Property-driven Machine Learning

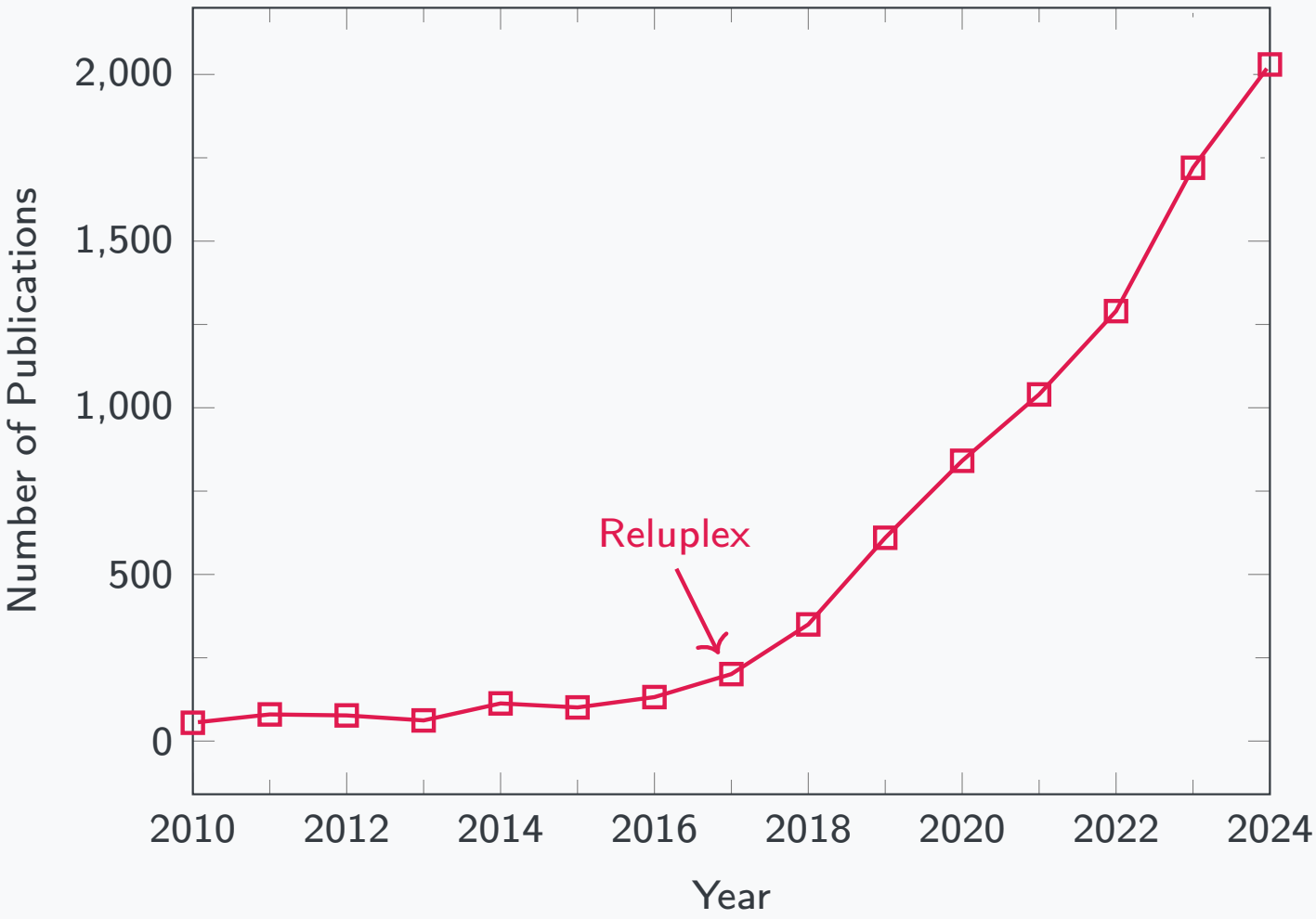
Thomas Flinkow

Department of Computer Science
Maynooth University

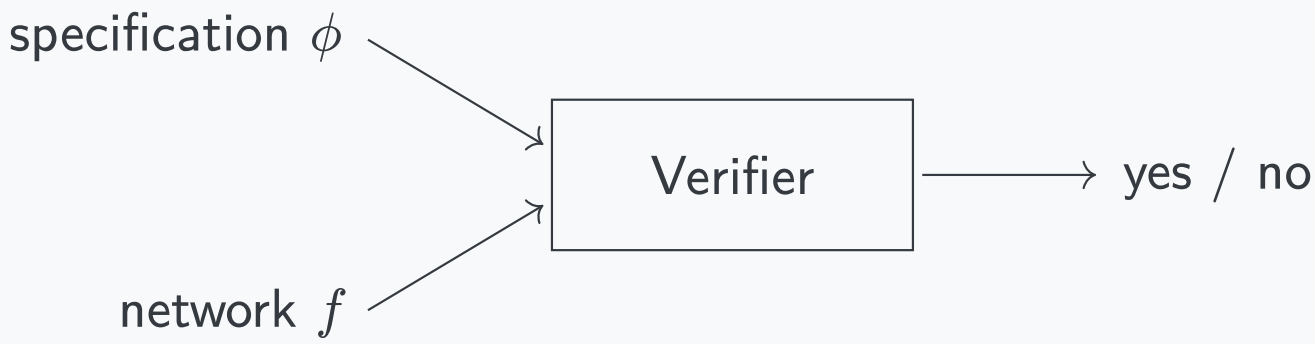
25th June 2025

Formal Verification of Neural Networks is Hot 🔥

Google Scholar search results for “formal verification” of “neural network”



Neural Network Verifiers

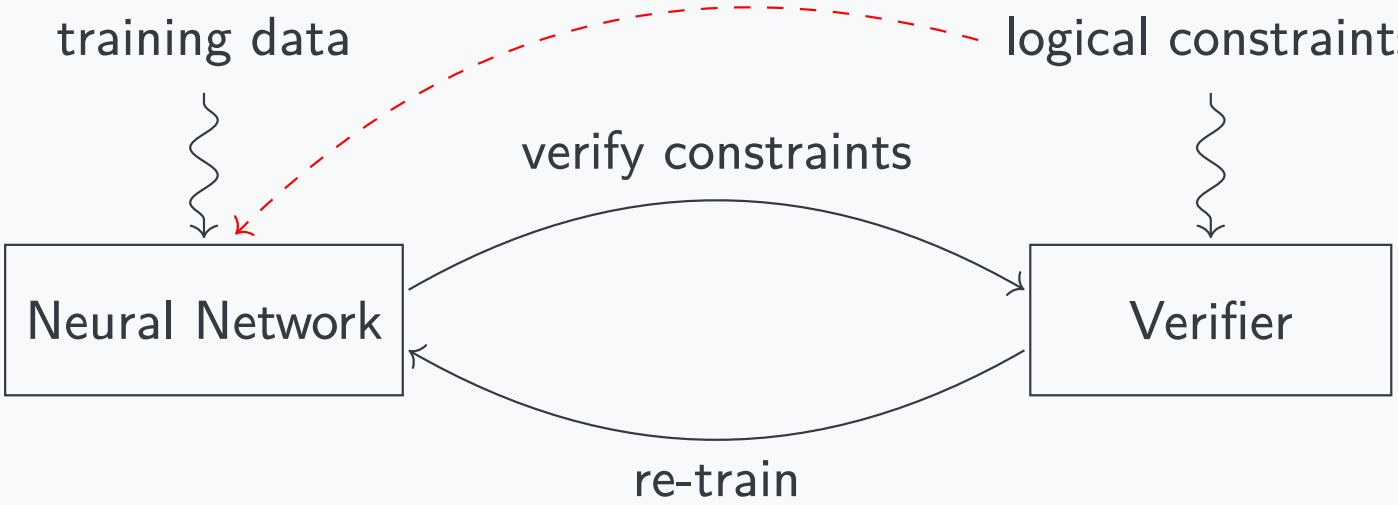


- **SMT:** Reluplex, Planet, Marabou, ...
- **MILP:** Branch-and-Bound, MIPVerify, Sherlock, ...
- **Abstract Interpretation:** AI2, DeepPoly, NNV, ...

From Data-driven Machine Learning...

https://xkcd.com/1838/

...to Property-driven Machine Learning



Note

Training with constraints does not guarantee their satisfaction!

Differentiable Logics[1,2,3,4]

Logic	Domain	$[\![\top]\!]$	$[\![\bot]\!]$	$[\![\neg x]\!]$	$[\![x \wedge y]\!]$	$[\![x \vee y]\!]$
DL2	$[0, \infty)$	0	∞	undefined[a]	xy	$x + y$
Gödel	$[0, 1]$	1	0	$1 - x$	$\min\{x, y\}$	$\max\{x, y\}$
Product					xy	$x + y - xy$
STL	$(-\infty, \infty)$	∞	$-\infty$	$-x$	omitted[b]	omitted[b]

[1] M. Fischer et al. (2019). 'DL2: Training and Querying Neural Networks with Logic'.
[2] E. van Krieken et al. (2022). 'Analyzing Differentiable Fuzzy Logic Operators'. DOI: 10.1016/j.artint.2021.103602.
[3] N. Ślusarz et al. (2023). 'Logic of Differentiable Logics: Towards a Uniform Semantics of DL'. DOI: 10.29007/c1nt.
[4] P. Varnai et al. (2020). 'On Robustness Metrics for Learning STL Tasks'. DOI: 10.23919/ACC45564.2020.9147692.

How do these differentiable logics compare?

Research Question

Given these different logics with different domains and operators, how do these compare in terms of:

- derivatives?
- logical consistency?
- in practice?

Results

Training with *any* differentiable logic leads to significantly improved constraint satisfaction.

Paper

T. Flinkow, B. A. Pearlmutter and R. Monahan (2025). 'Comparing Differentiable Logics for Learning with Logical Constraints'. In: *Science of Computer Programming* 244, p. 103280. ISSN: 0167-6423. DOI: 10.1016/j.scico.2025.103280

A General Approach for Property-driven Machine Learning

Goal

Train a network f to satisfy properties of the form

$$\forall \boldsymbol{x}.\, P(\boldsymbol{x}) \longrightarrow Q(f(\boldsymbol{x}))$$

Approach

1. translate P into a hyper-rectangle
2. handle $\forall \boldsymbol{x}.\, P(x)$ by finding an adversarial sample that violates P
3. translate $Q(f(\boldsymbol{x}))$ into additional loss

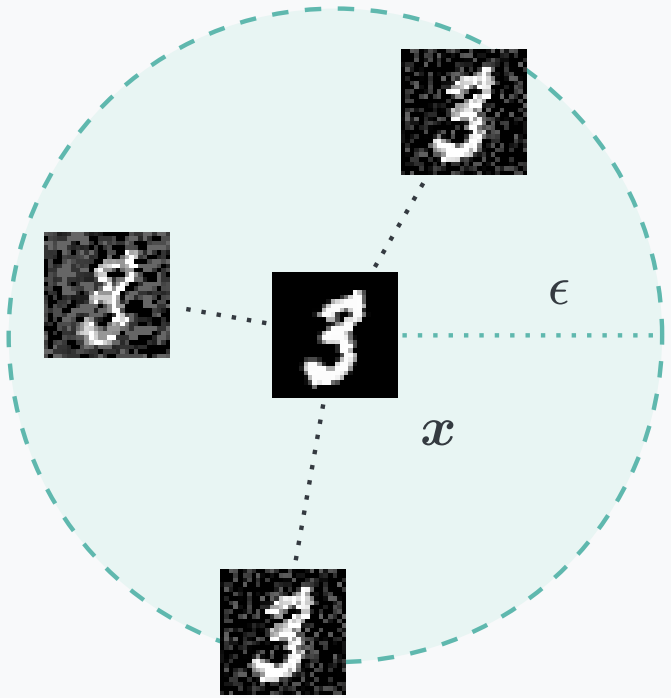
Preprint

T. Flinkow, M. Casadio, C. Kessler, R. Monahan and E. Komendantskaya (2025). *A General Framework for Property-Driven Machine Learning*. DOI: 10.48550/arXiv.2505.00466. arXiv: 2505.00466 [cs]

Local Robustness and ϵ-balls

Local robustness:

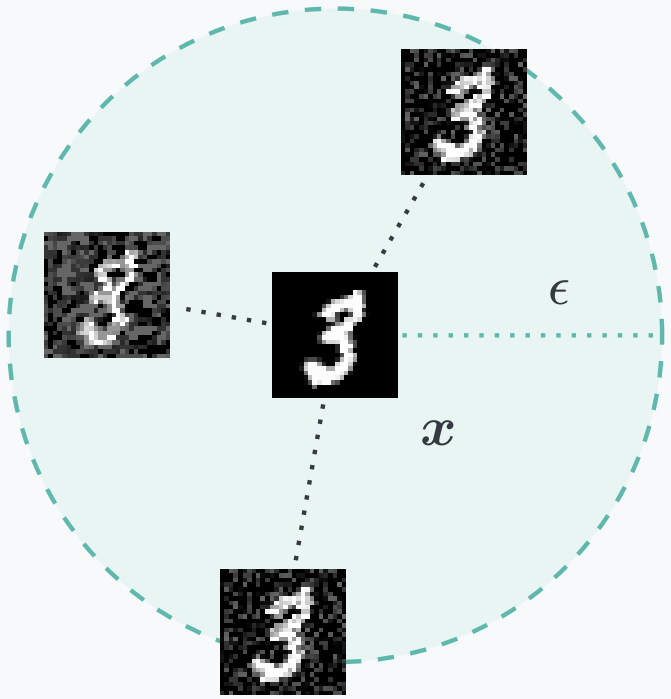
$$\forall \boldsymbol{x}'.\ \underbrace{\|\boldsymbol{x} - \boldsymbol{x}'\| \leq \epsilon}_{\mathbb{B}(\boldsymbol{x};\epsilon)} \longrightarrow \arg\max_i f(\boldsymbol{x}') = y$$



Local Robustness and ϵ-balls

Local robustness:

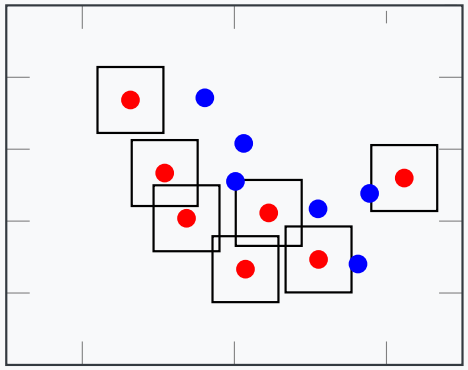
$$\forall \boldsymbol{x}'.\ \underbrace{\|\boldsymbol{x} - \boldsymbol{x}'\| \leq \epsilon}_{\mathbb{B}(\boldsymbol{x};\epsilon)} \longrightarrow \arg\max_i f(\boldsymbol{x}') = y$$



Problems

- Natural Language Processing: discrete space, no meaningful sentences in ϵ-ball
- Cyber-physical Systems: low-dimensional input space with semantics (e.g. velocity, distance)

From ϵ-balls to Hyper-rectangles

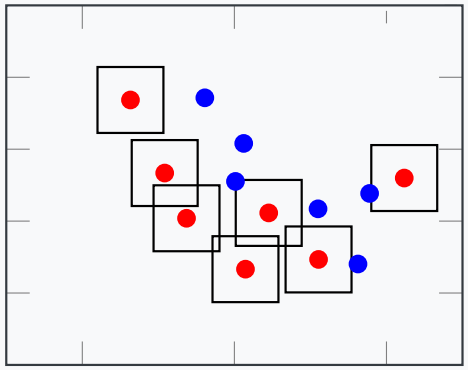


(a) ϵ-balls

- ℓ_∞-norm ϵ-ball:

$$\mathbb{B}(\boldsymbol{x}; \epsilon) := \{\boldsymbol{x} \in \mathbb{R}^m \mid x_i - \epsilon \leq x_i \leq x_i + \epsilon\}.$$

From ϵ-balls to Hyper-rectangles



(a) ϵ-balls

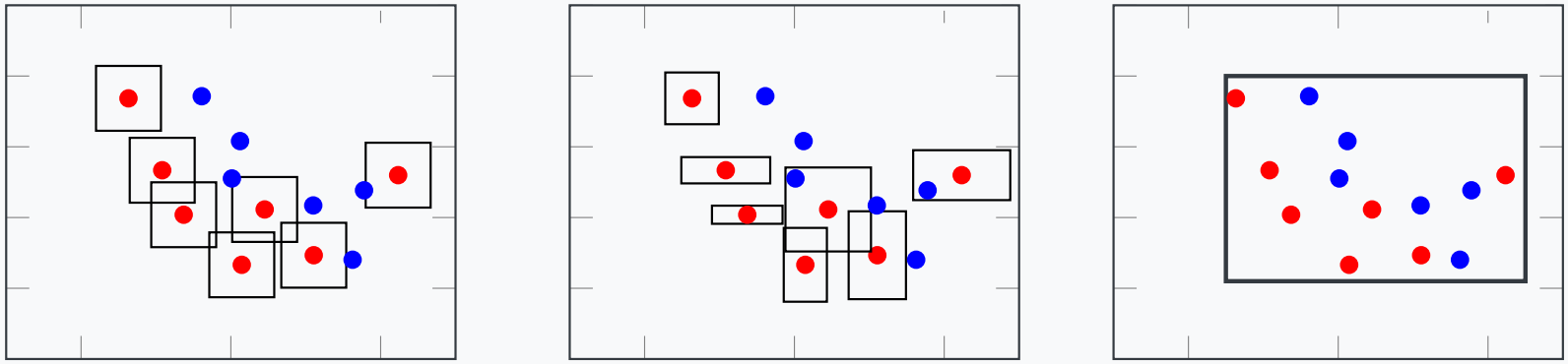
- ℓ_∞-norm ϵ-ball:

$$\mathbb{B}(\boldsymbol{x}; \epsilon) := \{\boldsymbol{x} \in \mathbb{R}^m \mid x_i - \epsilon \leq x_i \leq x_i + \epsilon\}.$$

- Hyper-rectangle:

$$\mathbb{H}(\boldsymbol{l}, \boldsymbol{u}) := \{\boldsymbol{x} \in \mathbb{R}^m \mid l_i \leq x_i \leq u_i\}.$$

From ϵ-balls to Hyper-rectangles



(a) ϵ-balls **(b)** local hyper-rectangles **(c)** a global hyper-rectangle

- ℓ_∞-norm ϵ-ball:

$$\mathbb{B}(\boldsymbol{x}; \epsilon) := \{\boldsymbol{x} \in \mathbb{R}^m \mid x_i - \epsilon \leq x_i \leq x_i + \epsilon\}.$$

- Hyper-rectangle:

$$\mathbb{H}(\boldsymbol{l}, \boldsymbol{u}) := \{\boldsymbol{x} \in \mathbb{R}^m \mid l_i \leq x_i \leq u_i\}.$$

1. Generalising ϵ-balls to Hyper-rectangles

- Standard ML:

$$\text{minimise} \underset{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}{\mathbb{E}} \left[\mathcal{L}(\boldsymbol{x},\boldsymbol{y};f)\right]$$

[5] I. J. Goodfellow et al. (2015). *Explaining and Harnessing Adversarial Examples*. DOI: 10.48550/arXiv.1412.6572.

[6] A. Madry et al. (2018). 'Towards Deep Learning Models Resistant to Adversarial Attacks'.

1. Generalising ϵ-balls to Hyper-rectangles

- Standard ML:

$$\text{minimise} \underset{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}{\mathbb{E}} \left[\mathcal{L}(\boldsymbol{x},\boldsymbol{y};f)\right]$$

- Adversarial Training (FGSM,[5] PGD[6]):

$$\text{minimise} \underset{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}{\mathbb{E}} \left[\max_{\boldsymbol{x}'\in\mathbb{B}(\boldsymbol{x};\epsilon)} \mathcal{L}(\boldsymbol{x}',\boldsymbol{y};f)\right]$$

[5] I. J. Goodfellow et al. (2015). *Explaining and Harnessing Adversarial Examples*. DOI: 10.48550/arXiv.1412.6572.

[6] A. Madry et al. (2018). 'Towards Deep Learning Models Resistant to Adversarial Attacks'.

1. Generalising ϵ-balls to Hyper-rectangles

- Standard ML:

$$\text{minimise} \underset{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}{\mathbb{E}} \left[\mathcal{L}(\boldsymbol{x},\boldsymbol{y};f)\right]$$

- Adversarial Training (FGSM,[5] PGD[6]):

$$\text{minimise} \underset{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}{\mathbb{E}} \left[\max_{\boldsymbol{x}'\in\mathbb{B}(\boldsymbol{x};\epsilon)} \mathcal{L}(\boldsymbol{x}',\boldsymbol{y};f)\right]$$

- Generalising ϵ-balls to hyper-rectangles:

$$\text{minimise} \underset{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}{\mathbb{E}} \left[\max_{\boldsymbol{x}'\in\mathbb{H}(\boldsymbol{x})} \mathcal{L}(\boldsymbol{x}',\boldsymbol{y};f)\right]$$

[5] I. J. Goodfellow et al. (2015). *Explaining and Harnessing Adversarial Examples*. DOI: 10.48550/arXiv.1412.6572.

[6] A. Madry et al. (2018). 'Towards Deep Learning Models Resistant to Adversarial Attacks'.

2. Generalising Robustness to Arbitrary Logical Specifications ϕ

- Adversarial Training with hyper-rectangles:

$$\text{minimise} \underset{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}{\mathbb{E}} \left[\max_{\boldsymbol{x}'\in\mathbb{H}(\boldsymbol{x})} \underbrace{\mathcal{L}(\boldsymbol{x}', \boldsymbol{y}; f)}_{\text{e.g. cross-entropy}} \right]$$

2. Generalising Robustness to Arbitrary Logical Specifications ϕ

- Adversarial Training with hyper-rectangles:

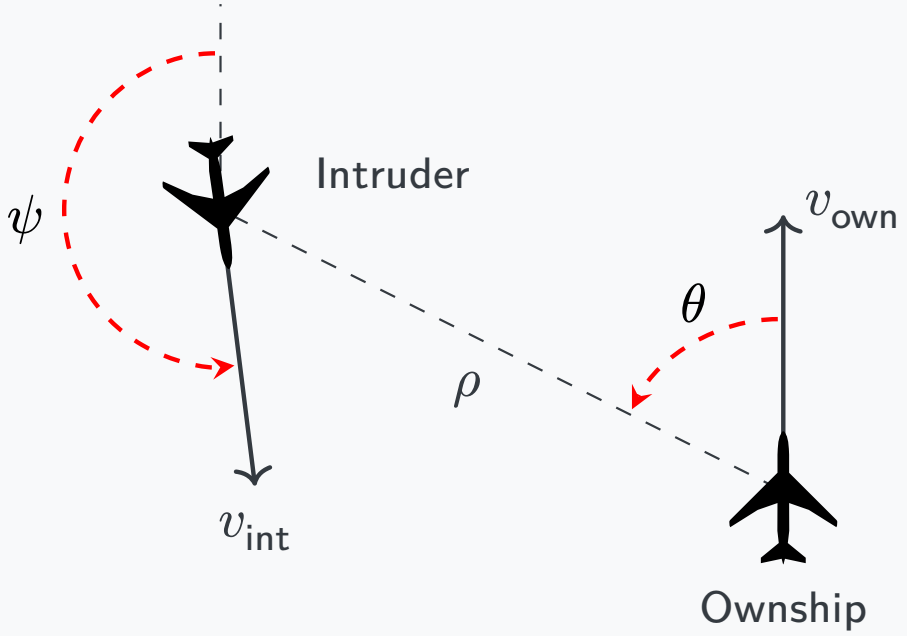
$$\text{minimise} \underset{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}{\mathbb{E}} \left[\max_{\boldsymbol{x}'\in\mathbb{H}(\boldsymbol{x})} \underbrace{\mathcal{L}(\boldsymbol{x}', \boldsymbol{y}; f)}_{\text{e.g. cross-entropy}} \right]$$

- Additional loss term for ϕ:

$$\text{minimise} \underset{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}{\mathbb{E}} \left[\lambda\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; f) + (1-\lambda) \max_{\boldsymbol{x}'\in\mathbb{H}(\boldsymbol{x})} [\![\phi]\!](\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{y}; f) \right]$$

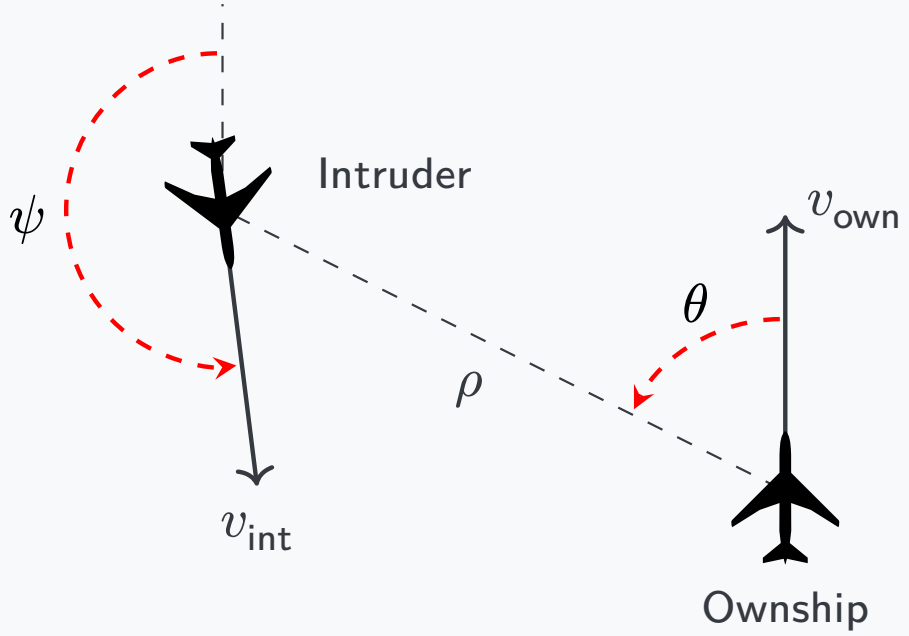
Example: ACAS Xu

ϕ: 'If intruder *near* and *approaching from the left*, network *advises "strong right"*'.



Example: ACAS Xu

ϕ: 'If intruder *near* and *approaching from the left*, network *advises "strong right"*'.

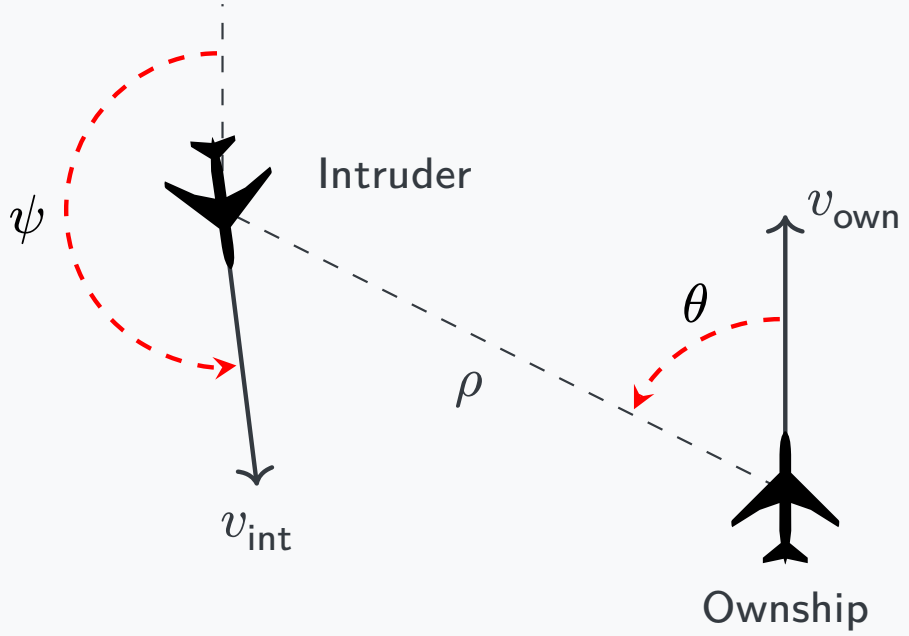


- $\mathbb{H}$ induced by "*near* and *approaching from the left*":

$$\rho \in [250\,\text{ft}, 400\,\text{ft}], \quad \theta \in [0.2\,\text{rad}, 0.4\,\text{rad}], \quad \psi \in [-\pi, -\pi + 0.005],$$
$$v_{\text{own}} \in [100\,\text{ft/s}, 400\,\text{ft/s}], \quad v_{\text{int}} \in [0\,\text{ft/s}, 400\,\text{ft/s}]$$

Example: ACAS Xu

ϕ: 'If intruder *near* and *approaching from the left*, network *advises "strong right"*'.



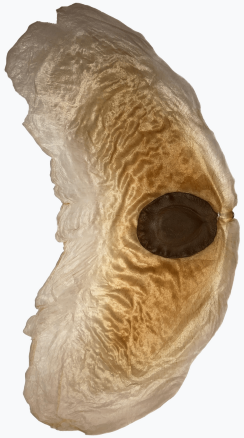
- $\mathbb{H}$ induced by "*near* and *approaching from the left*":

$$\rho \in [250\,\text{ft}, 400\,\text{ft}], \quad \theta \in [0.2\,\text{rad}, 0.4\,\text{rad}], \quad \psi \in [-\pi, -\pi + 0.005],$$
$$v_{\text{own}} \in [100\,\text{ft/s}, 400\,\text{ft/s}], \quad v_{\text{int}} \in [0\,\text{ft/s}, 400\,\text{ft/s}]$$

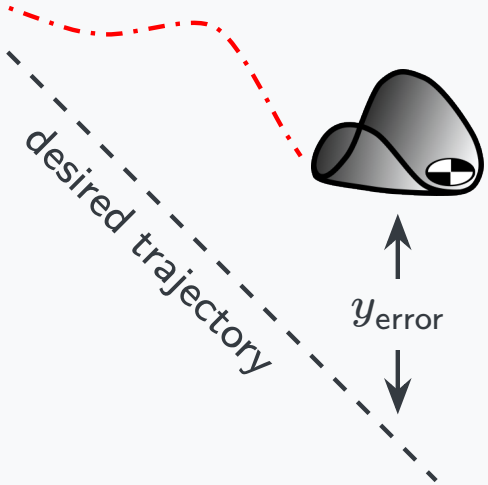
- Loss $[\![\phi]\!]$ induced by "*network advises 'strong right'*":

$$(y_{\text{SR}} < y_{\text{WL}}) \wedge (y_{\text{SR}} < y_{\text{WR}}) \wedge (y_{\text{SR}} < y_{\text{SL}}) \wedge (y_{\text{SR}} < y_{\text{COC}})$$

Experimental Results

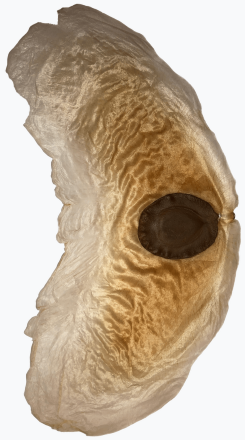


(a) An *Alsomitra macrocarpa* seed, capable of stable flight over long distances.

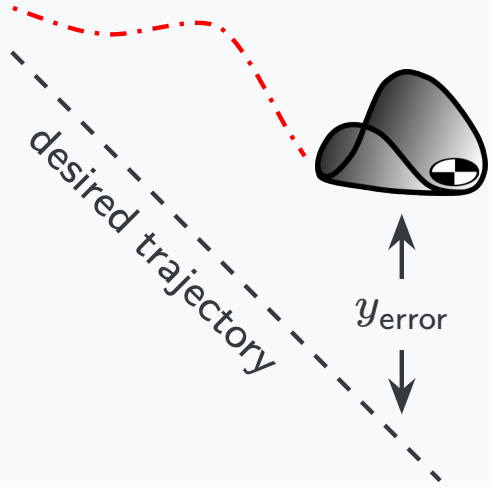


(b) The desired linear trajectory of the *Alsomitra*-inspired drone.

Experimental Results



(a) An *Alsomitra macrocarpa* seed, capable of stable flight over long distances.



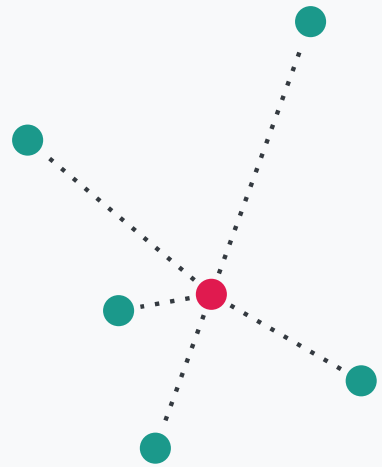
(b) The desired linear trajectory of the *Alsomitra*-inspired drone.

ϕ: *If the drone is above and close to the line, pitching down quickly and moving fast, the network will always make the drone pitch up.*

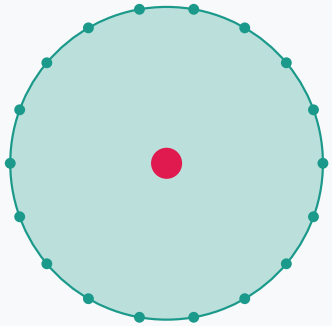
Logic	RMSE	CAcc (%)	CSec (%)
Baseline	3.6111×10^{-4}	0.00	0.00
DL2	1.2287×10^{-3}	100.00	95.31
Fuzzy logic	1.1632×10^{-3}	100.00	92.19

Future Work: Certified Training

Instead of *minimising* an *upper bound* on the loss, ...



(a) Adversarial Training



(b) Certified Training

...*maximise* a *lower bound* on robustness!

Summary & Conclusion

1. Motivation & Approach

Integrate logical specifications into training via differentiable logics.

2. Experimental Results

Increased constraint satisfaction, but no guarantees!

3. Future Work

Expressive specifications and formal guarantees.

Thomas Flinkow

Department of Computer Science
Maynooth University

Email: thomas.flinkow@mu.ie
https://www.cs.nuim.ie/~tflinkow/

Maynooth University
National University of Ireland Maynooth

Taighde Éireann Research Ireland

Thank you! Any questions?

- T. Flinkow, B. A. Pearlmutter and R. Monahan (2025). 'Comparing Differentiable Logics for Learning with Logical Constraints'. In: *Science of Computer Programming* 244, p. 103280. ISSN: 0167-6423. DOI: 10.1016/j.scico.2025.103280
- T. Flinkow, M. Casadio, C. Kessler, R. Monahan and E. Komendantskaya (2025). *A General Framework for Property-Driven Machine Learning*. DOI: 10.48550/arXiv.2505.00466. arXiv: 2505.00466 [cs]