Property-driven Machine Learning with Differentiable Logics

Maynooth University
National University of Ireland Maynooth

Thomas Flinkow, Barak A. Pearlmutter, Rosemary Monahan

Department of Computer Science, Maynooth University

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

Taighde Éireann Research Ireland

1. Motivation: Formal Verification of Neural Networks

Many formal verification tools can determine whether a neural network satisfies a given logical property. But how can we train a network to *learn* such a property in the first place?

2. Background: Property-driven ML

Standard ML: Given data x, target y, and loss $\mathcal{L}$,

$$\text{minimise} \underset{(x,y)\sim\mathcal{D}}{\mathbb{E}} \mathcal{L}(x,y).$$

Approach (based on Adversarial Training): Train network f to satisfy logical constraints of the form $\forall x.\, P(x) \rightarrow Q(f(x))$ by:

1. translating P into a *hyper-rectangle* $(\!|P|\!)$
2. translating Q into *real-valued, differentiable loss* $[\![Q]\!]$
3. finding a counterexample x^* in the input space $(\!|P|\!)$ that does *not satisfy* Q using PGD to maximise $[\![Q]\!]$:

$$x^* = \underset{x' \in (\!|P|\!)}{\arg\max}\, [\![Q]\!](x', y)$$

4. using this counterexample in training:

$$\text{minimise} \underset{(x,y)\sim\mathcal{D}}{\mathbb{E}} \big[\lambda \underbrace{\mathcal{L}(x,y)}_{\text{prediction loss}} + (1-\lambda) \underbrace{[\![Q]\!](x',y)}_{\text{logical constraint loss}}\big].$$

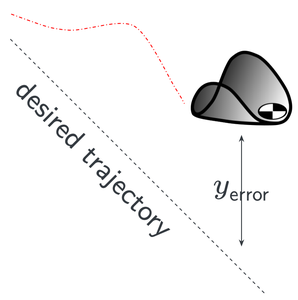
3. Research Questions

Many differentiable logics (DL2, STL, fuzzy logics, ...) with varying domains and operators, but how do they differ in terms of:

- **Learning behaviour:** gradients and suitability for optimisation.
- **Logical properties:** soundness and internal consistency.
- **Formal guarantees:** empirical performance *vs.* strong guarantees.

4. Differentiable Logics in Training

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



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

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

Logic	RMSE	Constraint Accuracy	Constraint Security
Baseline	3.6111×10^{-4}	0 %	0 %
DL2	1.2287×10^{-3}	100 %	95.3 %
Fuzzy logic	1.1632×10^{-3}	100 %	92.2 %

Findings: Property-driven training with *any* differentiable logic generally leads to significantly improved constraint satisfaction.

5. Current Work: Smooth Differentiable Logic

Idea: Prefer good gradients over logical soundness—just because PGD cannot find a counterexample, does *not* mean that none exist!

- less-than-or-equal atom as a *smooth approximation of* ReLU:

$$[\![x \leq y]\!] := \tau\, \mathrm{softplus}\big({}^{x-y-\delta}/_{\tau}\big)$$

- disjunction as a *smooth approximation of* $\min\{x_i\}$:
  1. via p-means:

$$[\![\bigvee x_i]\!] := \left(\frac{1}{n}\sum_{i=1}^{n} x_i^p\right)^{1/p}, \quad p < 0$$

  2. via log-sum-exp:

$$[\![\bigvee x_i]\!] := \tau\, \mathrm{LSE}({}^{x_1}/_{\tau}, \ldots, {}^{x_n}/_{\tau}), \quad \tau < 0$$

- conjunction as a *smooth approximation of* $\max\{x_i\}$ (dual of disjunction, via p-means with $p > 0$ or log-sum-exp with $\tau > 0$).

6. Property-driven ML and Formal Guarantees?

Example: Multi-label classification of dice faces.

Constraint: *Predictions must physically be possible, i.e. do not predict two faces at a time that are on opposite sides of the die.*
Using Marabou to determine verified constraint satisfaction on the test set consisting of 69 images on networks trained with $\epsilon = {}^{16}/_{255}$.

Logic	Prediction Accuracy	Constraint Accuracy	Constraint Security	Verified Satisfaction
Baseline	91.3 %	98.6 %	8.7 %	0 % $({}^{0}/_{66})$
DL2	91.3 %	100 %	37.7 %	0 % $({}^{0}/_{49})$
Gödel	91.3 %	100 %	7.3 %	0 % $({}^{0}/_{65})$
Reichenbach	92.8 %	98.6 %	7.3 %	0 % $({}^{0}/_{65})$
Our Logic	**91.3 %**	**100 %**	**100 %**	**100 %** $({}^{66}/_{66})$

Findings: Property-driven training yields some formal guarantees but often fails to establish strong ones.

7. Future Work: Formal Guarantees & Expressiveness

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

(a) Adversarial Training (b) Certified Training

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

Immediate next steps: Look into set-based training and train with a set of samples as opposed to using PGD to find counterexampels.

T. Flinkow, B. A. Pearlmutter, R. Monahan: **'Comparing differentiable logics for learning with logical constraints'**, In *Science of Computer Programming*, **244**, 103280 (September 2025). DOI: 10.1016/j.scico.2025.103280

This publication has emanated from research conducted with the financial support of Taighde Éireann – Research Ireland under grant number 20/FFP-P/8853.

T. Flinkow, M. Casadio, C. Kessler, E. Komendantskaya, R. Monahan: **'A General Framework for Property-Driven Machine Learning'**, (June 2025). PREPRINT: arXiv:2505.00466

This publication has emanated from research conducted with the financial support of Taighde Éireann – Research Ireland under grant number 20/FFP-P/8853 and by the ADAPT Research Ireland Centre for AI Driven Digital Content Technology at Maynooth University under grant number 13/RC/2106_P2.