Verification & Testing of Deep Learning Models: A Systems Perspective

Alexandros Koliousis[†] and Stavros Tripakis[‡]

[†]Northeastern University London [‡]Khoury College of Computer Sciences

1 Description of work

As Artificial Intelligence (AI) systems grow more integrated with our daily lives, be it driving using Tesla's Autopilot or conversing with OpenAI's ChatGPT, the associated risks escalate. Modern AI systems are typically powered by neural networks—vision or language deep learning models—that we need to verify and test in order to provide a "safety net" for users.

Current neural network (NN) verifiers [1, 2] not only struggle with the sheer scale of contemporary deep learning models (*viz.*, their number of non-linear activations), but also with the generalization of verified properties when these models are deployed "in the wild." The atomic unit of NN verification is a *local robustness* property that states that a model's output is robust (*i.e.*, does not change) to **all** bounded perturbations of a **specific** input; but what are the interesting inputs and perturbations that best characterize the environment(s) under which the system is expected to operate correctly?

Our system perspective to verification and testing of deep learning models is that of a closed-loop system comprising of a NN *controller*, taking actions or making decisions based on NN's predictions, and an *environment* with which the controller interacts. The problem we tackle is to take the functional requirements that deem an AI system safe, expressed in a formal specification language or logic (high-level reasoning), and "imagine" new ways the controller can fail, expressed as a collection of local robustness properties that can be verified independently (low-level reasoning); and then, working backwards from low to high-level reasoning, provide global safety guarantees to the system based on a limited set of robust examples or counter-examples.

Our idea to AI system verification combines logic-based with probabilistic reasoning. We propose to express a safety property as a directed acyclic graph of logical decision nodes, each node representing a decision based on a variable or a composition of decisions and each edge a logical operation—akin to a sentential decision diagram [3]. Terminal nodes, in our case, represent local robustness properties that can be checked by an auxiliary NN verifier. But even with an ideal verifier —one that always returns true or false—local robustness verification is inherently probabilistic in nature because we have to **sample** inputs from our available data set to verify. Therefore, our graph does not only represent logi rules, but also represents random variables (nodes) and their conditional dependencies (edges). This dual view allows us to understand how local properties contribute to the global safety one, and it also permits us to compute global probabilistic guarantees. We identify three research challenges:

How to speficy relevant local robustness properties? Local robustness states that a transformation of an input (typically, an ℓ_{inf}-norm bounded perturbation of radius ϵ) does not alter a model's outcome. We can imagine a number of invariants or equivariants for a model (based on inductive biases) and its data set (based on data biases), in relation to color, scale, rotation, permutations, etc. Choosing and configuring one, however, should be dictated by the system's intended deployment environment and scenarios therein, which need to be specified in some language—like Scenic [4] does for autonomous driving agents.

How to sample inputs efficiently? In practice, we cannot verify a local robustness property on every possible input. But if we want stronger probabilistic guarantees, we cannot choose inputs randomly either. We need a heurestic to choose the next input to verify that balances *exploitation* and *exploration* for better coverage [5, 6]. Intuitively, if there are inputs whose predictions are uncertain, they are likely a good starting place to explore that uncertainty; but we should also choose inputs that explore all modes of the output distribution.

How to design our framework? We plan to build upon and extend state-of-the-art probabilistic logic programming frameworks, such as DeepProbLog [7], and NN verifiers, such as α, β-CROWN [1] and MN-BAB [2].

2 Project details

We discuss the multidisciplinary nature of our project team, our proposed timeline, and the significance of our project.

About the team. Our project creates a cross-Atlantic link between two research groups at Northeastern University: one in Boston, working on formal verification of safety-critical cyber-physical systems (led by Tripakis), and one in London, working on verification and testing of deep learning models (led by Koliousis). Yuhao Zhou, a Boston-based PhD student supervised by Tripakis (expected to complete ca. 2026), works on formal verification of NN-controlled systems [8]. Arooj Arif, a London-based first-year PhD student supervised by Koliousis (completion ca. 2027), works on a test framework for deep learning models. The pair is complemented by Zahra Dehghanipour, another first-year PhD student in London supervised by Koliousis (completion ca. 2027), who works on scalable verification of deep learning models—an orthogonal but very relevant topic to this project.

Project timeline. Starting 1 March 2024, we anticipate to complete our project within a year:

Months 1–3. We begin with intensive team workshops to derive a single end-to-end scenario that validates the efficacy of our framework (e.g., a simple NN adder of handwritten digits, verifying that for any two images x and y of numbers x and y, respectively, and digit recognition model $\mathcal{M}$, $\mathcal{M}(\mathsf{x}) + \mathcal{M}(\mathsf{y}) = x + y$).

Months 4–6. We focus on the design and implementation of a generalized framework. During this period we would like to use part of EAI's discretionary funds to promote student mobility—Yuhao vising London, or Arooj and Zahra visiting Boston.

Months 7–9. We focus on experimentation on Northeastern's Discovery cluster.

Months 10–12. We focus on writing, in time for ICML 2025 (ca. January 2025) and subsequent deadlines. During this last leg, Koliousis and Tripakis work on NSF 24-541 (ca. Jan 2025), NSF 24-509 (ca. Feb 2025), or similar (see §3).

Significance. Our proposed work contributes to EAI's offerings on *Responsible AI Services* and an *AI Ethics Advisory Board* by providing a rigorous framework to translate ethical guidelines and safety standards into practice. Our work adheres to President Biden's *Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence* to conduct red-teaming tests for safer AI. This funding will result in joint publications to top-tier venues that will, in turn, considerably strengthen our position in subsequent grant applications.

3 Pathways to funding

Once our collaboration is established, we aim to submit a grant proposal in response to relevant NSF solicitations including: (*i*) ACED: Accelerating Computing-Enabled Scientific Discovery (NSF 24-541); (*ii*) FMitF: Formal Methods in the Field (NSF 24-509); and (*iii*) calls similar to SLES: Safe Learning-Enabled Systems (NSF 23-562). We are also keen to mobilize our academic networks in Europe, to collaborate on proposals to Horizon Europe, and in the UK, to collaborate under the memorandum on research cooperation between NSF and UKRI (NSF 23-128).

References

[1] Zhang, H. *et al.* General cutting planes for bound-propagation-based neural network verification. In *NeurIPS* (2022).

[2] Ferrari, C., Mueller, M. N., Jovanović, N. & Vechev, M. Complete Verification via Multi-Neuron Relaxation Guided Branch-and-Bound. In *ICLR* (2022).

[3] Kisa, D., Van den Broeck, G., Choi, A. & Darwiche, A. Probabilistic Sentential Decision Diagrams. In *Principles of Knowledge Representation and Reasoning* (2014).

[4] Fremont, D. J. *et al.* Scenic: A Language for Scenario Specification and Data Generation. *Machine Learning Journal* (2022).

[5] Sun, Y. *et al.* Structural Test Coverage Criteria for Deep Neural Networks. *ACM Trans. Embed. Comput. Syst.* **18** (2019).

[6] Bengio, E., Jain, M., Korablyov, M., Precup, D. & Bengio, Y. Flow Network based Generative Models for Non-Iterative Diverse Candidate Generation. In *NeurIPS* (2021).

[7] Manhaeve, R., Dumancic, S., Kimmig, A., Demeester, T. & Raedt, L. D. DeepProbLog: Neural Probabilistic Logic Programming. In *NeurIPS* (2018).

[8] Zhou, Y. & Tripakis, S. Compositional Inductive Invariant Based Verification of Neural Network Controlled Systems. *arXiv:2312.10842* (2023).