Stanford Center for AI Safety

Clark Barrett, David L. Dill,
Mykel J. Kochenderfer, Dorsa Sadigh

1 Introduction

Software-based systems play important roles in many areas of modern life, including manufacturing, transportation, aerospace, and healthcare. However, developing these complex systems, which are expected to be smart and reliable, is difficult, expensive, and error-prone. A key reason for this difficulty is that the sheer complexity of many systems keeps growing, making it increasingly difficult for human minds to form a comprehensive picture of all relevant elements and behaviors of the system and its environment.

To mitigate this difficulty, research in the field of artificial intelligence (AI) has been promoting a different approach to programming. Instead of having a human engineer provide program logic for handling all possible inputs, algorithms are given a set of training examples — typically (input, output) pairs — from which they automatically extrapolate a software implementation. The learned model is then able to generalize and produce desirable outputs, even for previously-unseen inputs. Modern AI techniques are increasingly scalable and efficient, and over the coming decade, AI-based systems will continue to be deployed in more and more real-world settings.

A key difficulty, however, is that we are currently unable to *reason* about AI systems. Indeed, we understand quite well the algorithms used for *training* them — this topic has been studied extensively — but, given a trained AI system, we have no way to make rigorous claims about its behavior. In classical, imperative programing one can often look at and reason about the code, write invariants, and prove certain properties of the system (either manually or automatically). Because such code is written by humans, good software engineering practices coupled with formal methods can ensure that it is also guaranteed to perform as expected. In machine-learned systems, however, the program amounts to a highly complex mathematical formula for transforming inputs into outputs. Humans can barely parse the formulas defining these systems, let alone reason about them. And off-the-shelf formal tools are so far able to reason about only very small instances of such systems. Currently, we have little recourse but to blindly trust that the training algorithms were sufficiently "clever" and have produced a system that is correct. However, if we are to use AI components in safety-critical systems, this situation is unsatisfactory.

2 Mission of the Center for AI Safety

The goal of the center for AI safety at Stanford is to play a leadership role in addressing this critical situation:

> The mission of the Stanford Center for AI Safety is to develop rigorous techniques for building safe and trustworthy AI systems and establishing confidence in their behavior and robustness, thereby facilitating their successful adoption in society.

3 Research Directions

Below, we outline some of the main research thrusts that we plan to pursue in order to facilitate the goal of having safe and reliable AI-based systems.

3.1 Formal Techniques for AI Safety

The term *formal methods* refers to a broad set of techniques for using precise mathematical modeling and reasoning to draw rigorous conclusions about complex systems. Formal methods are regularly used to ensure the safety, security, and robustness of conventional software and hardware systems, especially those that are used in safety-critical applications. A key area of focus will be to develop and adapt formal methods for AI-based systems.

Formal specifications for systems with AI components. AI components are present in many of today's autonomous and intelligent systems and can inevitably affect the safety, assurance, fairness, and performance of these systems interacting with uncertain and dynamic environments. For instance, autonomous cars use deep neural networks to classify and detect obstacles or pedestrians on a road; AI techniques are used in healthcare for diagnosis and in developing algorithms for medical devices; and domestic robots and assistive devices leverage AI algorithms to safely interact with humans. To provide any correctness guarantees for such systems, we first need to understand and formalize the desired, unexpected, or malicious behaviors that could be produced by these systems. These properties may specify the functionality of the inner AI components by defining their input-output behavior. Alternatively, the properties may be at the level of the overall system that encompasses multiple AI components interacting with one another and with other decision-making components. A challenging characteristic of such complex systems is the interplay between hardware, software, and algorithms, which requires analyzing safety of the AI-based systems at all levels. One goal of the center is to formally specify desirable, unexpected, and malicious properties of these systems.

Another goal is to understand the trade-offs between safety and other desirable properties. For instance, an unmanned aircraft needs to decide between safely exploring the space and achieving other objectives such as flying in a stable and efficient manner towards its destination. Similarly, an autonomous car needs to arbitrate between the safety of the vehicle and the

comfort and efficiency of the trip. An assistive robot must balance the active gathering of information about the intent of its user with the safety and expressiveness of its actions. We are exploring mathematical formalizations of properties such as safety, fairness, reliability, robustness, explainability, and efficiency, with the goal of developing formal techniques that are capable of addressing these specifications.

Formal verification of systems with AI components Given a specificaion, the next step is to develop tools and algorithms that can *verify* the correctness of machine-learned software with respect to a specification. This means checking that the specification holds for *every* possible input to the system. The ability to do this opens the door to reasoning about machine-learned systems in many ways. For instance, we could ask: "given a machine-learned program for driving a car, is it possible that if a person is crossing the street ahead, the car will not decelerate?" The automatic algorithm will be required to decide, for all possible situations involving a car and a pedestrian, whether it is possible for the car not to decelerate. The result will be either a conclusion that this is impossible, or a *counter-example* — a specific scenario — for which the violation occurs. Another example in flight collision avoidance would be "is it possible that two aircraft are dangerously close to each other, and yet the system does not recommend to the pilots to steer away?"

As a first step in this direction, we have developed an algorithm, called Reluplex, capable of proving properties of deep neural networks (DNNs) or providing counter-examples if the properties fail to hold. The algorithm handles DNNs with the Rectified Linear Unit (ReLU) activation function. A naive approach to this problem is to analyze separately the two cases when the input of each ReLU is negative (when the output of the ReLU is constant) and non-negative (when the output is equal to the input), leading to an exponential explosion of combinations. Unlike previous attempts to verify DNNs, the Reluplex algorithm is designed to delay or avoid this case analysis. Reluplex can solve problems that are an order of magnitude larger than was previously possible. Ongoing work aims to further improve the scalability of the Reluplex approach, to extend it to handle a broader class of activation functions and network topologies, and to use it in collaboration with AI system developers to verify real systems.

Analysis of adversarial robustness The trend of deploying DNNs as controllers of key systems has raised questions regarding their security. Whereas security issues in traditional software have been extensively studied (and still dramatic issues are being discovered), the question of security for systems with DNNs is largely new, and could have serious implications unless addressed.

One notable example is that of *adversarial examples*, small adversarial perturbations applied to correctly-classified inputs that can "fool" a DNN into misclassifying them. Many state-of-the-art DNNs have been shown to be susceptible to this phenomenon and many strategies have been developed to train DNNs that are more robust to adversarial examples.

Here, too, verification can provide an invaluable tool for improving network security — in particular in the context of adversarial examples. One can phrase the problem of finding adversarial examples as a verification problem, and use a verification tool to prove that *no adversarial examples exist* for given input domains and allowed amounts of perturbation. This makes it possible to measure the effectiveness of defensive techniques in an objective way that does not depend on attack techniques currently in existence. We aim to continue exploring general techniques that will aid in understanding and addressing issues of adversarial robustness.

Automatic test-case generation Providing interesting and realistic test-cases can be a challenging problem for systems with AI-based components. Today, most AI-based systems depend on large datasets for training and testing. However, the size of the dataset alone is not a predictor of how well the system performs. For instance, one might make a statement about safety of autonomous cars based on the number of miles the car has driven. However, just reaching a certain number of miles is not enough to ensure the safety of the vehicle. For example, if all the miles are driven on the same highway, the car has not seen more challenging driving scenarios such as difficult intersections or roundabouts and would thus not be able to reason about these scenarios. We would like to systematically test and validate such complex systems by generating challenging scenarios to specifically test the AI components, the input-output behavior of an AI component used as part of the more complex system, and the interplay of the components with each other and with the larger system. As part of our center, we plan to explore active learning techniques along with formal methods to automatically generate interesting test-cases that help with verification and validation of AI-based systems. In addition, formal techniques have the potential to provide scalable *model checking* algorithms that can help with verification of desired properties in large state space systems such as autonomous cars interacting with complex environments.

3.2 Learning and Control for AI Safety

Safe exploration and learning for better perception by AI systems A common characteristic of AI agents is their ability to update their models and adapt to changes in the environment. This adaptibility requires actively or passively gathering information about the world. For instance, a quadcoptor might not know the exact weight of its payload, but by applying various control inputs (e.g. thrust, yaw, pitch, roll) it can gain confidence about this value. However, such explorations could put the vehicle itself at the risk of becoming unstable and could also lead to a violation of safety constraints. As part of the center, we will explore situations in which safety and exploration objectives can be in conflict with each other. Balancing exploration and exploitation has been a long-standing problem in AI. Our goal is to design systems that intelligently and safely balance learning about the uncertainties of the environment with exploitation of safety knowledge in order to develop better perception for

autonomous systems in a provably safe manner. This trade-off becomes even more challenging in multi-agent settings, where multiple AI-based systems must collaborate in a dynamic environment to safely explore the uncertainty in the environment or in the autonomous agents themselves. In addition, there is a strong link between these ideas of exploration and exploitation and the coupling of perception and planning for autonomous agents. Active learning methods are commonly leveraged to efficiently gather information about the environment for better perception and planning. We plan to study such techniques for an efficient coupling of perception and planning through safe learning.

Safe control of AI agents Controlling an agent safely requires reasoning about the uncertain effects of the agent's decisions on operational objectives and safety constraints. The agent generally relies on imperfect sensor information, which results in uncertainty about the current state of the world. The effects of the agent's actions are also difficult to predict, though we may be able to learn probabilistic models from data or construct them from expert judgment. Designers of AI systems often have to make challenging trade-offs between safety and operational performance objectives. We will explore methods for building flexible models for sensors, dynamics, and objectives along with computational techniques for using these models to generate safe control strategies for AI agents. Focusing on coupling perception and planning, we believe safe and robust control and optimization techniques are required to guarantee correctness of safety properties in uncertain and dynamic environments. We plan to combine our planning methods with safe learning strategies that decide on safe and informative actions for intelligent and autonomous agents. Through our center, we plan to bridge the gap between various methods that in some way address safety specifications, such as robust and adaptive control, learning and optimization, and reactive synthesis from logical specifications.

3.3 Transparency for AI Safety

Explainable, accountable, and fair AI As we have seen in recent years, many AI-based systems have been under scrutiny due to lack of transparency and explainability. AI-based systems can, for example, exaggerate social bias. They can also provide outcomes that locally optimize a specific desirable objective, but that when generalized can result in unfair and unexpected outcomes. Such outcomes can be due to issues such as reward misalignment, reward hacking, and negative side effects. These issues are usually studied in the setting of safety for Artificial General Intelligence (AGI). For example, we can design an autonomous car that is rewarded for changing lanes and avoiding collisions. However, the vehicle needs to balance between how much we care about changing lanes immediately as opposed to keeping distance with the vehicles in the destination lane. For specific reward functions, we might observe *conservative* behavior where the autonomous vehicle never changes lanes, or we might observe more *risk-taking* behavior where the autonomous

car changes lanes with not enough margins between the vehicles. The design of reward functions is a fundamental element in transparency, explainability, and safety of autonomous systems. As part of this center, we plan to focus on specific concerns about transparency and explainability of AI systems, by building algorithms that can provide reasons and explanations for their actions. We will look into understanding features of learning-based systems, and robustness analysis of optimization based methods used in learning and control.

In addition, we plan to study the safety and fairness implications of AI systems that optimize a local reward function. For instance, local planning by autonomous cars can result in efficient local interactions between the vehicles. However, the larger implications of these interactions for the traffic network must be addressed in parallel, e.g., how do autonomous cars affect the congestion on roads? What routing algorithms do autonomous cars need to use for efficient mixed-autonomy networks? What routing algorithms should ride-sharing companies use to address fairness and safety issues in a city? These issues are exacerbated when the systems are composed of deep neural networks. As part of our center, we plan to study safety in the context of fairness, accountability, and explainability for autonomous and intelligent systems that are composed of learning-based components.

Diagnosis and repair for systems with AI components Although we will explore better learning and control for autonomous intelligent systems, there is no guarantee that AI agents will always be capable of arriving at a safe solution. There are many situations in which a safe strategy is not feasible in a particular environment. For instance, an autonomous vehicle might not be able to decide on a safe controller when driving in complex environments, or sometimes the *safe* strategy might be too conservative to allow the autonomous vehicle to take any actions. One approach is to not even consider difficult driving scenarios such as unprotected left turns or handling roundabouts. As part of this center, we would like to systematically address this challenge. We plan to develope algorithms that diagnose and understand potential failures of autonomous and intelligent systems in complex environments. Using formal techniques such as specification mining or desired property monitoring, challenging scenarios can be detected. In addition, we will develop minimum violation analyses for safety properties. These will enable us to produce a minimal inconsistent subset of a given specification. The information about this minimal set and the trade-offs between our objectives can help us design potential repairs. Therefore, we plan to study minimal repairs required to fix the potential failures detected and diagnosed in an online setting.