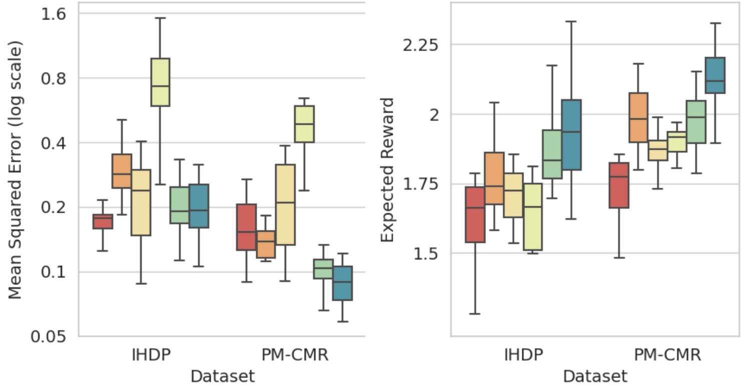
- keeps track of belief and counterfactual values
- builds and solves a game without storing complete strategy

Our variant

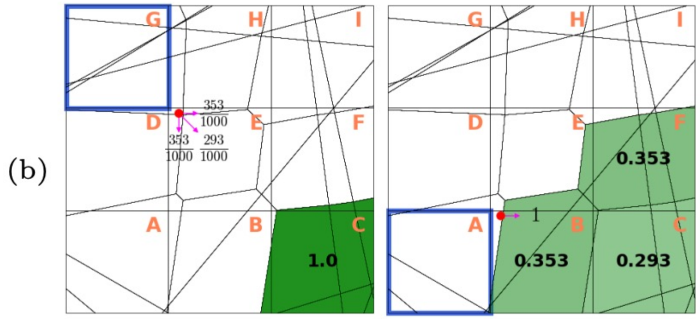
- precomputes HSVI lower bound
- keeps track of belief and PWC function α_1
- solves a single LP at each stage

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

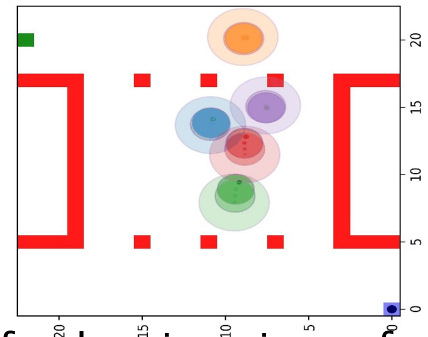
Multiple applications and NN verification use cases!



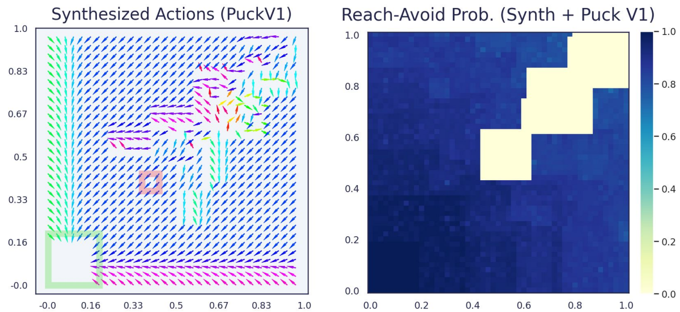
Sample-efficient policy learning



Protocol verification



Safe planning via conformal prediction



Optimal controller synthesis

http://fun2model.org/

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
  - provable guarantees needed for high-stakes decisions
  - safety, dependability, 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