Position: Formal Methods are the Principled Foundation of Safe AI

Gagandeep Singh [1 2] **Deepika Chawla** [3]

Abstract

Artificial Intelligence (AI) has the potential to transform society and the economy. However, both public and private entities have been increasingly expressing significant concern about the potential of state-of-the-art AI models to cause societal and financial harm. This lack of trust is because they can misbehave, and we lack a clear, principled understanding of when, why, or how they fail. Formal methods offer a principled foundation to mitigate risks, yielding stronger notions of trust than possible with intuition or empirical methods, such as bug finding and benchmarking. In this position paper, we will describe how formal methods can be used for (i) proving that a trained model satisfies desirable safety properties, (ii) guiding the model updates during training towards satisfying safety properties, and (iii) reliably explaining and interpreting the black-box workings of AI models. We will discuss the challenges hindering the broader adoption of formal methods for AI safety and outline future directions for overcoming them.

1. Introduction

Deep neural networks (DNNs) are currently the dominant technology in artificial intelligence (AI) and have shown impressive performance in diverse applications, including autonomous driving (Bojarski et al., 2016), medical diagnosis (Amato et al., 2013), text generation (Brown et al., 2020), and logical reasoning (Pan et al., 2023). However, they can often fail unpredictably, causing concerns about their safety and trust when deployed in the real world (Ribeiro et al., 2016; Szegedy et al., 2014; Kurakin et al., 2017; Huang et al., 2023; Vega et al., 2024). Although standard training optimizes the model's accuracy, it does not take into account desirable properties such as *robustness* (the DNN should behave similarly for similar inputs), *fairness* (the DNN output should not depend too much on some legally protected attribute, such as gender or race), and *privacy* (the model should not leak confidential information). As a result, state-of-the-art models remain untrustworthy. Building trust in AI is essential to realizing its vast potential to positively transform society and the economy, and is one of the grand research challenges today.

Safety-Informed DNN Deployment Cycle. Figure 1 presents a general safety-informed pipeline for DNN development, applicable to any application domain (e.g., finance, vision, NLP). Safety, accuracy, and efficiency can often conflict with each other. DNN accuracy improves with model size, but that increases the inference cost (Huang et al., 2017). Similarly, models maximizing safety can have reduced accuracy (Tsipras et al., 2019). For example, a DNN classifier that always predicts the same class for all inputs is robust but has very low accuracy. As a result, it may not be possible to obtain DNNs that optimize all three objectives simultaneously. Depending on the target application, a developer may prioritize accuracy over safety/efficiency or vice-versa. The goal of safety-informed DNN development is to ensure an application-specific acceptable balance between accuracy, safety, and efficiency.

In this pipeline, first, representative training data for the target application is collected and a DNN is trained to maximize its accuracy on test inputs from the training distribution. Next, a domain expert creates (manually or algorithmically) a set of formal safety specifications (e.g., robustness, fairness) mathematically characterizing the expected DNN behavior in different real-world scenarios (Katz et al., 2017; Chaudhary et al., 2024; Chen et al., 2021). The number of inputs covered by these specifications can be infinite.

The expert then checks whether the model meets the safety standards. Since DNNs may not satisfy all the specifications, the standards can require that at least a significant fraction of all specifications be satisfied for trustworthiness. If the model meets the criteria, then the DNN is considered fit for deployment. Otherwise, it is iteratively repaired (e.g., by fine-tuning) until we obtain the desired balance between accuracy, safety, and efficiency.

[*]Equal contribution [1]School of Computing and Data Science, University of Illinois Urbana-Champaign (UIUC), IL, USA [2]Institute of Government and Public Affairs, IL, USA [3]Independent Researcher, USA. Correspondence to: Gagandeep Singh <ggnds@illinois.edu>, Deepika Chawla <deepinder.chawla.eco@gmail.com>.

Workshop on Technical AI Governance (TAIG) at ICML 2025, Vancouver, Canada.

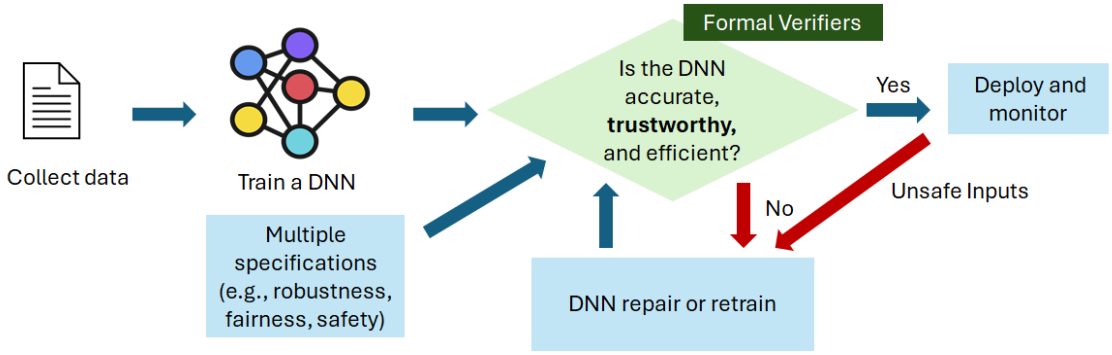


Figure 1. Development pipeline for building accurate, trustworthy, and efficient DNNs. Formal verification is used for testing model trustworthiness (green diamond).

During deployment, the DNN inputs are monitored for distribution shifts, i.e., the inputs are not covered by existing safety specifications. If the runtime system detects a distribution shift, it reports representative samples to the domain experts. They then design new specifications, and the model undergoes another round of repair (or full retraining) (Sotoudeh & Thakur, 2021).

How Formal Methods Can Help. For checking that the model satisfies safety specifications, the standard practice is to evaluate the DNN behavior on a finite set of inputs satisfying the specifications (Liang et al., 2023; Guo et al., 2023). However, this cannot guarantee safe and trustworthy DNN behavior on all specification inputs. The unseen set can be huge and contain inputs often seen during real-world deployment. To address these limitations, there is growing work on checking the safety of DNN models and interpreting their behavior, on an infinite set of unseen inputs from safety specifications using formal methods, which **provides a more reliable metric for measuring a model's safety than standard empirical methods** (Chakravarthy et al., 2022; Xu et al., 2024). For example, a repaired DNN preserving the original test set accuracy and efficiency but satisfying the trustworthy specifications more often is a better model than the unrepaired one as it is less likely to show undesirable behavior during real-world deployment. Formal methods can be leveraged during training and inference to guide the model to satisfy desirable safety and trustworthiness properties. Finally, formal methods can be used to build reliable explanations and interpretations that improve transparency and foster greater trust in AI models.

So Why are Formal Methods Not as Widespread? A common explanation is that formal methods demand a strong foundation in logic and mathematics, which many DNN developers find daunting. However, that only explains a part of the problem. Precise mathamtical details of how automatic differentiation works are just as complex, yet DNN developers routinely train their models without grappling all those details (Sherman et al., 2021; Krawiec et al., 2022). The deeper challenge is a fundamental mismatch of priorities. Practitioners building formal method tools value providing the strongest possible safety guarantees for specific scenarios, over usability and actionable feedback (Brix et al., 2024). On the other hand, the developers of real-world DNNs value speed, usability, and actionable feedback. Their goal is **not necessarily to achieve the strongest possible guarantees, but to rapidly assess and improve model safety within fast-moving development cycles**. Compared to empirical evaluation, applying formal methods to conduct safety evaluation can be time consuming. This is because unlike training a DNN, where high-level, optimized libraries abstract away the math, current tools, are not user friendly, not optimized to handle larger models, are developed for specific DNNs and use cases, and cannot be easily adopted to application-specific demands, making them unsuitable for fast-paced modern DNN development cycles.

Our Position. We believe that to make formal methods more widespread, developers must see them as a natural and valuable part of their workflow. This requires making formal methods easy to integrate into existing development pipelines — not an added burden. This means that we first need to make optimized, user-friendly, and adaptable tools (Singh et al., 2024; 2025a), even if they cannot provide the strongest possible safety guarantees. These lighter-weight safety assurances can still provide valuable actionable insights to developers that are not possible with empirical evaluation.

Improved adoption can lead to a **virtuous cycle: as formal methods become easier to use and more prevalent, developer interest grows, driving demand for stronger guarantees and fostering the development of even more powerful tools**. To start this cycle, it is crucial to make formal methods feel approachable rather than daunting. Therefore, we focus on a gentle introduction to formal methods emphasizing actionable insights over the strongest possible

safety assurances. A more detailed, intuitive explanation of the formal methods for verification, training, explanations, and interpretations presented in this paper is available in (Singh et al., 2025b). For an in-depth discussion of the broader challenges in delivering strong safety guarantees, we refer readers to the excellent analyses by (Dalrymple et al., 2024; Seshia & Sadigh, 2016).

This paper is organized as follows: we first describe how safety and trustworthy properties can be formally specified for DNNs. Next, we will discuss how formal methods can provide a principled foundation for proving that DNNs satisfy desirable safety properties, training them to be provably safe, and enabling reliable explanations and interpretations. Finally, we provide policy recommendations for building governable and safe AI grounded in formal methods.

2. Formal Specifications for DNNs

To reduce risks from AI models in real-world scenarios, we must move beyond ad hoc testing and examples (Wang et al., 2023). Instead, we need unambiguous, formal specifications of the safety properties. Unlike input-output examples, which offer limited snapshots of desirable behaviors, **formal specifications provide comprehensive, precise definitions of what 'safe' means, making them essential for building effective AI governance frameworks**. To develop formal specifications, we model a trained DNN as a function f. Its input x can be images, text, videos, sensor measurements, or other data. We denote the output of the DNN as $f(x)$, which can be a classification of the input into one of the predefined classes, a regression that estimates a continuous value, or the set of tokens generated by a language model.

For a trained DNN f, a developer specifies the property of interest using two formulas: (1) *the precondition* φ, which specifies the set of inputs on which the DNN should not misbehave and (2) *the postcondition* ψ, which specifies safe and trustworthy behaviors of the DNN for the given inputs. These behaviors are typically constraints on the DNN's outputs. The preconditions and postconditions are domain-dependent and usually designed by DNN developers.A tool for DNN verification (*a verifier*) aims to automatically check if the postcondition on the DNN's outputs is satisfied for all inputs specified by the precondition (Singh et al., 2019; Wang et al., 2021; Wu et al., 2022).

A property specification is a tuple (φ, ψ), where φ is the precondition and ψ is the postcondition. Both formulas φ and ψ typically represent *an infinite number of inputs/outputs*. We denote the set of the results of the evaluations of the DNN on all inputs described by the precondition φ as $f(\varphi) = \{f(x) \mid x \in \varphi\}$. The verifier then checks for the inclusion of the set of possible executions of the DNN into the set of outputs that satisfy the postcondition, i.e., $f(\phi) \subseteq \psi$ holds. *Single execution* specifications, as shown in Figure 2, require that each DNN output $f(x)$ where $x \in \varphi$ must independently satisfy ψ (Balunovic et al., 2019). *Relational* specifications require reasoning about multiple related executions of the same or different DNNs (Banerjee et al., 2024b). φ and ψ can also define distributions leading to *probabilistic* specifications (Chaudhary et al., 2025).

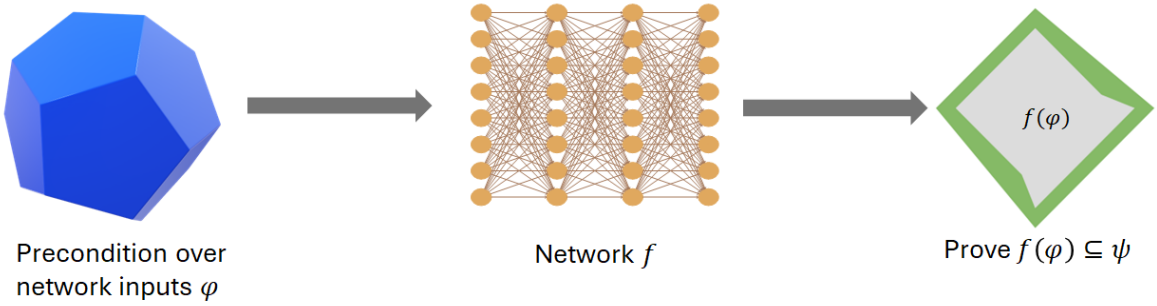
Local and Global properties. The set of specifications for DNNs can be broadly classified as *local* or *global*. The precondition φ for local properties defines a local neighborhood around a sample input from the test set. For example, given a test image correctly classified as a car by a DNN, the commonly used *local robustness property* specifies that if the original image was classified as a car, then all images generated by rotating the original image within $\pm d$ degrees are also classified as a car (Yang et al., 2023).

In contrast, global properties are not defined with respect to a specific test input (Kabaha & Drachsler-Cohen, 2024). Verifying global properties yields stronger safety guarantees compared to local properties, however, global properties are difficult to formulate for popular domains, such as vision and NLP, where the individual features processed by the DNN have no clear semantic meaning. While verifying local properties is not ideal, the local verification results enable testing the safety of the model on an infinite set of unseen inputs, not possible with standard methods.

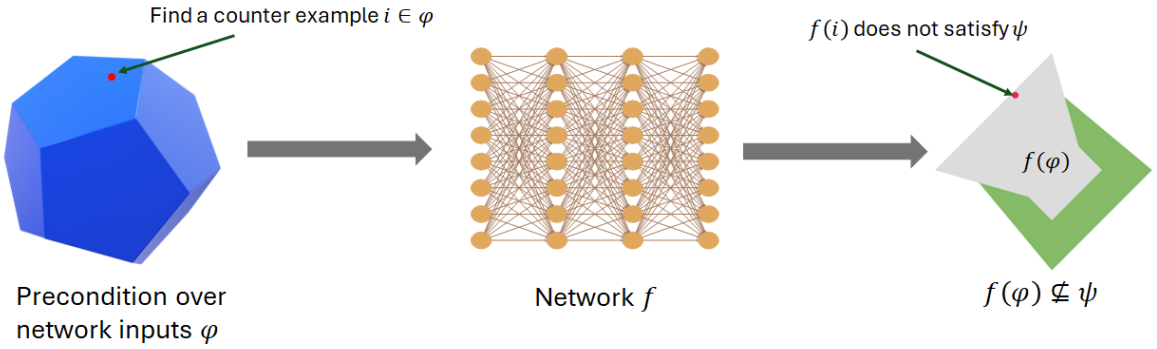
3. Formal Verification

DNN verifiers are typically *white-box*, requiring access to model parameters (Singh et al., 2019). DNN verification is an undecidable problem in general. Certain problems, such as robustness verification of feedforward DNNs with ReLU activations, are decidable but still NP-complete (Katz et al., 2017). State-of-the-art verifiers are therefore *incomplete* in general, i.e., they can fail to prove a specification when it holds. However, when they succeed, the DNN will satisfy the specification. The verifier works by computing an overapproximation $g(\varphi) \supseteq f(\varphi)$. It starts with φ and symbolically propagates it sequentially through the DNN layers. $g(\varphi)$ is typically a convex shape that is easier to compute than $f(\varphi)$. The verifier then checks whether $g(\varphi) \subseteq \psi$ holds. If it holds, then $f(\varphi) \subseteq \psi$ is true. Otherwise, the result is unknown.

There is a tradeoff between the cost and overapproximation error (also known as precision) of an incomplete verifier: expensive verifiers are more precise, while cheap verifiers are imprecise. The key consideration in designing an efficient verifier applicable to real-world DNNs is managing this tradeoff. For efficient verification, researchers have developed numerous methods for symbolic propagation for DNN verification (Li et al., 2020). These methods can



(a) Verified



(b) Counterexample

Figure 2. Single execution specifications require that the DNN output for each input from φ must independently satisfy ψ.

scale to realistic DNNs with millions of neurons, or more than 100 layers, verifying diverse safety properties in different real-world applications (Müller et al., 2021; Wang et al., 2021). We refer the reader to the excellent discussion in (Kwiatkowska & Zhang, 2023; König et al., 2024) for more details. *Black-box* verifiers leverage sampling and statistical estimation to provide probabilistic guarantees and do not require access to model parameters, making them applicable to closed-source models as well. **These scale to advanced models like LLMs** (Chaudhary et al., 2025). We note that a line of work based on randomized smoothing (Cohen et al., 2024) aims to provide probabilistic robustness guarantees on a *smoothed* model obtained through repeated sampling of DNN outputs over perturbed inputs. The inference cost of a smoothed model is orders of magnitude higher than that of the original DNN, making this technique less practical for building safe, accurate, and efficient models.

4. Training and Inference

DNNs trained with standard training often do not satisfy safety specifications as safety satisfaction is not part of their training objective. Adversarial or counter-example guided training augment the training data with violating examples during training, however, the trained models still cannot be proven to be safe in most cases (Madry et al., 2017). To overcome these limitations, *certified training* methods have been developed in recent years that directly incorporate the verifier computations within the training loop and generate models with a high degree of provability, i.e., they are more likely to satisfy specifications and are relatively easier to prove than DNNs obtained with competing methods (Mirman et al., 2018; Wong & Kolter, 2018).

In certified training, if the model f does not satisfy the specification, as checked by a verifier, its weights are updated to increase the provability. The gradient updates are derived by formulating a differentiable property loss on the verifier output, which measures how far the model is from satisfying the property. Since gradient updates are derived from the verifier code, their computations must be expressible as a differentiable function of model weights and parallelizable on GPUs for scalability. Overall, certified training can be seen as training f where the model updates are derived by differentiating the surrogate approximation of the DNN within φ, computed by the verifier.

While certified training improves the provability, safety specifications can be in conflict with accuracy. Using an imprecise verifier during training can result in overregularization and a significant reduction in the standard accuracy (Gowal et al., 2018). However, precise verifiers often have complicated code, which makes the optimization problem too

complicated to solve during training, yielding suboptimal results (Jovanovic et al., 2022). Also, employing a verifier during training is more expensive than when used for checking specifications on an already trained DNN, as now the verifier is called during every training iteration. Balancing the provability, accuracy, and cost is therefore the main challenge when developing state-of-the-art methods. Impressive results for wireless (Xu et al., 2024) and autonomous driving (Yang et al., 2023) have been achieved, where **trained models obtain high accuracy and provable safety**.

For generative models, *constrained generation* offers a powerful way to enforce formal properties directly during inference (Beurer-Kellner et al., 2024; Stoian & Giunchiglia, 2025). This has been used to eliminate syntax and semantic errors and improve the overall quality of structured outputs for applications in code generation (Ugare et al., 2025a;b), data serialization, symbolic reasoning tasks (Banerjee et al., 2025), and privacy-aware text generation that prevents revealing sensitive data (Ugare et al., 2025a).

5. Reliable Explanations and Interpretations

Popular methods for explaining DNN predictions identify relevant input features that influence the DNN output the most (Ribeiro et al., 2016; 2018). However, they do not give guarantees about the robustness of the generated explanations. Relying on non-robust explanations can lead to a false sense of confidence in an untrustworthy model. Recently, researchers have leveraged DNN verifiers to generate explanations with robustness guarantees, reliably improving DNN transparency (Wu et al., 2023; 2024).

DNN verifiers generate high-dimensional convex shapes at different layers, capturing complex relationships between neurons and DNN inputs to prove DNN safety. However, the individual neurons and inputs in the DNN do not have any semantic meaning, unlike the variables in programs, therefore it is not clear whether the safety proofs are based on any meaningful features learned by the DNN. If the DNN is proven to be safe, but the proof is based on meaningless features not aligned with human intuition, then the DNN behavior cannot be considered trustworthy.

Proof interpretation builds upon *proof features* computed by projecting the high-dimensional convex shapes onto individual neurons (Banerjee et al., 2024a). The proof features can be analyzed independently by generating the corresponding interpretations. Since certain proof features can be more important for the proof than others, a priority function over the proof features that signifies the importance of each proof feature in the complete proof is defined. The method extracts a set of proof features by retaining only the more important parts of the proof that preserve the property. Proof interpretations of DNNs trained with certified training methods (Zhang et al., 2020) yield novel insights showing that DNNs can satisfy robustness properties, but their behavior can still be untrustworthy. This observation suggests **the need to develop novel methods that train DNNs with trustworthy predictions and interpretations**.

6. Policy Recommendations

Safe deployment of increasingly more powerful and capable models requires rigorous safety assessment and construction. Empirical evaluations on standard benchmarks, while useful, are inherently limited — they offer only partial assessment of the risks and failure modes. Formal methods, by contrast, provide a principled foundation for systematically defining, verifying, and enforcing desirable safety properties, making them a critical component of the long-term solution.

From an AI governance perspective, the lack of formally defined and verifiable specifications for desirable properties — such as fairness, robustness, alignment, or catastrophic risks — poses a significant obstacle. Without precise formal definitions, it will be difficult to enforce meaningful rules or standards in practice. Policymakers and researchers need to work together to establish actionable, formalized definitions of key safety and ethical criteria that can guide both regulation and implementation.

Investing in formal methods has the potential to offer **transformative improvements to AI safety than marginal gains possible with incremental improvements to empirical methods**. However, realizing this vision requires a strategic, staged approach. Policies should aim to enable the **gradual integration of formal methods into DNN development pipelines**, rather than imposing unrealistic expectations upfront. In the short term, this means incentivizing and investing in the development of user-friendly, adaptable tools that prioritize ease of use and actionable insights over providing the strongest possible safety guarantees — which remain too ambitious given current tools and workflows. This pragmatic approach can set in motion a virtuous cycle: as formal methods become more approachable and useful, their adoption grows, paving the way for progressively stronger guarantees as the field matures.

As formal methods become more integrated, the demand for expertise will grow. There is currently a severe shortage of practitioners with expertise in formal methods, particularly in the context of deep learning. This skills gap threatens to slow adoption of formal methods just when it is most needed. To address this, policies and investments should focus on significantly expanding educational and training opportunities that make formal methods appear less daunting, funding interdisciplinary research and fostering collaborations between the formal methods, AI, and social sciences communities to build the necessary human capital.

References

Amato, F., López, A., Peña-Méndez, E. M., Vaňhara, P., Hampl, A., and Havel, J. Artificial neural networks in medical diagnosis. *Journal of Applied Biomedicine*, 11 (2), 2013.

Balunovic, M., Baader, M., Singh, G., Gehr, T., and Vechev, M. Certifying geometric robustness of neural networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Banerjee, D., Singh, A., and Singh, G. Interpreting robustness proofs of deep neural networks. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=Ev10F9TWML.

Banerjee, D., Xu, C., and Singh, G. Input-relational verification of deep neural networks. *Proc. ACM Program. Lang.*, 8(PLDI), June 2024b. doi: 10.1145/3656377. URL https://doi.org/10.1145/3656377.

Banerjee, D., Suresh, T., Ugare, S., Misailovic, S., and Singh, G. Crane: Reasoning with constrained llm generation. *CoRR*, abs/2502.09061, 2025. URL https://arxiv.org/abs/2502.09061.

Beurer-Kellner, L., Fischer, M., and Vechev, M. T. Guiding llms the right way: Fast, non-invasive constrained generation. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=pXaEYzrFae.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Brix, C., Bak, S., Johnson, T. T., and Wu, H. The fifth international verification of neural networks competition (vnn-comp 2024): Summary and results, 2024. URL https://arxiv.org/abs/2412.19985.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Chakravarthy, A., Narodytska, N., Rathis, A., Vilcu, M., Sharif, M., and Singh, G. Property-driven evaluation of rl-controllers in self-driving datacenters. In *Workshop on Challenges in Deploying and Monitoring Machine Learning Systems (DMML)*, 2022.

Chaudhary, I., Lin, S., Tan, C., and Singh, G. Specification generation for neural networks in systems. *CoRR*, abs/2412.03028, 2024. doi: 10.48550/ARXIV.2412.03028. URL https://doi.org/10.48550/arXiv.2412.03028.

Chaudhary, I., Hu, Q., Kumar, M., Ziyadi, M., Gupta, R., and Singh, G. Certifying counterfactual bias in LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HQHnhVQznF.

Chen, Y., Wang, S., Qin, Y., Liao, X., Jana, S., and Wagner, D. A. Learning security classifiers with verified global robustness properties. In *Proc. Conference on Computer and Communications Security (CCS)*, pp. 477–494. ACM, 2021.

Cohen, N., Ducoffe, M., Boumazouza, R., Gabreau, C., Pagetti, C., Pucel, X., and Galametz, A. Verification for object detection – ibp iou, 2024. URL https://arxiv.org/abs/2403.08788.

Dalrymple, D., Skalse, J., Bengio, Y., Russell, S., Tegmark, M., Seshia, S., Omohundro, S., Szegedy, C., Goldhaber, B., Ammann, N., Abate, A., Halpern, J., Barrett, C. W., Zhao, D., Zhi-Xuan, T., Wing, J. M., and Tenenbaum, J. B. Towards guaranteed safe AI: A framework for ensuring robust and reliable AI systems. *CoRR*, abs/2405.06624, 2024. doi: 10.48550/ARXIV.2405.06624. URL https://doi.org/10.48550/arXiv.2405.06624.

Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T. A., and Kohli, P. On the effectiveness of interval bound propagation for training verifiably robust models. *CoRR*, abs/1810.12715, 2018.

Guo, Z., Jin, R., Liu, C., Huang, Y., Shi, D., Supryadi, Yu, L., Liu, Y., Li, J., Xiong, B., and Xiong, D. Evaluating large language models: A comprehensive survey, 2023. URL https://arxiv.org/abs/2310.19736.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., and Murphy, K. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Huang, Y., Gupta, S., Xia, M., Li, K., and Chen, D. Catastrophic jailbreak of open-source llms via exploiting generation. *CoRR*, abs/2310.06987, 2023. doi: 10.48550/ARXIV.2310.06987. URL https://doi.org/10.48550/arXiv.2310.06987.

Jovanovic, N., Balunovic, M., Baader, M., and Vechev, M. T. On the paradox of certified training. *Trans. Mach. Learn. Res.*, 2022, 2022. URL https://openreview.net/forum?id=atJHLVyBi8.

Kabaha, A. and Drachsler-Cohen, D. Verification of neural networks' global robustness. *Proc. ACM Program. Lang.*, 8(OOPSLA1):1010–1039, 2024. doi: 10.1145/3649847. URL https://doi.org/10.1145/3649847.

Katz, G., Barrett, C., Dill, D., Julian, K., and Kochenderfer, M. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. In *Proc. 29th Int. Conf. on Computer Aided Verification (CAV)*, pp. 97–117, 2017.

König, M., Bosman, A. W., Hoos, H. H., and van Rijn, J. N. Critically assessing the state of the art in neural network verification. *J. Mach. Learn. Res.*, 25:12:1–12:53, 2024. URL https://jmlr.org/papers/v25/23-0119.html.

Krawiec, F., Peyton Jones, S., Krishnaswami, N., Ellis, T., Eisenberg, R. A., and Fitzgibbon, A. Provably correct, asymptotically efficient, higher-order reverse-mode automatic differentiation. *Proc. ACM Program. Lang.*, 6 (POPL), January 2022. doi: 10.1145/3498710. URL https://doi.org/10.1145/3498710.

Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial examples in the physical world. In *ICLR (Workshop)*. OpenReview.net, 2017.

Kwiatkowska, M. and Zhang, X. When to trust AI: advances and challenges for certification of neural networks. *CoRR*, abs/2309.11196, 2023. doi: 10.48550/ARXIV.2309.11196. URL https://doi.org/10.48550/arXiv.2309.11196.

Li, L., Qi, X., Xie, T., and Li, B. Sok: Certified robustness for deep neural networks. *CoRR*, abs/2009.04131, 2020. URL https://arxiv.org/abs/2009.04131.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., Newman, B., Yuan, B., Yan, B., Zhang, C., Cosgrove, C., Manning, C. D., Re, C., Acosta-Navas, D., Hudson, D. A., Zelikman, E., Durmus, E., Ladhak, F., Rong, F., Ren, H., Yao, H., WANG, J., Santhanam, K., Orr, L., Zheng, L., Yuksekgonul, M., Suzgun, M., Kim, N., Guha, N., Chatterji, N. S., Khattab, O., Henderson, P., Huang, Q., Chi, R. A., Xie, S. M., Santurkar, S., Ganguli, S., Hashimoto, T., Icard, T., Zhang, T., Chaudhary, V., Wang, W., Li, X., Mai, Y., Zhang, Y., and Koreeda, Y. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=iO4LZibEqW. Featured Certification, Expert Certification, Outstanding Certification.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Mirman, M., Gehr, T., and Vechev, M. Differentiable abstract interpretation for provably robust neural networks. In *Proc. International Conference on Machine Learning (ICML)*, pp. 3578–3586, 2018.

Müller, C., Serre, F., Singh, G., Püschel, M., and Vechev, M. T. Scaling polyhedral neural network verification on gpus. In *Proceedings of Machine Learning and Systems 2021, MLSys 2021, virtual, April 5-9, 2021*. mlsys.org, 2021.

Pan, L., Albalak, A., Wang, X., and Wang, W. Y. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. *CoRR*, abs/2305.12295, 2023. doi: 10.48550/ARXIV.2305.12295. URL https://doi.org/10.48550/arXiv.2305.12295.

Ribeiro, M. T., Singh, S., and Guestrin, C. "why should I trust you?": Explaining the predictions of any classifier. In *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144. ACM, 2016.

Ribeiro, M. T., Singh, S., and Guestrin, C. Anchors: High-precision model-agnostic explanations. In McIlraith, S. A. and Weinberger, K. Q. (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 1527–1535. AAAI Press, 2018. doi: 10.1609/AAAI.V32I1.11491. URL https://doi.org/10.1609/aaai.v32i1.11491.

Seshia, S. A. and Sadigh, D. Towards verified artificial intelligence. *CoRR*, abs/1606.08514, 2016. URL http://arxiv.org/abs/1606.08514.

Sherman, B., Michel, J., and Carbin, M. Lambdas: computable semantics for differentiable programming with higher-order functions and datatypes. *Proceedings of the ACM on Programming Languages*, 5(POPL):1–31, 2021.

Singh, A., Sarita, Y., Mendis, C., and Singh, G. Constraintflow: A DSL for specification and verification of neural network analyses. *CoRR*, abs/2403.18729, 2024. doi: 10.48550/ARXIV.2403.18729. URL https://doi.org/10.48550/arXiv.2403.18729.

Singh, A., Sarita, Y. C., Mendis, C., and Singh, G. Automated verification of soundness of dnn certifiers. *Proc. ACM Program. Lang.*, 9(OOPSLA1), April 2025a. doi: 10.1145/3720509. URL https://doi.org/10.1145/3720509.

Singh, G., Gehr, T., Püschel, M., and Vechev, M. An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages*, 3(POPL), 2019.

Singh, G., Laurel, J., Misailovic, S., Banerjee, D., Singh, A., Xu, C., Ugare, S., and Zhang, H. Safety and trust in artificial intelligence with abstract interpretation. *Foundations and Trends® in Programming Languages*, 8(3-4):250–408, 2025b. ISSN 2325-1107. doi: 10.1561/2500000062. URL http://dx.doi.org/10.1561/2500000062.

Sotoudeh, M. and Thakur, A. V. Provable repair of deep neural networks. In *PLDI '21: 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation, Virtual Event, Canada, June 20-25, 2021*, pp. 588–603. ACM, 2021.

Stoian, M. C. and Giunchiglia, E. Beyond the convexity assumption: Realistic tabular data generation under quantifier-free real linear constraints. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=rx0TCew0Lj.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. In *ICLR (Poster)*, 2014.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. In *proc. International Conference on Learning Representations, ICLR*. OpenReview.net, 2019.

Ugare, S., Gumaste, R., Suresh, T., Singh, G., and Misailovic, S. Itergen: Iterative semantic-aware structured llm generation with backtracking. In *ICLR*, 2025a.

Ugare, S., Suresh, T., Kang, H., Misailovic, S., and Singh, G. Syncode: Llm generation with grammar augmentation. *Trans. Mach. Learn. Res.*, 2025, 2025b.

Vega, J., Chaudhary, I., Xu, C., and Singh, G. Bypassing the safety training of open-source llms with priming attacks. In *The Second Tiny Papers Track at ICLR 2024, Tiny Papers @ ICLR 2024, Vienna, Austria, May 11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=nz8Byp7ep6.

Wang, B., Chen, W., Pei, H., Xie, C., Kang, M., Zhang, C., Xu, C., Xiong, Z., Dutta, R., Schaeffer, R., Truong, S. T., Arora, S., Mazeika, M., Hendrycks, D., Lin, Z., Cheng, Y., Koyejo, S., Song, D., and Li, B. Decodingtrust: A comprehensive assessment of trustworthiness in GPT models. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=kaHpo8OZw2.

Wang, S., Zhang, H., Xu, K., Lin, X., Jana, S., Hsieh, C.-J., and Kolter, J. Z. Beta-crown: Efficient bound propagation with per-neuron split constraints for complete and incomplete neural network verification. *arXiv preprint arXiv:2103.06624*, 2021.

Wong, E. and Kolter, J. Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *Proc. International Conference on Machine Learning, ICML*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5283–5292. PMLR, 2018.

Wu, H., Barrett, C., Sharif, M., Narodytska, N., and Singh, G. Scalable verification of gnn-based job schedulers. *Proc. ACM Program. Lang.*, 6(OOPSLA2), oct 2022.

Wu, M., Wu, H., and Barrett, C. Verix: towards verified explainability of deep neural networks. *Advances in neural information processing systems*, 36:22247–22268, 2023.

Wu, M., Li, X., Wu, H., and Barrett, C. Better verified explanations with applications to incorrectness and out-of-distribution detection, 2024. URL https://arxiv.org/abs/2409.03060.

Xu, C., Banerjee, D., Vasisht, D., and Singh, G. Support is all you need for certified vae training. In *The Thirteenth International Conference on Learning Representations*, 2024.

Yang, R., Laurel, J., Misailovic, S., and Singh, G. Provable defense against geometric transformations. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Zhang, H., Chen, H., Xiao, C., Gowal, S., Stanforth, R., Li, B., Boning, D., and Hsieh, C.-J. Towards stable and efficient training of verifiably robust neural networks. In *Proc. International Conference on Learning Representations (ICLR)*, 2020.