Rapid advances in the reasoning and decision-making capabilities of machine learning (ML) models over the last decade—powered by the combination of deep neural networks (DNNs), vast amounts of data, and computational hardware suitable for highly parallel computation—has already led to learning-enabled systems being deployed in critical areas of society such as transportation networks, healthcare, financial systems, industrial manufacturing, and even government and judicial decision-making. As the use of learning-enabled systems proliferates, there is an urgent need to ensure that these systems are deployed in safe and socially beneficial ways. DNNs, however, are well-known to be **brittle**—slight changes in the operating environment of these systems compared to the environment used for training them can lead to drastic degradation in their performance—and **opaque**—the logic used internally by the DNNs to perform their reasoning is hard to discern from their structure and parameters. My work advances principled techniques to address the brittleness and opacity of DNNs and for analyzing the behavior of learning-enabled systems to evaluate their safety.

My research draws on techniques and perspectives from the **formal methods** literature, which presents powerful tools for systematically analyzing the behavior of software systems. Formal methods view programs as formal, symbolic objects that can be logically analyzed to ascertain if they meet their behavioral specifications. I have extensive experience in applying and extending formal methods for the analysis of traditional software systems [1, 2, 3, 4, 5, 6, 7, 8, 9]. My ongoing work on improving the safety of learning-enabled systems using these methods can be broadly categorized into:

- **Certifying the Robustness of DNNs**: One way to formally analyze the brittleness of DNNs is using the notion of *local robustness*. A DNN $f \in X \to Y$ is locally robust at an input $x \in X$ if $\forall x' \in X. ||x - x'|| \leq \epsilon \implies f(x) = f(x')$. In my work, I have discovered flaws in existing popular methods for certifying local robustness of DNNs [10, 11], have proposed new approaches for local robustness certification [12, 13, 14], and have clarified the role of the local robustness guarantee in the context of overall system safety [15].

- **Understanding DNN Reasoning by Inferring Concept Representations**: A first step towards understanding the logical reasoning used internally by DNNs is to infer the relationship between the neuron values at intermediate DNN layers and high-level human-understandable concepts (for instance, colors or shapes in an image). I have worked on techniques for extracting such concept representations expressed as logical constraints over neuron values [16]. I have also demonstrated how such representations can help improve the engineering of DNNs by enabling fundamental software engineering activities such as automated testing, debugging, requirements analysis, and formal verification.

- **Analyzing the Safety of Learning-Enabled Systems**: While logical specifications of safe behavior are hard to state for DNNs in isolation because of their data-driven nature, they are more readily available for learning-enabled systems with DNN components. However, formal safety analysis of such systems is challenging; the environments in which these systems operate can be difficult to model mathematically and existing analysis algorithms do not scale to these complex systems. I have developed new sound, probabilistic [17, 18] and worst-case [19] abstractions that enable formal analysis of learning-enabled cyber-physical systems such as autonomous robots. These abstractions not only enable analysis but also allow synthesis of provably safe controllers for closed-loop systems that use DNNs for perception. I have also developed efficient techniques for repairing DNN outputs at run-time [20] in order to ensure safe operation of learning-enabled systems.

In general, my research in the emerging area of **Trustworthy AI** sits at the intersection of **formal methods** and **machine learning**. The interdisciplinary nature of this research and the leading role played by industry in pushing the boundaries of learning-enabled systems necessitates a collaborative approach, involving researchers from diverse backgrounds and industry professionals, to tackle the problems. As a postdoctoral researcher in the Carnegie Mellon University Security and Privacy Institute, I have already built strong connections with multiple CMU faculty members as well as with researchers from University of York, UIUC, U Wisconsin-Madison, NASA, Boeing, SRI, and VMware Research, and have multiple ongoing research projects with these collaborators. As a faculty member, I will continue developing and strengthening connections with researchers from academia and industry, and ensure that while my research is built on solid theoretical ideas, it is also grounded in the practical challenges raised by the most complex learning-enabled systems being built by industry.

Vision

We find ourselves in a moment where it seems very likely that future software, cyber-physical, and socio-technical systems will regularly incorporate AI/ML components as part of their architecture. My vision is to develop the

mathematical theory and computational tools needed to be able to efficiently and automatically analyze the safety of such complex, real-world learning-enabled systems deployed in safety-critical settings. This requires developing new techniques for testing and verifying computational systems, and I hope to leverage the decades of research on engineering trustworthy traditional software systems for this challenge. However, unlike traditional systems where it is at least theoretically feasible to seek a fully formal proof of safety, the lack of specifications for DNNs and their opaque nature necessitates new kinds of analyses for learning-enabled systems that fuse formal proofs with empirical evidence. I will develop the formal foundations of such safety arguments. I will also continue my work on discovering and addressing the brittleness of DNNs, specially for new families of powerful models such as large language models (LLMs) and multi-modal models. Finally, I believe that resolving the opacity of DNNs, and understanding the internal computational structures and abstractions that they learn is one of the grand challenges of our times. Making progress on this problem requires adopting a perspective that, so far, has belonged to the realm of neuroscience. I envision a future where we have automated tools to analyze and understand the internals of DNNs, and will continue my ongoing research on understanding DNNs to make progress towards this future.

Certifying the Robustness of DNNs

Attacks. Ensuring local robustness of DNNs has proved to be a hard challenge. Although DNNs can achieve state-of-the-art classification accuracies on a variety of important tasks, DNN classifiers with comparable certified robust accuracies[1] remain elusive, even when trained in a robustness-aware manner. Consequently, a number of post-training approaches have been proposed to reduce the brittleness of DNNs. One popular approach is to certify local robustness at run-time (inference-time). The DNN abstains from prediction if it cannot be certified as locally robust at the given, possibly perturbed, input. This ensures that the DNN is only used for prediction at inputs where it is guaranteed to not be subject to perturbation attacks. However, through my work [10], I have shown that adversaries can exploit such run-time checks to force the DNN to unnecessarily abstain from prediction and drastically reduce model utility. Another common approach for improving robustness is to use an ensemble of DNNs for prediction. I discovered that the popular *cascading ensemble* approach is unsound [14], i.e., when a cascading ensemble is certified as locally robust at an input x, there can, in fact, be inputs x' in the ϵ-ball centered at x, such that the cascade's prediction at x' is different from x. I also demonstrated how adversaries can exploit this unsoundness to attack cascading ensembles.

Defenses. I have contributed to not only identifying attacks but also to developing new certification mechanisms. In recent work [14], I presented a new run-time robustness certification mechanism that balances efficiency and precision by combining under-approximate, over-approximate, and exact local robustness certifiers. A weakness of the local robustness specification is that it only provides a guarantee about the *local* behavior of the model. In order to provide a guarantee on global model behavior, I have proposed a probabilistic notion of robustness that I named *probabilistic Lipschitzness*. Assuming a distribution over model inputs, the property requires that for any randomly sampled pair of inputs, with a high probability, it should be the case that the difference between the outputs of the model (assuming continuous outputs which would be the logits in the case of a classifier) is bounded by the difference between the inputs. I designed a probabilistic static analysis algorithm, based on abstract interpretation, for certifying DNNs with respect to this property [12, 13]. Recently, I have also written an expository article [15] that clarifies the nature of the local robustness guarantee from the perspective of global model behavior and makes the case that research on training and certification procedures for local robustness continues to be important even though the progress on training robust models has been slow.

Future Directions. The problem of learning DNN classifiers with high certified robust accuracies remains unsolved even after intensive efforts by the community over the last few years. The problem has been exacerbated with the growing popularity of generative models such as large language models (LLMs). LLMs are not only susceptible to misclassifications due to small perturbations but they also exhibit new failure modes. They are susceptible to *jailbreaking* attacks that cause them to ignore their safety guardrails and generate undesirable text. They are also known to *hallucinate*, i.e., make up false information. While it is important to develop training-time interventions that can help learn models without these susceptibilities, progress on such solutions has been slow. In the meanwhile, organizations have shown willingness to deploy vulnerable models. Run-time mechanisms are a promising approach to harden existing DNNs against different kinds of attacks. These mechanisms decide whether to abstain from prediction or not based on the current input, output, and internal model state. In my

[1] Percentage of inputs where the classifier is accurate and certified as locally robust.

past research, I have already explored the use of run-time mechanisms for hardening image classifiers against perturbation attacks, and I am excited to develop new solutions that can harden LLMs.

I am also interested in studying the brittleness of code models—ML models designed to solve coding tasks—to adversarial perturbations and developing defenses for such models. In an ongoing project [21], I am studying the susceptibility of LLMs such as GPT-3.5, GPT-4, Claude, and CodeLlama to semantics-preserving adversarial code perturbations when used for code summarization (i.e., the task of generating the function name given the function body). Initial results suggest that these models are highly susceptible to perturbations but run-time defenses based on "unperturbing" the code inputs can be effective. Unlike images, it is possible to leverage static analyses to transform semantically-equivalent code into a canonical form that is immune to perturbations.

Understanding DNN Reasoning by Inferring Concept Representations

Unlike traditional software applications whose logic is driven from input-output specifications, DNNs are inherently *opaque*, as their logic is learned from examples of input-output pairs. The lack of high-level abstractions makes it challenging to interpret the logical reasoning employed by a DNN and hinders the use of standard software engineering practices such as automated testing, debugging, requirements analysis, and formal verification that have been established for producing high-quality software.

I have worked on addressing this challenge by proposing a *concept-guided* approach to neural network engineering [16]. My work draws on the insight that, in a typical DNN, the early layers extract the important concepts from the inputs while the later dense layers encode the symbolic, decision-making logic in terms of these concepts. The proposed approach therefore first extracts high-level, human-understandable concept representations from the trained DNN. This enables us to reason about the DNN through the lens of the concepts and to drive the aforementioned software engineering activities.

The concept representations associate neuron values at the intermediate layers with higher-level abstractions that have clear semantic meaning (e.g., shapes in an image). These high-level concept representations have been empirically observed to take the form of *logical rules* ($\texttt{pre} \implies \texttt{post}$) where the precondition ($\texttt{pre}$) describes a geometric shape (typically, a convex region or a halfspace characterized by a direction) in the latent space defined by an internal layer of the neural network and the postcondition ($\texttt{post}$) denotes the presence (or absence) of the concept. These formal, checkable rules enable evaluating the quality of the datasets, retrieving and labeling new data, understanding scenarios where models make correct and incorrect predictions, detecting incorrect (or out-of-distribution) samples at run-time, and verifying models against human-understandable requirements.

Future Directions. Understanding how DNNs represent high-level concepts and what algorithms they use internally for solving computational tasks remain challenging open questions that I am very keen to investigate. Towards the goal of understanding the concept representations learned by a DNN, I have an ongoing project where we are trying to leverage CLIP, a state-of-the-art multi-modal (vision and language) model, as a powerful lens for analyzing and understanding DNN latent spaces. This research is aided by recent empirical observations that the latent spaces of various high-quality DNNs (such as ResNets, Vision Transformers, CLIP, LLMs, and Diffusion Models) can be mapped to each other via simple linear maps. This opens up powerful new capabilities for leveraging the high-quality representations learnt by foundation models for analyzing other, smaller models. I intend to continue exploring this direction in my future research. I also plan to develop formal methods that leverage the knowledge of concept representations for analyzing the behavior of DNNs with respect to specifications expressed in terms of high-level, human-understandable concepts.

Towards the goal of understanding the algorithms internally used by DNNs, I have another ongoing project where we have trained a small model with an attention-based Transformer architecture to solve the 2-SAT problem for instances with a fixed number of clauses and variable (10 and 5, respectively), and are attempting to reverse engineer the algorithm that the model has learnt for this task. This style of analysis, referred to as *mechanistic interpretability* in recent literature, is in its infancy and currently performed in a fairly ad-hoc manner. I am interested developing more rigorous tools and techniques, inspired by formal methods and static program analysis frameworks, that can help reliably undertake such analyses on a variety of models.

Analyzing the Safety of Learning-Enabled Systems

Autonomous systems such as self-driving vehicles, social robots, and recommendation systems are meant to operate in complex physical and digital environments with unknown dynamics. They need to perceive and reason about their environments using high-dimensional data streams (such as images) generated by rich sensors (such

as cameras) and increasingly use DNNs for this purpose. These systems are closed-loop, with components that perceive, act, and update the state of the system as well as the environment. Formally analyzing the safety of these systems is challenging because of the complexity of the system components—the DNNs can have millions to billions of parameters, and mathematical models of other components such as the environment in which the system operates and the sensors are highly complicated.

I have developed symbolic, compositional techniques [17, 18, 19] for formal safety analysis of learning-enabled autonomous systems that addresses the above challenges and have used it to analyze an experimental autonomous system developed by Boeing for guiding airplanes on taxiways. The decomposition of this system is illustrated in Fig 1. The key insight for dealing with the complexity is to conservatively abstract away the hard-to-analyze components, namely, the DNN-based perception (p) and environment dynamics (e), and replace their composition ($p \circ e$) with either probabilistic or worst-case abstractions. The resulting system becomes amenable to formal verification using off-the-shelf (probabilistic) model checking tools. Furthermore, the artifacts from the verification are leveraged to improve the DNN, resulting in increased performance and safety.

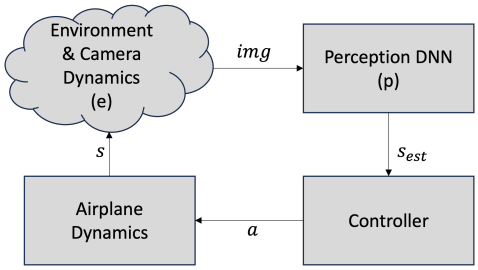

Figure 1: Closed-loop System

The abstractions I have proposed not only enable analysis but also allow synthesis of provably safe controllers for closed-loop systems that use DNNs for perception [17]. I have worked on synthesizing such provably safe controllers for a system intended to maintain driver alertness levels in a shared control Level 3 autonomous car as well for a system to avoid collisions between mobile service robots. I have also designed efficient techniques for repairing DNN outputs at run-time [20] in order to ensure safe operation of ML-enabled systems.

Probabilistic Analysis. The probabilistic abstraction [18] yields probabilistic safety guarantees about the system with respect to safety properties expressed in probabilistic computation tree logic. The key idea is to replace the DNN, sensors, and environment with a compact probabilistic abstraction built from *confusion matrices* computed for the DNN on representative data sets. Developers routinely use confusion matrices to evaluate DNNs, so the analysis approach is closely aligned with existing work-flows, facilitating its adoption in practice. The approach is *compositional*—the probabilistic component is computed separately from the rest of the system. The size of the probabilistic abstraction is linear in the size of the output of the DNN, and is independent of the number of the DNN parameters or the complexity of the sensors and the environment. The approach also leverages local, DNN-specific analyses, such as certifiers for local robustness, as run-time guards to refine the abstractions and increase the safety of the overall system. As the probabilities in the abstraction are estimated based on empirical data, they are subject to error. I have explored the use of *confidence intervals*, in addition to point estimates for these probabilities, to strengthen the guarantees of the analysis.

Worst-Case Analysis. The worst-case abstraction [19] helps provide *non-probabilistic* guarantees for system-level safety properties, expressed as finite state automata or constraints in (fluent) linear temporal logic. The idea here is take an abductive reasoning approach and analyze the system in the absence of the DNN, the sensors, and the environment, assuming their worst-case behaviour. The analysis generates *assumptions* encoding *sequences* of DNN predictions that guarantee system-level safety. The assumptions are the weakest in the sense that they characterize the output sequences of all the possible DNNs that, plugged into the autonomous system, satisfy the properties. These assumptions can be leveraged as *run-time monitors* over a deployed DNN to guarantee the safety of the overall system. Moreover, the assumptions can be mined to extract local properties on DNN behavior, which in turn can be used for the separate testing and neuro-symbolic training of the DNNs.

Future Directions. There are many open problems in applying formal analyses to learning-enabled systems deployed in practice. First, the specifications of safety are non-obvious, specially for learning-enabled systems deployed in socio-technical settings such as recommendation systems, healthcare, judicial decision-making, and financial systems. Second, while my research so far has focused on systems with discrete dynamics, many systems can be faithfully modeled only with continuous dynamics. Third, complicated systems such as autonomous cars use a hierarchy of components, namely, route planners, behavioral decision-makers, motion planners, and low-level controllers. Such systems are beyond the scope of the techniques I have developed so far. I am very eager to address these open questions. Designing suitable abstractions is essential for extending formal analyses to these settings, so my current work presents a natural starting point for tackling these problems. Longer term, I am interested in developing the formal foundations of safety arguments that fuse formal proofs with empirical evidence (for instance, about the behavior of components such as DNNs).

References

(* indicates equal contribution, α indicates alphabetical ordering)

[1] **Ravi Mangal**, Mayur Naik, and Hongseok Yang. A Correspondence Between Two Approaches to Interprocedural Analysis in the Presence of Join. In *Proceedings of the 23rd European Symposium on Programming Languages and Systems - Volume 8410*, ESOP '14, 2014.

[2] Xin Zhang, **Ravi Mangal**, Radu Grigore, Mayur Naik, and Hongseok Yang. On Abstraction Refinement for Program Analyses in Datalog. In *Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI '14, 2014.

[3] Xin Zhang, **Ravi Mangal**, Mayur Naik, and Hongseok Yang. Hybrid Top-down and Bottom-up Interprocedural Analysis. In *Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI '14, 2014.

[4] **Ravi Mangal**, Xin Zhang, Mayur Naik, and Aditya V. Nori. Solving Weighted Constraints with Applications to Program Analysis. Technical report, Georgia Institute of Technology, 2015.

[5] **Ravi Mangal**, Xin Zhang, Aditya V. Nori, and Mayur Naik. A User-guided Approach to Program Analysis. In *Proceedings of the 10th Joint Meeting on Foundations of Software Engineering*, ESEC/FSE '15, 2015.

[6] **Ravi Mangal**, Xin Zhang, Aditya V. Nori, and Mayur Naik. Volt: A Lazy Grounding Framework for Solving Very Large MaxSAT Instances. In *International Conference on Theory and Applications of Satisfiability Testing*, SAT '15, 2015.

[7] Xin Zhang, **Ravi Mangal**, Aditya V. Nori, and Mayur Naik. Query-guided Maximum Satisfiability. In *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '16, 2016.

[8] **Ravi Mangal**, Xin Zhang, Aditya Kamath, Aditya V. Nori, and Mayur Naik. Scaling Relational Inference Using Proofs and Refutations. In *Thirtieth AAAI Conference on Artificial Intelligence*, AAAI '16, 2016.

[9] Sulekha Kulkarni, **Ravi Mangal**, Xin Zhang, and Mayur Naik. Accelerating Program Analyses by Cross-program Training. In *Proceedings of the 2016 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications*, OOPSLA '16, 2016.

[10] Klas Leino*, Chi Zhang*, **Ravi Mangal***, Matt Fredrikson, Bryan Parno, and Corina Păsăreanu. Degradation Attacks on Certifiably Robust Neural Networks. *Transactions on Machine Learning Research*, 2022.

[11] **Ravi Mangal***, Zifan Wang*, Chi Zhang*, Klas Leino, Corina Păsăreanu, and Matt Fredrikson. On the Perils of Cascading Robust Classifiers. In *International Conference on Learning Representations*, ICLR '23, 2023.

[12] **Ravi Mangal**, Aditya V. Nori, and Alessandro Orso. Robustness of Neural Networks: A Probabilistic and Practical Approach. In *Proceedings of the 41st International Conference on Software Engineering: New Ideas and Emerging Results*, ICSE-NIER '19, 2019.

[13] **Ravi Mangal**, Kartik Sarangmath, Aditya V. Nori, and Alessandro Orso. Probabilistic Lipschitz Analysis of Neural Networks. In *International Static Analysis Symposium*, SAS '20. Springer, 2020.

[14] **Ravi Mangal** and Corina Păsăreanu. A Cascade of Checkers for Run-time Certification of Local Robustness. In *5th Workshop on Formal Methods for ML-Enabled Autonomous Systems*. Springer, 2022.

[15] **Ravi Mangal***, Klas Leino*, Zifan Wang*, Kai Hu*, Weicheng Yu, Corina Păsăreanu, Anupam Datta, and Matt Fredrikson. Is certifying ℓ_p robustness still worthwhile? *arXiv preprint arXiv:2310.09361*, 2023.

[16] (α) Divya Gopinath, Luca Lungeanu, **Ravi Mangal**, Corina Păsăreanu, Siqi Xie, and Huafeng Yu. Feature-guided Analysis of Neural Networks. In *Fundamental Approaches to Software Engineering*, FASE'23. Springer, 2023.

[17] (α) Radu Calinescu, Calum Imrie, **Ravi Mangal**, Genaína Nunes Rodrigues, Corina Păsăreanu, Misael Alpizar Santana, and Gricel Vázquez. Discrete-event Controller Synthesis for Autonomous Systems with Deep-learning Perception Components. *arXiv preprint arXiv:2202.03360*, 2022.

[18] Corina S Păsăreanu, **Ravi Mangal**, Divya Gopinath, Sinem Getir Yaman, Calum Imrie, Radu Calinescu, and Huafeng Yu. Closed-loop analysis of vision-based autonomous systems: A case study. In *International Conference on Computer Aided Verification*, pages 289–303. Springer, 2023.

[19] Corina Păsăreanu, **Ravi Mangal**, Divya Gopinath, and Huafeng Yu. Assumption generation for the verification of learning-enabled autonomous systems. In *International Conference on Runtime Verification*. Springer, 2023.

[20] Klas Leino, Aymeric Fromherz, **Ravi Mangal**, Matt Fredrikson, Bryan Parno, and Corina Păsăreanu. Self-correcting Neural Networks for Safe Classification. In *5th Workshop on Formal Methods for ML-Enabled Autonomous Systems*. Springer, 2022.

[21] Chi Zhang, Zifan Wang, **Ravi Mangal**, Matt Fredrikson, Limin Jia, and Corina Păsăreanu. Transfer attacks and defenses for large language models on coding tasks. *arXiv preprint arXiv:2311.13445*, 2023.