Towards Correct-by-Construction Machine-Learnt Models

Thomas Flinkow*, Barak A. Pearlmutter and Rosemary Monahan

Department of Computer Science, Maynooth University, Maynooth, Co. Kildare, Ireland

Abstract

Various neural network verifiers have been developed to ensure that a neural network satisfies desired properties after training. A promising approach for creating correct-by-construction machine-learnt models is to incorporate explicit logical constraints into the training process via so-called differentiable logics. This paper provides an overview of our research area, our preliminary results, as well as an outline of future research directions.

Keywords

formal verification, machine learning, differentiable logics

1. Introduction

It has been shown that neural networks fail to learn background knowledge from data alone and are susceptible to adversarial inputs [1, 2], which has implications for their use in safety-critical domains.

Numerous verifiers for neural networks have emerged in the past few years, such as Reluplex [3], Marabou [4, 5], Branch-and-Bound [6], NNV [7], and α,β-CROWN [8–13], winner of the recent Neural Network Verification Competitions (VNN-COMP) [14–17]. For an overview of state-of-the art verifiers, we refer the interested reader to [18–21].

Verification of neural networks is typically limited to neural networks with fixed weights that have ceased learning [22]. A step in the direction of correct-by-construction neural networks are so called *differentiable logics*, used to incorporate logical constraints into the machine learning process.

2. Background

Machine learning. In gradient-based machine learning, optimal parameters $\boldsymbol{\theta}^+$ (such as neural network weights) are determined by minimising a *loss function*, $\mathcal{L}$, which quantifies the error between the predicted output and the desired output. This optimisation is typically achieved using gradient descent methods. The goal is to find the set of parameters $\boldsymbol{\theta}^+$ that minimises the loss function, formally expressed as

$$\boldsymbol{\theta}^+ = \arg\min_{\boldsymbol{\theta}}\ \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}), \tag{1}$$

where $\boldsymbol{x}$ represents the input data and $\boldsymbol{y}$ denotes the corresponding desired output.

Differentiable logics. The idea of learning with constraints is to incorporate a logical constraint ϕ into this optimisation process by translating the logical constraint into an additional loss term $\mathcal{L}_\phi$.

$$\boldsymbol{\theta}^+ = \arg\min_{\boldsymbol{\theta}}\ \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) + \lambda\mathcal{L}_\phi(\boldsymbol{x}, \boldsymbol{y}). \tag{2}$$

Note that the additional loss term introduces a new hyperparameter λ that is responsible for balancing the different loss terms. As explained in Section 3, in our experimental evaluation [23] we used the adaptive loss-balancing approach GradNorm [24] in order to find close-to-optimal values for λ.

PhD Symposium of the 19th International Conference on Integrated Formal Methods (iFM) at the University of Manchester, UK, 12 November 2024.

*Corresponding author.

thomas.flinkow@mu.ie (T. Flinkow); barak@pearlmutter.net (B. A. Pearlmutter); rosemary.monahan@mu.ie (R. Monahan)
0000-0002-8075-2194 (T. Flinkow); 0000-0003-0521-4553 (B. A. Pearlmutter); 0000-0003-3886-4675 (R. Monahan)

Various translations that map logical constraints into real-valued, differentiable functions have been defined in the literature, such as semantic loss [25], DL2 [26], designed specifically for incorporating constraints into neural networks, or fuzzy logic based ones [27–30], which exploit the fact that fuzzy logics are real-valued logics that often use operators that happen to be differentiable-almost-everwhere.

Specialised network architectures. Note that incorporating logical constraints into the machine learning pipeline via additional loss terms as done in Eq. (2) does not guarantee constraint satisfaction; other approaches exist that incorporate logical constraints into the network architecture, such as proposed by Li and Srikumar [31], DeepProbLog [32], Logic Tensor Networks (LTNs) [33, 34], MultiPlexNet [35], CNN [36], and CNN$^+$ [37].

3. Contributions to Date

The theory of these differentiable logics is well-studied [38–41] in the literature with respect to various interesting properties, such as (1) the shadow-lifting [42] property of a conjunction $x \wedge y$, which requires the truth value of the conjunction to increase when the truth value of one of its contituents increases, (2) whether implication operators admit classical logic reasoning such as Modus Ponens and Modus Tollens [39], and (3) the logical consistency [38] of operators, which looks at the maximum truth value obtainable for tautologies when using certain operators.

Given the wide range of possible logic translations available, our initial research question was: what is the optimal translation for use in training?

To address this question, we provide in [23] an experimental comparison of differentiable logic operators. Additionally, we provide a Python implementation [43] of various differentiable logics in PyTorch [44], implemented in a way that makes it easy to train any neural network on any dataset with arbitrary constraints.

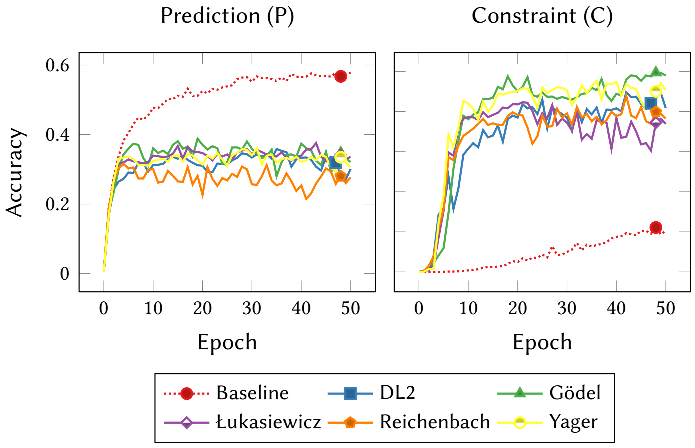
In order for our experimental comparison to be as fair as possible, we utilised Projected Gradient Descent (PGD) [45] to use a constraint counterexample in training as initially suggested by [26], which allows each logic to have the most impact on the learning process, and additionally we use the adaptive loss-balancing approach GradNorm [24] in order to estimate the parameter λ from Eq. (2) to balance the different loss terms, allowing each logic to perform at its best.

Experimental results. We obtained somewhat surprising results: while we expected to confirm theoretic results from the literature, we found that shadow-lifting conjunctions were not necessarily the best choice; neither were those implications that closely follow Modus Ponens and Modus Tollens reasoning. In general, training with any differentiable logic will lead to improved constraint satisfaction (albeit at an expense of prediction accuracy, as reported previously by Tsipras et al. [46]). However, the performance of the differentiable logics depends highly on the specific task at hand.

For example, we compared the performance of five different logic translation for training a neural network on the German Traffic Sign Recognition Benchmark (GTSRB) [47] to satisfy the constraint "the sum of probabilities of all elements in a group of related traffic signs should either be very high or very low". Here, we consider groups of related traffic signs (e.g. the group of all speed limit signs) in order to add background knowledge into the network.

As can be seen in Fig. 1, training with any differentiable logic leads to improved constraint accuracy and reduced prediction accuracy, however, the difference between the different logics is not as pronounced as expected from their theoretical properties.

Conclusions for future research. Instead of trying to find a single best one-size-fits-all differentiable logic that should be used in all scenarios, it might prove to be more fruitful to investigate what logical constraints mandate what properties the logic translation should exhibit. In the following section, we collate some interesting research areas which we have identified and which we plan to investigate in the future.



Logic	P	C
Baseline	56.73	22.15
DL2	31.92	84.12
Gödel	**34.76**	**99.39**
Łukasiewicz	34.13	74.54
Reichenbach	28.00	79.76
Yager	33.44	90.06

Figure 1: Training a network to satisfy a logical constraint on GTSRB with different logics. Surprisingly, the best-performing logic is the Gödel fuzzy logic, despite not having favourable theoretical properties such as shadow-lifting. Here, "Prediction Accuracy" is the percentage of correct predictions, and "Constraint Accuracy" the percentage of the constraint being satisfied.

4. Areas for Future Work

Specifications for machine learning. A common problem in the machine learning context is the lack of well-defined, general-purpose specifications [48–50] beyond often-used properties such as local robustness, which requires the neural network to be stable against slight perturbations to an input.

Additionally, despite there being complete verification techniques based on SMT or abstract interpretation, these require being able to specify a meaningful region of the input space. This is often infeasible in all but the most low-dimensional, interpretable settings such as the verification [3] of the experimental neural network compression [51] of the airborne collision avoidance system ACAS Xu [52], where meaningful regions of the input space can be expressed via constraints on the position and velocities of different aeroplanes.

For high-dimensional input spaces such as encountered in image classification, distinguishing meaningful images from noise is usually impossible, and verification is therefore usually limited to point-wise verification, which cannot provide any guarantees for the network behaviour on unseen data.

Going forward, it might prove to be beneficial to explore types of general-purpose properties (such as robustness or monotonicity) one might expect a neural network to satisfy across various applications.

Expressivity of differentiable logics. Logical constraints used in training are usually expressed in propositional logic, as in the ROAD-R dataset [53] for autonomous driving, which incorporates background knowledge such as ¬(Pedestrian ∧ Cyclist) or ¬(Traffic light green ∧ Traffic light red) into video frames. These constraints are sufficient to correct the network predictions if they do not align with the background knowledge, however, the authors note that future extensions of the dataset will investigate more expressive properties beyond propositional logic.

While properties such as local robustness [54] around point $\boldsymbol{x}_0$ are usually expressed as

$$\forall \boldsymbol{x}.\ \|\boldsymbol{x} - \boldsymbol{x}_0\|_\infty \leq \epsilon \rightarrow \|\mathcal{N}(\boldsymbol{x}) - \mathcal{N}(\boldsymbol{x}_0)\|_\infty \leq \delta, \tag{3}$$

the universal quantification is normally handled outside of the constraints by employing PGD to approximate the worst possible perturbation in the neighbourhood of $\boldsymbol{x}_0$ as initially suggested by Fischer et al. [26], however, a unifying approach capable of handling general universal (and existential) quantifiers is provided by Ślusarz et al. [40].

Going beyond first-order logic, especially in contexts such as video or natural language processing, one might like to employ temporal properties to model time-dependent behaviours. There are already differentiable temporal logics [42, 55–57]. We plan to investigate the ways in which these logics differ and identify the strengths and weaknesses of each.

Additionally, Farrell et al. [50] suggest there could be a need for probabilistic properties. To this end, approaches have been developed such as DeepProbLog [32] that allow for incorporating probabilistic constraints into neural networks.

Certified training. Using PGD to find the worst perturbation around a point as done for Eq. (3) does not provide any guarantees as it minimises a lower bound on the worst-case loss [19]. Instead of finding a worst perturbation, it would be interesting to investigate approaches based on certified training such as proposed by [58–61].

This area will be the immediate focus of our work, as we expect it to provide a solid foundation that all subsequent research efforts into expressive specifications and logics can benefit from.

Acknowledgments

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under grant number 20/FFP-P/8853.

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, 2014. doi:10.48550/arXiv.1312.6199. arXiv:1312.6199.

[2] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and Harnessing Adversarial Examples, 2015. doi:10.48550/arXiv.1412.6572. arXiv:1412.6572.

[3] G. Katz, C. Barrett, D. L. Dill, K. Julian, M. J. Kochenderfer, Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks, in: R. Majumdar, V. Kunčak (Eds.), Computer Aided Verification, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2017, pp. 97–117. doi:10.1007/978-3-319-63387-9_5.

[4] G. Katz, D. A. Huang, D. Ibeling, K. Julian, C. Lazarus, R. Lim, P. Shah, S. Thakoor, H. Wu, A. Zeljić, D. L. Dill, M. J. Kochenderfer, C. Barrett, The Marabou Framework for Verification and Analysis of Deep Neural Networks, in: I. Dillig, S. Tasiran (Eds.), Computer Aided Verification, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2019, pp. 443–452. doi:10.1007/978-3-030-25540-4_26.

[5] H. Wu, O. Isac, A. Zeljić, T. Tagomori, M. Daggitt, W. Kokke, I. Refaeli, G. Amir, K. Julian, S. Bassan, P. Huang, O. Lahav, M. Wu, M. Zhang, E. Komendantskaya, G. Katz, C. Barrett, Marabou 2.0: A Versatile Formal Analyzer of Neural Networks, 2024. doi:10.48550/arXiv.2401.14461. arXiv:2401.14461.

[6] R. Bunel, I. Turkaslan, P. Torr, M. Pawan Kumar, J. Lu, P. Kohli, Branch and bound for piecewise linear neural network verification, Journal of Machine Learning Research 21 (2020).

[7] H.-D. Tran, X. Yang, D. Manzanas Lopez, P. Musau, L. V. Nguyen, W. Xiang, S. Bak, T. T. Johnson, NNV: The Neural Network Verification Tool for Deep Neural Networks and Learning-Enabled Cyber-Physical Systems, in: S. K. Lahiri, C. Wang (Eds.), Computer Aided Verification, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2020, pp. 3–17. doi:10.1007/978-3-030-53288-8_1.

[8] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, L. Daniel, Efficient Neural Network Robustness Certification with General Activation Functions, 2018. doi:10.48550/arXiv.1811.00866. arXiv:1811.00866.

[9] K. Xu, Z. Shi, H. Zhang, Y. Wang, K.-W. Chang, M. Huang, B. Kailkhura, X. Lin, C.-J. Hsieh, Automatic Perturbation Analysis for Scalable Certified Robustness and Beyond, 2020. doi:10.48550/arXiv.2002.12920. arXiv:2002.12920.

[10] K. Xu, H. Zhang, S. Wang, Y. Wang, S. Jana, X. Lin, C.-J. Hsieh, Fast and Complete: Enabling Complete Neural Network Verification with Rapid and Massively Parallel Incomplete Verifiers, 2021. doi:10.48550/arXiv.2011.13824. arXiv:2011.13824.

[11] S. Wang, H. Zhang, K. Xu, X. Lin, S. Jana, C.-J. Hsieh, J. Z. Kolter, Beta-CROWN: Efficient Bound Propagation with Per-neuron Split Constraints for Complete and Incomplete Neural Network Robustness Verification, 2021. doi:10.48550/arXiv.2103.06624. arXiv:2103.06624.

[12] H. Zhang, S. Wang, K. Xu, L. Li, B. Li, S. Jana, C.-J. Hsieh, J. Z. Kolter, General Cutting Planes for Bound-Propagation-Based Neural Network Verification, 2022. doi:10.48550/arXiv.2208.05740. arXiv:2208.05740.

[13] Z. Shi, Q. Jin, Z. Kolter, S. Jana, C.-J. Hsieh, H. Zhang, Neural Network Verification with Branch-and-Bound for General Nonlinearities, 2024. doi:10.48550/arXiv.2405.21063. arXiv:2405.21063.

[14] S. Bak, C. Liu, T. Johnson, The Second International Verification of Neural Networks Competition (VNN-COMP 2021): Summary and Results, 2021. doi:10.48550/arXiv.2109.00498. arXiv:2109.00498.

[15] M. N. Müller, C. Brix, S. Bak, C. Liu, T. T. Johnson, The Third International Verification of Neural Networks Competition (VNN-COMP 2022): Summary and Results, 2022. doi:10.48550/arXiv.2212.10376. arXiv:2212.10376.

[16] C. Brix, S. Bak, C. Liu, T. T. Johnson, The Fourth International Verification of Neural Networks Competition (VNN-COMP 2023): Summary and Results, 2023. doi:10.48550/arXiv.2312.16760. arXiv:2312.16760.

[17] C. Brix, M. N. Müller, S. Bak, T. T. Johnson, C. Liu, First three years of the international verification of neural networks competition (VNN-COMP), International Journal on Software Tools for Technology Transfer 25 (2023) 329–339. doi:10.1007/s10009-023-00703-4.

[18] X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu, X. Yi, A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability, Computer Science Review 37 (2020) 100270. doi:10.1016/j.cosrev.2020.100270.

[19] C. Urban, A. Miné, A Review of Formal Methods applied to Machine Learning (2021). doi:10.48550/arXiv.2104.02466. arXiv:2104.02466.

[20] C. Liu, T. Arnon, C. Lazarus, C. Strong, C. Barrett, M. J. Kochenderfer, Algorithms for Verifying Deep Neural Networks, Foundations and Trends in Optimization 4 (2021) 244–404. doi:10.1561/2400000035.

[21] A. Albarghouthi, Introduction to Neural Network Verification, 2021. doi:10.48550/arXiv.2109.10317. arXiv:2109.10317.

[22] M. Kwiatkowska, Safety verification for deep neural networks with provable guarantees, in: Leibniz International Proceedings in Informatics, LIPIcs, volume 140, 2019. doi:10.4230/lipics.concur.2019.1.

[23] T. Flinkow, B. A. Pearlmutter, R. Monahan, Comparing Differentiable Logics for Learning with Logical Constraints, 2024. doi:10.48550/arXiv.2407.03847. arXiv:2407.03847.

[24] Z. Chen, V. Badrinarayanan, C.-Y. Lee, A. Rabinovich, GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks, in: Proceedings of the 35th International Conference on Machine Learning, PMLR, 2018, pp. 794–803. URL: https://proceedings.mlr.press/v80/chen18a.html.

[25] J. Xu, Z. Zhang, T. Friedman, Y. Liang, G. V. den Broeck, A Semantic Loss Function for Deep Learning with Symbolic Knowledge, 2018. doi:10.48550/arXiv.1711.11157. arXiv:1711.11157.

[26] M. Fischer, M. Balunovic, D. Drachsler-Cohen, T. Gehr, C. Zhang, M. Vechev, DL2: Training and Querying Neural Networks with Logic, in: Proceedings of the 36th International Conference on Machine Learning, PMLR, 2019, pp. 1931–1941.

[27] E. Giunchiglia, M. C. Stoian, T. Lukasiewicz, Deep Learning with Logical Constraints, in: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, Vienna, Austria, 2022, pp. 5478–5485. doi:10.24963/ijcai.2022/767.

[28] Z. Li, Z. Liu, Y. Yao, J. Xu, T. Chen, X. Ma, J. Lü, Learning with Logical Constraints but without Shortcut Satisfaction, in: The Eleventh International Conference on Learning Representations, 2022.

[29] H. He, W. Dai, M. Li, Reduced Implication-bias Logic Loss for Neuro-Symbolic Learning, 2023. doi:10.48550/arXiv.2208.06838. arXiv:2208.06838.

[30] M. Stoian, E. Giunchiglia, T. Lukasiewicz, Exploiting T-norms for Deep Learning in Autonomous Driving, in: CEUR Workshop Proceedings, volume 3432, 2023, pp. 369–380.

[31] T. Li, V. Srikumar, Augmenting Neural Networks with First-order Logic, in: A. Korhonen, D. Traum, L. Màrquez (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 292–302. doi:10.18653/v1/P19-1028.

[32] R. Manhaeve, S. Dumancic, A. Kimmig, T. Demeester, L. De Raedt, DeepProbLog: Neural Probabilistic Logic Programming, in: Advances in Neural Information Processing Systems, volume 31, Curran Associates, Inc., 2018.

[33] L. Serafini, A. d'Avila Garcez, Logic Tensor Networks: Deep Learning and Logical Reasoning from Data and Knowledge, 2016. doi:10.48550/arXiv.1606.04422. arXiv:1606.04422.

[34] S. Badreddine, A. d'Avila Garcez, L. Serafini, M. Spranger, Logic Tensor Networks, Artificial Intelligence 303 (2022) 103649. doi:10.1016/j.artint.2021.103649. arXiv:2012.13635.

[35] N. Hoernle, R. M. Karampatsis, V. Belle, K. Gal, MultiplexNet: Towards Fully Satisfied Logical Constraints in Neural Networks, Proceedings of the AAAI Conference on Artificial Intelligence 36 (2022) 5700–5709. doi:10.1609/aaai.v36i5.20512.

[36] E. Giunchiglia, T. Lukasiewicz, Multi-Label Classification Neural Networks with Hard Logical Constraints, Journal of Artificial Intelligence Research 72 (2021) 759–818. doi:10.1613/jair.1.12850.

[37] E. Giunchiglia, A. Tatomir, M. C. Stoian, T. Lukasiewicz, CCN+: A neuro-symbolic framework for deep learning with requirements, Int. J. Approx. Reasoning 171 (2024). doi:10.1016/j.ijar.2024.109124.

[38] M. M. Grespan, A. Gupta, V. Srikumar, Evaluating Relaxations of Logic for Neural Networks: A Comprehensive Study, 2021. doi:10.48550/arXiv.2107.13646. arXiv:2107.13646.

[39] E. van Krieken, E. Acar, F. van Harmelen, Analyzing Differentiable Fuzzy Logic Operators, Artificial Intelligence 302 (2022) 103602. doi:10.1016/j.artint.2021.103602. arXiv:2002.06100.

[40] N. Ślusarz, E. Komendantskaya, M. Daggitt, R. Stewart, K. Stark, Logic of Differentiable Logics: Towards a Uniform Semantics of DL, in: EPiC Series in Computing, volume 94, EasyChair, 2023, pp. 473–493. doi:10.29007/c1nt.

[41] R. Affeldt, A. Bruni, E. Komendantskaya, N. Ślusarz, K. Stark, Taming Differentiable Logics with Coq Formalisation, 2024. doi:10.48550/arXiv.2403.13700. arXiv:2403.13700.

[42] P. Varnai, D. V. Dimarogonas, On Robustness Metrics for Learning STL Tasks, in: 2020 American Control Conference (ACC), 2020, pp. 5394–5399. doi:10.23919/ACC45564.2020.9147692.

[43] T. Finkow, GitHub repository: tflinkow/comparing-differentiable-logics, 2024. URL: https://github.com/tflinkow/comparing-differentiable-logics.

[44] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An Imperative Style, High-Performance Deep Learning Library, in: Advances in Neural Information Processing Systems, volume 32, Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

[45] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards Deep Learning Models Resistant to Adversarial Attacks, 2018. URL: https://openreview.net/forum?id=rJzIBfZAb.

[46] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, A. Madry, Robustness May Be at Odds with Accuracy, 2018. URL: https://openreview.net/forum?id=SyxAb30cY7.

[47] J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel, The German Traffic Sign Recognition Benchmark: A multi-class classification competition, in: The 2011 International Joint Conference on Neural

Networks, IEEE, San Jose, CA, USA, 2011, pp. 1453–1460. doi:10.1109/IJCNN.2011.6033395.

[48] S. A. Seshia, A. Desai, T. Dreossi, D. J. Fremont, S. Ghosh, E. Kim, S. Shivakumar, M. Vazquez-Chanlatte, X. Yue, Formal Specification for Deep Neural Networks, in: S. K. Lahiri, C. Wang (Eds.), Automated Technology for Verification and Analysis, volume 11138, Springer International Publishing, Cham, 2018, pp. 20–34. doi:10.1007/978-3-030-01090-4_2.

[49] M. Leucker, Formal Verification of Neural Networks?, in: G. Carvalho, V. Stolz (Eds.), Formal Methods: Foundations and Applications, Lecture Notes in Computer Science, Springer International Publishing, 2020, pp. 3–7. doi:10.1007/978-3-030-63882-5_1.

[50] M. Farrell, A. Mavridou, J. Schumann, Exploring Requirements for Software that Learns: A Research Preview, in: A. Ferrari, B. Penzenstadler (Eds.), Requirements Engineering: Foundation for Software Quality, Lecture Notes in Computer Science, Springer Nature Switzerland, 2023, pp. 179–188. doi:10.1007/978-3-031-29786-1_12.

[51] K. D. Julian, M. J. Kochenderfer, M. P. Owen, Deep Neural Network Compression for Aircraft Collision Avoidance Systems, Journal of Guidance, Control, and Dynamics 42 (2019) 598–608. doi:10.2514/1.g003724. arXiv:1810.04240.

[52] M. P. Owen, A. Panken, R. Moss, L. Alvarez, C. Leeper, ACAS Xu: Integrated Collision Avoidance and Detect and Avoid Capability for UAS, in: 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC), 2019, pp. 1–10. doi:10.1109/dasc43569.2019.9081758.

[53] E. Giunchiglia, M. C. Stoian, S. Khan, F. Cuzzolin, T. Lukasiewicz, ROAD-R: The autonomous driving dataset with logical requirements, Machine Learning 112 (2023) 3261–3291. doi:10.1007/s10994-023-06322-z.

[54] M. Casadio, E. Komendantskaya, M. L. Daggitt, W. Kokke, G. Katz, G. Amir, I. Refaeli, Neural Network Robustness as a Verification Property: A Principled Case Study, in: S. Shoham, Y. Vizel (Eds.), Computer Aided Verification, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2022, pp. 219–231. doi:10.1007/978-3-031-13185-1_11.

[55] Y. Xie, F. Zhou, H. Soh, Embedding Symbolic Temporal Knowledge into Deep Sequential Models, in: 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 4267–4273. doi:10.1109/ICRA48506.2021.9561952.

[56] Z. Xu, Y. S. Rawat, Y. Wong, M. Kankanhalli, M. Shah, Don't Pour Cereal into Coffee: Differentiable Temporal Logic for Temporal Action Segmentation, in: Advances in Neural Information Processing Systems, 2022.

[57] D. Li, M. Cai, C.-I. Vasile, R. Tron, Learning Signal Temporal Logic through Neural Network for Interpretable Classification, in: 2023 American Control Conference (ACC), 2023, pp. 1907–1914. doi:10.23919/ACC55779.2023.10156357.

[58] E. Wong, Z. Kolter, Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope, in: Proceedings of the 35th International Conference on Machine Learning, PMLR, 2018, pp. 5286–5295. URL: https://proceedings.mlr.press/v80/wong18a.html.

[59] E. Wong, F. Schmidt, J. H. Metzen, J. Z. Kolter, Scaling provable adversarial defenses, in: Advances in Neural Information Processing Systems, volume 31, Curran Associates, Inc., 2018. URL: https://papers.nips.cc/paper_files/paper/2018/hash/358f9e7be09177c17d0d17ff73584307-Abstract.html.

[60] M. Mirman, T. Gehr, M. Vechev, Differentiable Abstract Interpretation for Provably Robust Neural Networks, in: Proceedings of the 35th International Conference on Machine Learning, PMLR, 2018, pp. 3578–3586. URL: https://proceedings.mlr.press/v80/mirman18b.html.

[61] A. Raghunathan, J. Steinhardt, P. Liang, Certified Defenses against Adversarial Examples, 2020. doi:10.48550/arXiv.1801.09344. arXiv:1801.09344.