Safety-Assured Design and Adaptation of Connected and Autonomous Vehicles

Xin Chen, Jiameng Fan, Chao Huang, Ruochen Jiao, Wenchao Li, Xiangguo Liu, Yixuan Wang, Zhilu Wang, Weichao Zhou, and Qi Zhu

1 Introduction

Connected and autonomous vehicles (CAVs) have the potential to transform the way we travel. They hold promise for increased mobility, reduced traffic congestion and better fuel efficiency with automated control, as well as the creation of a cooperative network that includes cars, traffic lights and other roadside infrastructures. Autonomous vehicles (AVs) typically employ a wide array of sensors to gather information about the road environment, and then use sophisticated techniques to fuse and process this data to come to a navigation decision in real time in an automated fashion. Many of the underlying components in an AV, such as perception, planning and control make use of deep learning or deep neural networks (DNNs) due to their superior performance. Moreover, greater benefits on safety and fuel economy can be achieved by enabling vehicles to exchange information with one another. In a connected vehicle (CV) system, vehicles are expected to exchange V2X (vehicle-to-everything) messages with surrounding vehicles and roadside units

X. Chen
University of Dayton, Dayton, OH, USA
e-mail: xchen4@udayton.edu

J. Fan · W. Li · W. Zhou
Boston University, Boston, MA, USA
e-mail: jmfan@bu.edu; wenchao@bu.edu; zwc662@bu.edu

C. Huang
University of Liverpool, Liverpool, UK
e-mail: chao.huang2@liverpool.ac.uk

R. Jiao · X. Liu · Y. Wang · Z. Wang · Q. Zhu (✉)
Northwestern University, Evanston, IL, USA
e-mail: ruochen.jiao@northwestern.edu; xiangguoliu2023@northwestern.edu; yixuanwang2024@northwestern.edu; zhilu.wang@northwestern.edu; qzhu@northwestern.edu

V. K. Kukkala, S. Pasricha (eds.), *Machine Learning and Optimization Techniques for Automotive Cyber-Physical Systems*, https://doi.org/10.1007/978-3-031-28016-0_26

(RSUs) for extended perception range, to learn about traffic status down the road, and to coordinate their planning and control decisions. Realizing these potentials of CAVs, however, require tackling the immense challenge of assuring their safety in uncontrolled, public road environments. Numerous recent accidents involving autonomous vehicles are reflective of the safety concerns that loom large in the rapid advancement of CAV technologies [38, 62, 65, 66, 78]. The U.S. Department of Transportation (USDOT) launched the Automated Vehicle Transparency and Engagement for Safe Testing (AV TEST) Initiative in June 2020 to improve the safety in the development and testing of automated driving systems [51]. The USDOT has also started deploying test sites for connected vehicle applications in Florida, New York, and Wyoming [70].

This book chapter will survey recent advances in designing and operating CAVs with safety assurance. Instead of reviewing existing safety standards and industry practices, it aims to bring into focus new methodologies and techniques that have the potential to reshape how we approach the problem of safety assurance of CAVs, paying special attention to two categories of problems—(1) safety verification of CAVs that employ neural network-based components and (2) system adaptation and design with safety guarantees. The chapter will end with a discussion of outstanding technical challenges, broader applications of the surveyed techniques, and the authors' outlook on this important topic of safety assurance of CAVs.

2 Safety Verification of Neural Network-Based Components in CAVs

In CAVs, neural network-based components have been widely used for sensing, perception and prediction, and increasingly being tried for planning and control as well. It is thus critical to conduct safety verification of these neural network-based components for ensuring overall system safety. In particular, this includes conducting robustness of individual neural networks, in particular those used for sensing, perception and prediction, and performing safety verification of a neural network controlled/planned system.

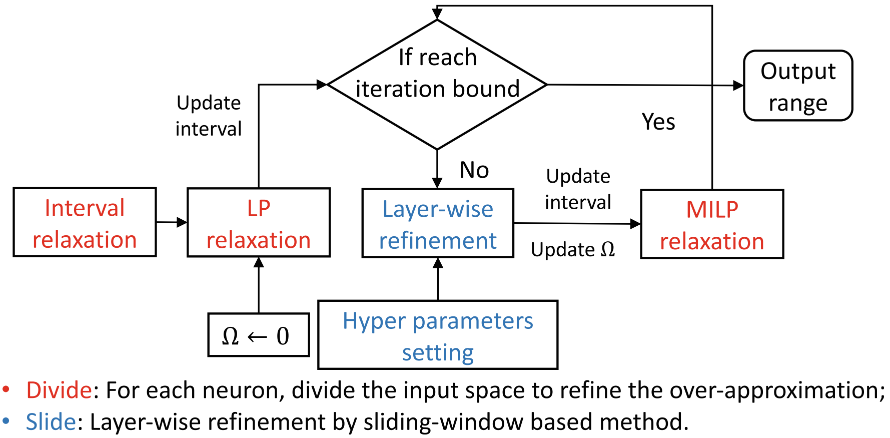
2.1 Robustness Analysis of Deep Neural Networks

Local Robustness Analysis of Neural Networks Robustness is one of the key metrics to measure how stable a neural network's outputs are under random noises, external perturbation, or adversarial attacks to its inputs. Recent studies have in particular highlighted the lack of robustness against adversarial perturbations for neural networks [21, 67]. These adversarial perturbations construct a local input region around each inputs. A neural network is verified to be robust if the neural network outputs are guaranteed to be correct for each local input region, i.e., verification of the **local robustness**.

Measurement of robustness can take the form of upper and lower bounds on certain key input and output parameters. For individual deep learning components, the robustness analysis problem can often be reduced to output range analysis of the neural networks. State-of-the-art methods for output range analysis mainly fall into two categories: constraint programming (CP) [14, 36] and abstract interpretation [63, 71]. CP-based methods can perform exact analysis of the neural networks. However, the scale of deep neural networks limits the usage of these methods because they require encoding an entire network into a large nonlinear programming problem (or an SMT problem) and then solving it. The main drawback with abstract interpretation, on the other hand, is that it is difficult to propagate the dependencies for nonlinear operations across layers [48]. While such methods can scale with the network's size, the performance degrades as the network becomes deeper.

In [29], we propose a layer-wise refinement method, *LayR* to compute a guaranteed and overapproximated range for the output of the neural network for a adversarially perturbed input region. By checking the overapproximated range, we can verify whether the neural network is robust against all possible adversarial perturbations within the input region. *LayR* bridges abstract interpretation with mixed integer linear programming (MILP) and iteratively improves approximation precision by systematically increasing the number of integer variables, as shown in Fig. 1.

Global Robustness Analysis of Neural Networks Most of the efforts in the literature focus on verifying/certifying the local robustness, which characterizes the robustness property for a small region of network input space. However, there are



- Divide: For each neuron, divide the input space to refine the over-approximation;
- Slide: Layer-wise refinement by sliding-window based method.

Fig. 1 Divide-and-slide structure of LayR: Ω defines the number of slack integer variables of all the layers. In the refining process, Ω is monotonically increased to improve the output range estimation, until the iteration bound is reached

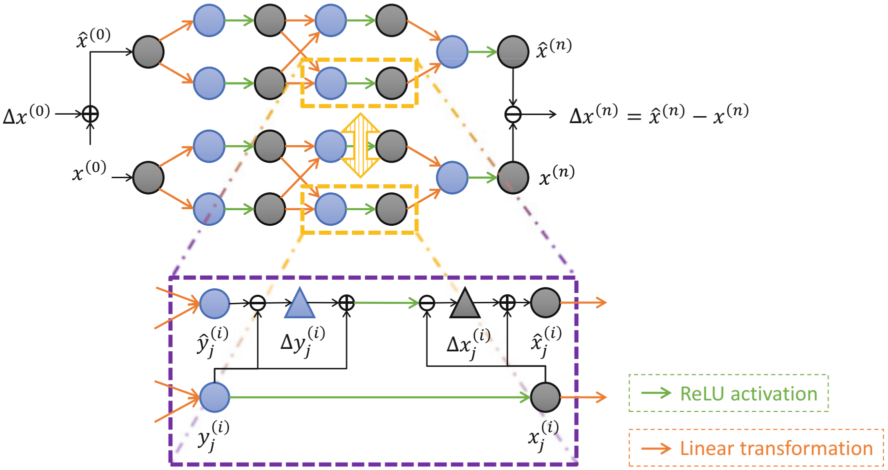


Fig. 2 Interleaving twin-network encoding (ITNE) for neural network global robustness certification. The hidden layer neurons are connected between the two copies of the neural network by distance variables $\Delta y_j^{(i)}$ and $\Delta x_j^{(i)}$

a lot of scenarios that need the robustness property over the entire network input domain, especially for cases that the network input samples cannot be obtained in advance. For instance, for image processing neural networks (like the perception modules in CAVs), the exact input samples during runtime are not always known at design time. In those cases, the global robustness property of the network should be considered, which can bound the worst-case output variation under perturbation for all possible network inputs. Directly conducting local robustness verification for all possible regions in the entire input domain by leveraging the divide-and-conquer techniques is not practical, especially for networks with high-dimension inputs, such as image inputs, as the complexity of divide-and-conquer is exponential to the input dimension. In [77], we developed an efficient global robustness certification algorithm that encoding two copies of the neural network side-by-side, as shown in Fig. 2. One network copy encodes the inference of a normal input while the other one encodes the inference of the disturbed input. Such encoding is formulated as an optimization problem that maximizes the output variation for all possible inputs and perturbations. The differences of hidden neurons between two networks are considered during the relaxation of the optimization problem to efficiently derive a tight over-approximation of the neural network output variation bound. Such over-approximated global robustness can be leveraged to enable the formal verification of the perception neural networks in CAV systems.

2.2 Safety Verification of Neural-Network Controlled Systems

An important class of CAVs can be described by a physical process such as the change of the velocities or distances of vehicles regulated by a learning-enabled controller which can be a neural network. We call such systems neural-network controlled systems. A *Neural-Network Controlled System (NNCS)* is a special sampled-data system which consists of a continuous-time physical process (plant) defined by an ordinary differential equation (ODE) and a feed-forward neural network (FNN) controller which works at discrete time moments. Figure 3 illustrates an execution of an NNCS. The physical process is defined by an ODE $\dot{x} = f(x, u)$ wherein x is the state variable and u is the control input. The FNN controller samples the system state every δ_c time and updates the control input value. Such a system is often safety-critical and it is significant formally verified the safety before implementation.

The safety verification problem asks whether a system can be in an unsafe situation or not. For example, it is crucial to know whether the distance between any of two connected vehicles could be too close at a near future time. Many safety verification problems can be reduced to *reachability problems*, that is, *determining whether the given state can be reached by the system.* Unfortunately, the reachability problem is not decidable even for linear hybrid systems [2, 24]. Hence, most of the existing reachability analysis techniques for hybrid dynamical systems seek to compute an overapproximation of the reachable set. If this overapproximation set does not contain any unsafe state, then the system is safe. Otherwise the safety is unknown, and either the reachable set overapproximation should be refined or an unsafe execution should be found.

NNCSs are particular hybrid dynamical systems such that only the dynamics is updated by the controller, while the system executions are still continuous. Therefore, regardless of the noises or uncertainties between the plant and the controller, an NNCS shows deterministic behavior from an initial state. In other words, a system execution, i.e., the reachable state and control input used at any time, is uniquely determined by the initial state, and we call the function that maps

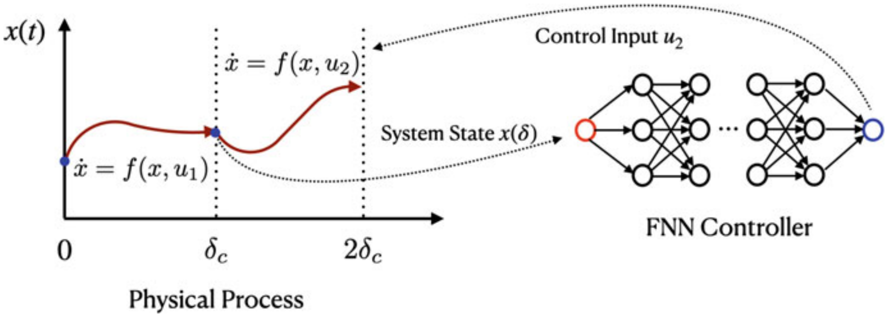


Fig. 3 State evolution of a neural-network controlled system

the initial state to its reachable state at a time *flowmap* which is essentially the solution of the piecewise ODE in Fig. 3. Hence, the reachability analysis task on an NNCS becomes computing the range of the flowmap w.r.t a given set of initial states.

The overapproximate reachability computation for NNCSs is at least as hard as that on general nonlinear sampled-data systems, and the main challenge is to accurately approximate the flowmap function which is a composition of a series of alternative neural network mappings and ODE evolution, and it often does not have a closed-form expression. *Set propagation* [9] is a popular scheme for computing time-bounded reachable sets under such dynamics. From a given initial state set, a set-propagation approach iteratively computes the reachable sets in small and consecutive time intervals the union of which is a cover of the time horizon. The reachable set segment which is also known as *flowpipe* computed in each iteration is propagated to the next time interval. For an NNCS, such an algorithm alternatively computes the flowpipes for the ODE and the output range of the controller until the upper bound of the time horizon is reached. A set-propagation approach for NNCS is often developed in the following two ways.

Pure Range Overapproximation A range overapproximation approach can be directly built by combining a neural network output range analysis method [15, 25, 36, 63, 69, 71, 73] and a reachability computation tool for ODEs [1, 8, 50]. It alternatively computes the reachable sets of the two components and propagates the result to the future time. Such a method mainly focuses on the range overapproximation and often cannot track the state dependency in a flowmap, therefore hard to control the accumulation of overapproximation error on highly nonlinear dynamics.

Functional Overapproximation A functional overapproximation approach seeks to compute an overapproximation for the flowmap function instead of only its range. Most of the existing methods [16, 19, 20, 26, 31–34] in this category uses *Taylor Models (TM)* [47] as the functional overapproximations. Unlike range overapproximations, a functional overapproximation is obtained by composing the functional overapproximations for the sub-components in a system, and it often requires more computational effort than computing a range overapproximation. However, functional overapproximations are able to keep the state dependency in flowmaps and effectively limit the accumulation of overapproximation error in reachability computation. Figure 4 illustrates an functional overapproximation represented by a TM for the output range of an FNN controller at the time $t = k\delta$. The actual flowmap that transforms an initial state x_0 to the control input $u_k = \kappa \circ \Phi(x_0, k\delta)$ used at $t = k\delta$ is overapproximated by a TM $p(x_0) + I$ wherein p is a polynomial and I is an interval remainder.

We briefly introduce the techniques we developed for computing functional overapproximations for the reachable sets of NNCSs.

ReachNN In [26], we present the ReachNN technique to compute reachable set overapproximations for NNCSs. The main contribution is an approach to obtain a TM-like overapproximation for the end-to-end relation of a neural network whose

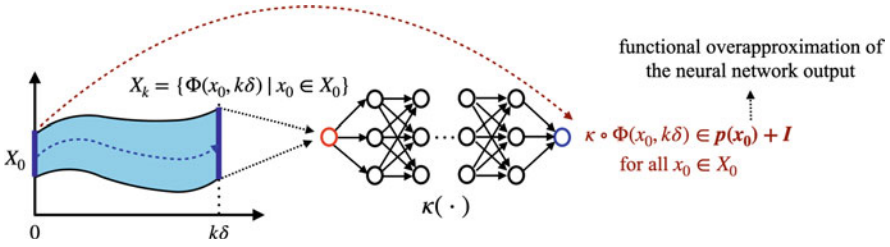

$X_k = \{\Phi(x_0, k\delta) \mid x_0 \in X_0\}$
X_0
0
$k\delta$
$\kappa(\cdot)$
functional overapproximation of the neural network output
$\kappa \circ \Phi(x_0, k\delta) \in p(x_0) + I$
for all $x_0 \in X_0$

Fig. 4 Functional overapproximation of the control input range

activation functions are assumed to be all continuous. By Weierstrass approximation theorem [52] such a neural network over a compact input set can be uniformly approximated as closely as desired by a polynomial. The main method first computes a Bernstein interpolation for the input-output mapping of the neural network, and then a conservative interval remainder for it can be evaluated based on the adaptively selected samples from the input set, and an estimation of the Lipschitz constant of the neural network. We show that this method can be integrated with the reachability tool Flow* [8] which computes TM flowpipes for ODEs, and generate TM reachable sets which approximately keep the state dependency for NNCSs.

ReachNN* ReachNN* [20] leverages GPU-based parallel computing to compute the sampling-based error bound estimation in ReachNN. To further improve the runtime and error bound estimation, ReachNN* also features optional controller re-synthesis via a technique called *verification-aware knowledge distillation* [19] to reduce the Lipschitz constant of the neural network controller. ReachNN* demonstrated 7× to 422× efficiency improvement over ReachNN across a set of benchmarks.

The Polynomial Arithmetic (POLAR) Framework POLAR [31] is introduced for computing TM functional overapproximations for neural network outputs using layer-by-layer propagation. It is an extension of the standard TM arithmetic by introducing (A) Bernstein approximations for the activation functions in neural networks and (B) the symbolic representation of TM remainders in the layer-by-layer propagation framework for computing the output range of a neural network. It can be seamlessly integrated with the reachability tool Flow* to compute TM flowpipes for NNCSs. POLAR has the following main differences from ReachNN: (1) POLAR only uses Bernstein polynomials in approximating activating functions which are always univariate, but ReachNN needs to compute a multivariate Bernstein polynomial when the neural network has multiple inputs. It is much more time costly to compute multivariate Bernstein polynomials than the univariate ones. (2) POLAR uses layer-by-layer propagation framework to compute TM outputs for neural networks, however ReachNN performs an end-to-end overapproximation.

3 System Adaptation and Design with Safety Assurance

For safety-critical systems like CAVs, ensuring safety is a central focus during both the design stage and the runtime operation of them. It is a very challenging task, given the rapid increase of system functional complexity in terms of both scale and features, the usage of advanced architectural components such as multicore CPUs and GPUs, the stringent and contradicting requirements on various objectives such as performance, cost, fault tolerance, and reliability, the adoption of emerging machine learning components, particularly those based on deep neural networks, and the close interaction with a dynamic surrounding environment [60, 86]. In this section below, we will discuss these challenges in CAV design and adaptation, and introduce some of the proposed approaches to them, including those that leverage the methods from Sect. 2 as the underlying safety verification tools.

3.1 Safety-Assured Runtime Adaptation

The dynamic and uncertain environment of CAVs could put changing requirements on their objectives. For instance, a vehicle may need to enhance its planning, navigation and control performance in difficult-to-navigate terrains via more frequent sampling and processing [11–13] (especially for level 5 autonomy), to strengthen its security protection in an adversarial environment by adding monitoring tasks or authentication methods [40, 49], to improve its soft error tolerance in radioactive surroundings through task re-execution [41, 81], or to mitigate the impact under severe communication disturbance by running more computation locally. It is thus critical for those systems to be able to ***adapt*** to the dynamic environment and operation context.

Two major challenges in enabling runtime adaptation are to ensure that during and after the adaptation process, (1) functional safety is guaranteed, and (2) resource and timing constraints are met. To address the first requirement, we may leverage various verification/validation techniques, including those introduced in Sect. 2. To ensure both requirements, however, it is important to develop holistic approaches that span across functional, software, and hardware layers. Next, we will introduce our recent works in this area, along with some of the related works.

Opportunistic Intermittent Control with Safety Guarantees For safety-critical autonomous systems such as robots and automated vehicles, control schemes are often designed conservatively so that system safety can be maintained in a wide variety of situations [10, 43, 56]. During the operation of these systems, however, such schemes can be overly conservative and result in unnecessary resource and/or energy consumption. In [27, 28, 40, 41], we make the observation that certain control steps, even if they are skipped, do not impact either the performance or safety of the overall system. Armed with this observation, we propose an online scheme that opportunistically skips control computation and the corresponding actuation steps by learning specific characteristics of the system's operating environment. We further show that safety could be maintained with this more efficient control scheme.

Specifically, to address the safety, we first compute a *strengthened safe set* based on the notion of *robust control invariant* and *backward reachable set* of the underlying safe controller. Intuitively, the strengthened safety set represents the states at which the system can accept any control input at the current step and be able to stay within safe states, with the underlying safe controller applying input from the next step on. We then develop a monitor to check whether the system is within such strengthened safe set at each control step. Whenever it is found that the system state is out of the strengthened safe set, the monitor will require the system to apply the underlying safe controller for guaranteeing system safety. To efficiently leverage the characteristics of specific operation context and environment, we develop two approaches to leverage the characteristics of operation context and environment when the system is within the strengthened safe set, depending on the type of the underlying safe controller and whether the characteristics are known explicitly. In the simpler case where the safe controller has an analytic expression and the characteristics can be explicitly captured, we use a model-based approach to decide the skipping choices by solving a mixed integer programming (MIP) program. Otherwise, we use a deep reinforcement learning (DRL) approach to learn the mapping from the current state and the historical characteristics to the skipping choices, which implicitly reflects the impact of specific operation context and environment. Our approach is applied to a vehicle adaptive cruise control (ACC) example and shown to provide significant savings in actuation energy and computation load.

Switching Among Multiple Controllers with Safety Guarantees The work in [30] is our first attempt towards the safety adaptation and design for learning-enable systems, allowing a safe, efficient and intelligent switch between different system modes. Motivated by this work, we start considering a more general case, where *switching among multiple existing controllers*, including possibly both model-based ones and neural network-based ones, can be conducted to address system adaptation needs. This is show in Fig. 5. Note that the case where a control step is skipped can be viewed as a special case of switching to a trivial controller.

For safety-critical systems such as CAVs, the key to enable such switching among multiple controllers is to formally ensure safety. In [72], we extend the work from [30] to achieve energy-efficient control adaptation with safety guarantees by switching among multiple controllers (including neural network based ones) via *control invariant set* computation and reinforcement learning. Once a system starts from a control invariant set, it will never leave the set and therefore the safety can be guaranteed. However, it is a hard problem to compute the control invariant set for neural network controlled systems. To solve this problem, we first partition the system space into multiple regions, and on each small local region, we overly approximate the neural network controller by Bernstein polynomials with bounded error. After this transformation, we obtain a hybrid system with polynomial dynamics and compute the invariant set by solving a semi-definite programming

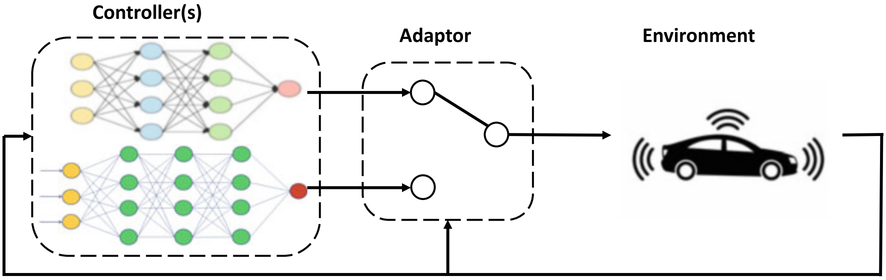


Fig. 5 Adaptation through switching among multiple controllers, which could include both model-based ones and neural network-based ones

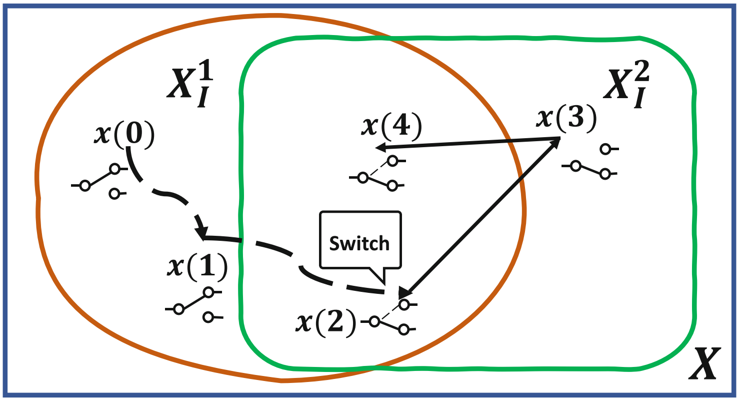


Fig. 6 An example illustrating the energy-efficient switching control with safety guarantees in [72]. We compute the control invariant sets X_I^1 and X_I^2 for controllers κ_1 and κ_2, respectively, and efficiently switch between them based on DRL when the system state is within the intersection of the two invariant sets. For example, in the figure, a control switching happens when the system is at $x(2)$, where both controllers can be safely chosen, and DRL picks κ_2 for energy efficiency

(SDP) problem. The union of all the invariant sets define the safe adaptation space, where we apply deep reinforcement learning (DRL) to learn an energy-efficient strategy. Figure 6 shows an example illustrating our framework. In two case studies, including an ACC example, our framework with invariant set and DRL achieves the best safety-energy consumption efforts when compared to baseline methods.

Cross-Layer Adaptation with Safety-Assured Job Skipping

For many practical systems such as CAVs, the ability to adapt to dynamic requirements is often limited by the tight resource constraints. Moreover, most safety-critical systems employ rigid timing requirements, such as periodic execution and hard deadlines, to guarantee the functionality under worst-case analysis, which further restricts the system adaptation ability. In these cases, it is important to address adaptation with cross-layer approaches.

In the literature, there are a number of methods that adapt task execution with cross-layer consideration. For instance, in [61], the simplex control architecture is proposed, where multiple controllers are being switched at runtime based on the system state and a safety controller keeps the system safe. In [11], an online adaptation approach is proposed for hard real-time systems to temporarily increase control sampling frequency under disturbances while maintaining schedulability. In [5–7], feedback schedulers assign new sampling periods to control tasks during runtime to optimize the control performance under earilest deadline first (EDF) scheduling. In [57], an approach is proposed to adaptively minimize tasks' usage of high quality-of-service resources while meeting control performance requirements.

In [75], different from the previous adaptation approaches that are based on traditional hard timing constraints, we propose an approach that explores proactive task job skippings based on the dynamic system state for state-aware tasks and static *weakly-hard constraints* for other state-unaware tasks. Note that with weakly-hard constraints [4, 55], occasional deadline misses are allowed in a bounded manner. Such paradigm provides more flexibility on the system design than traditional hard real-time constraints, while still allows the possibility of formally guaranteeing functional correctness that soft deadlines cannot provide, using formal analysis techniques such as those in [27, 28].

More specifically, we propose a cross-layer runtime adaptation framework in [75] that allows proactive skipping of task executions and re-allocate resources to the tasks that need performance improvement, as shown in Fig. 7. The system safety is guaranteed under the execution skipping, while the runtime task status is taken into account to maximize the freedom of resource re-allocation. This adaptation framework also involves an efficient runtime scheduler to ensure the timing property during the resource re-allocation. Based on the resource re-allocation, this adaptation framework achieves the dynamic adaptation goals in the best-effort manner. Case study on a robot car example demonstrates the effectiveness of this approach in meeting adaptation needs with safety assurance.

Runtime Safety-Guided Policy Repair For learning-based control systems, runtime safety assurance is particularly crucial and yet challenging. A common approach to providing such kind of assurance is to pair a learning-based controller with a safety controller at runtime. The learning-based controller is usually the primary controller. It learns control policy to attain high performance for the task through data-driven methods. However, it does not provide any safety guarantee especially in scenarios unseen during the training stage. The safety controller is tasked with predicting impending safety violation and taking over control when it deems necessary. It is often designed based on conservative models, has inferior performance compared with its learning-based counterpart, and may require significant computation resources if implemented online.

In order to mitigate the performance loss resulted from the undesirable alternations from the learning-based controller the safety controller while preserving safety, we propose to repair the learning-based controller's control policy by leveraging the interventions carried out by the safety controller in [85]. A naive repair

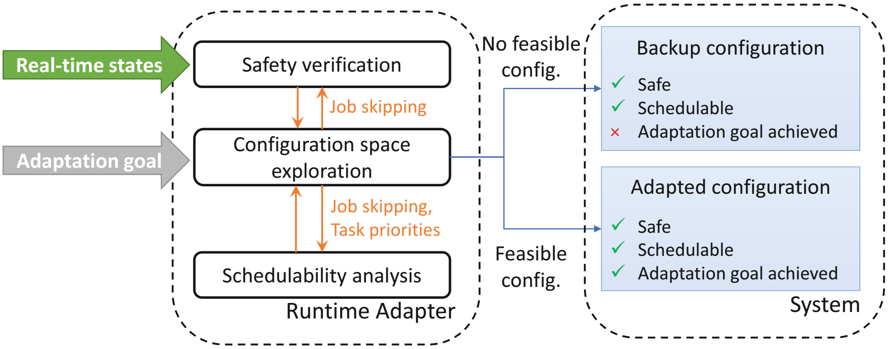


Fig. 7 An overview of the cross-layer runtime adaptation framework proposed in [72]. The system initially runs under a backup configuration that guarantees schedulability and safety. During runtime, an adaptation goal can be given by an external party. The adapter explores the configuration space to search for a feasible solution that achieves the adaptation goal, while ensuring schedulability and safety. If a solution is found, the system will run at this new configuration; otherwise, it will stay at the backup configuration

scheme is to have the learning-based controller learn from the safe control inputs generated by the safety controller until the policy no longer perform unsafe behavior. However, re-training the policy may undermine the policy's performance for the task. To address this, we introduce minimally deviating policy repair via trajectory synthesis. Basically, we synthesize safe trajectories such that by learning from those trajectories, the policy is safe and its parameters are minimally changed, as shown in Fig. 8. This policy repair scheme require naive policy repair as a precondition so that a safe policy is present. Then we formulate an optimization problem where the objective is to perturb the parameters of the safe policy to regress towards those of the original unsafe policy while the constraint is that the perturbation should not result in the policy generating unsafe trajectories. We use local linearization to transform this optimization problem into a trajectory optimization problem. The motivation is that, after applying the optimal perturbation to the policy parameters, the policy should be able to generate the trajectories solved from the trajectory optimization problem. This work can be viewed as data augmentation strategy where the data is optimized specifically for the learning model.

End-to-End Uncertainty-Based Adaptation for Mitigating Adversarial Attacks to CAVs Performing runtime adaptation for CAVs may significantly improve system safety, robustness and security in practice. For instance, in [35], we present an approach for runtime detection and mitigation of adversarial attacks. CAVs have been shown to be susceptible to adversarial attacks, where small perturbations in the input may cause significant errors in the perception results and lead to system failure. For instance, [84] designs a malicious billboard to attack end-to-end deep learning-based driving models. [59] generates a dirty road patch with carefully-designed adversarial patterns, which can appear as normal dirty patterns for human drivers while leading to significant perception errors and causing vehicles to deviate

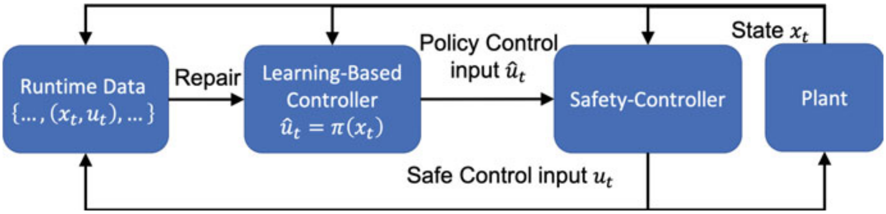


Fig. 8 The combination of a learning-based controller and a safety controller provides runtime safety assurance. Given the state x_t, the safety controller filters the control input $\hat{u}_t$ generated by the learning-based control policy π, and produces a safe control input u_t to the plant. The data (x_t, u_t) is collected at runtime to repair the policy π

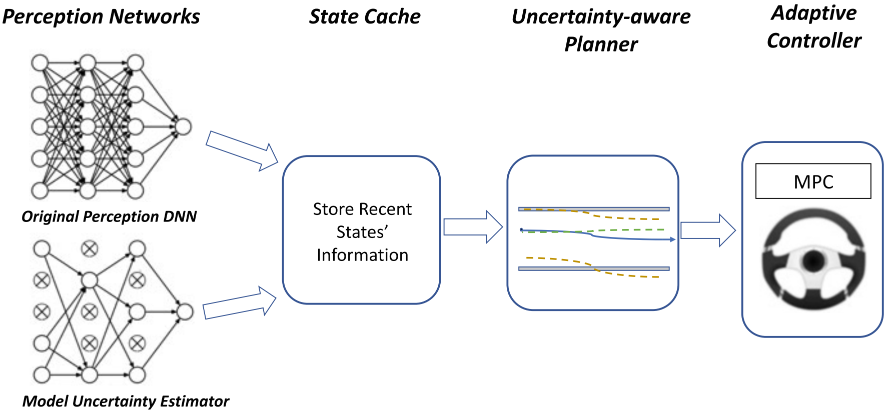

Perception Networks
State Cache
Uncertainty-aware Planner
Adaptive Controller
Original Perception DNN
Model Uncertainty Estimator
Store Recent States' Information
MPC

Fig. 9 An end-to-end detection and mitigation framework for adversarial attacks to CAVs [35]. In the perception module, the original neural network is to predict lane lines with confidence value and the data uncertainty while the other neural network is used to estimate the model uncertainty by Monte-Carlo dropout. The state cache will store recent predictions and then the planner will select one based on confidence values. The planner will calculate the center line in a safe region by considering both uncertainties and lane predictions. Finally, the controller will optimize the low-level control by an uncertainty-aware MPC

from their lanes within as short as 1 s. On the defense side, most previous works focus on detecting anomaly in the input data [39, 44] or making the perception neural networks themselves more robust against input perturbation [46].

In [35], instead of addressing adversarial attacks only on perception module, we develop *an uncertainty-based end-to-end approach* that detects and mitigates adversarial attacks throughout perception, planning, and control modules. In particular, we measure the confidence and uncertainty of perception modules, and conduct robust adaptation in the following modules accordingly based on the uncertainty analysis, as shown in Fig. 9. We apply the framework to the commercial automated lane centering system in OpenPilot and demonstrate that the impact of attacks can be reduced by up to 90%.

3.2 Safety-Driven Learning and System Design

Besides runtime adaptation, another critical aspect for CAV safety is to **design and learn** neural network-based components that can ensure system safety (i.e., not entering unsafe states) and robustness (i.e., being safe under disturbances from random noises or malicious attacks). Next, we will first introduce works that improve the robustness of neural networks, and then introduce techniques that try to learn safe neural network-based controllers from multiple experts, with verification in the loop, and based on physical information, respectively.

Learning Provably Robust Neural Networks Most of the current verification techniques for learning-enabled systems focus on analyzing trained systems, e.g., whether a trained neural network satisfies some specification. It is more desirable to have these systems "correct-by-construction". In fact, the same power of modern compute and data that has been fueling data-driven learning can be leveraged to scale up verification and enable provably-correct training of neural networks. We give such an example below.

For adversarial robustness problems in neural networks [3, 23, 45, 79, 83], given a model f_θ, loss function $\mathcal{L}$, and training data distribution $\mathcal{X}$, the training algorithm aims to minimize the loss whereas the adversary aims to maximize the loss within a neighborhood $\mathbb{S}(\boldsymbol{x}, \epsilon)$ of each input data $\boldsymbol{x}$ as follows:

$$\min_{\theta} E_{(\boldsymbol{x},y)\in\mathcal{X}} \left[\max_{\boldsymbol{x}'\in\mathbb{S}(\boldsymbol{x},\epsilon)} \mathcal{L}(f_\theta(\boldsymbol{x}'), y) \right] \tag{1}$$

In general, the inner maximization is intractable. Most existing techniques focus on finding an approximate solution. There are two main approaches to approximate the inner loss (henceforth referred to as *robust loss*). One direction is to generate adversarial examples to compute a lower bound of robust loss. The other is to compute an upper bound of robust loss by over-approximating the model outputs.

Verification techniques [17, 36, 53, 54, 58] for neural networks can be used to compute a certified upper bound of *robust loss* (henceforth referred to as *abstract loss*). Given a neural network, a simple way to obtain this upper bound is to propagate value bounds across the network, also known as interval bound propagation (IBP) [23, 48]. Techniques such as CROWN [82], DeepZ [63], MIP [68] and RefineZono [64], can compute more precise bounds, but also incur much higher computational costs. Building upon these upper bound verification techniques, approaches such as DIFFAI [48] construct a differentiable *abstract loss* corresponding to the upper bound estimation and incorporate this loss function during training. However, [23] and [83] observe that a tighter approximation of the upper bound does not necessarily lead to a network with low robust loss. They show that IBP-based methods can produce networks with state-of-the-art certified robustness. More recently, COLT [3] proposed to combine adversarial training and zonotope propagation. Zonotopes are a collection of affine forms of the input variables and intermediate vector outputs in the neural network. The idea is to train the network with the so-called latent adversarial examples which are adversarial examples that lie inside these zonotopes. *AdvIBP* [18] proposed a principled framework for combining adversarial loss and abstract loss. Fan and Li [18] argues that minimizing *adversarial loss* and minimizing *abstract loss* can be viewed as bounding the true *robust loss* from two ends. From an optimization perspective, this amounts to an optimization problem with two objectives and can be solved using gradient descent methods if both objectives are semi-smooth. Inspired by the work on moment estimates [37], *AdvIBP* proposed a novel joint training scheme to compute the weights adaptively and minimize the joint objective with unbiased gradient estimates. For efficient training, *AdvIBP* uses FGSM and random initialization for computing the adversarial loss and IBP for computing the abstract loss. We summarize and compare the key features in Table 1.

Learning Neural Network Controllers from Multiple Experts In Sect. 3.1, we present an approach for switching among multiple controllers, including both model-based and neural network-based, with safety assurance [72]. After observing the benefit of such switching control, we then further propose a framework to automatically learn a better neural network-based controller from those multiple existing ones, by learning a system-level ensemble strategy and robust distillation via adversarial examples [74], as shown in Fig. 10. Specifically, we ensemble the multiple controllers by learning a linear combination weight for each expert through reinforcement learning optimization to enhance the control safety and efficiency. To achieve better verifiability based on the observation that smaller Lipschitz constant of the neural network leads to stronger robustness, we conduct teacher-student knowledge distillation with a novel probabilistic adversarial training to obtain the final controller. The final learned controller shows better control robustness when facing measurement noise and adversarial attacks, higher control energy efficiency, and better verifiability in terms of reachable set and invariant set computation.

Verification-in-the-Loop Control Learning with Safety Guarantees Traditionally, control synthesis/learning for a safety-critical system often follows the *design-then-verify* open-loop process, which could result in many iterations between design and verification, and may still fail to provide any safety guarantees. In [76], we instead propose a closed-loop process for control learning by integrating the verification results into the design module via propagating the feedback as an approximated gradient, i.e., a *design-while-verify* process. In particular, the verification results refer to the computed reachable set in this work. We establish two distance metrics, including the geometric distance and the Wasserstein distance, to measure how far the computed reachable set of the current controller is from the goal region and the unsafe region. We then add perturbations to the controller and approximate the gradient for it by a difference method for update until the final reach-avoid property is met.

Physics-Aware Safety-Assured Design of Hierarchical Neural Network Planner for CAVs In designing CAVs in practice, it is critical to consider the safety

Table 1 Comparison of different methods for training robust neural networks. We highlight the loss function used in each method. If there is an *abstract loss* used in training or post-training verification, we also list the corresponding verification method. We categorize the methods along five dimensions, with ✓ indicating a desirable property or an explicit consideration

Method	L-2ptoss	*Abstract loss*	Efficiency[a]	Empirical robustness	Provable robustness	No weight[b] tuning/ scheduling
Baseline	Regular loss	n/a	✓			n/a
FGSM [22]	Adversarial loss	n/a	✓	✓		n/a
FGSM+random init [80]	Adversarial loss	n/a	✓	✓		n/a
PGD [45]	Adversarial loss	n/a		✓		n/a
COLT [3]	Latent adversarial loss	RefineZono[c]		✓	✓	n/a
DIFFAI [48]	Abstract loss[d]	DeepZ	✗[e]		✓	n/a
CROWN-IBP [83]	Regular loss + abstract loss	CROWN + IBP	✓		✓	
IBP method [23]	Regular loss + abstract loss	IBP	✓		✓	
***AdvIBP* [18]**	***Adversarial loss + abstract loss***	**IBP**	✓	✓	✓	✓

The bold text is the best performing method for training robust neural networks

[a] The efficiency baseline is the training time for each epoch during regular training. ✓ represents the training time is comparable to the baseline

[b] The weights here represent the weights for the different losses if there are multiple of them

[c] RefineZono is not used to construct an abstract loss. Instead, it is used to generate latent adversarial examples and for post-training verification

[d] In their experiments, DIFFAI shows that adding regular loss with a fixed weight can achieve better performance

[e] DIFFAI can also use IBP for training and verification for improved efficiency. However, the best robustness results are achieved using DeepZ

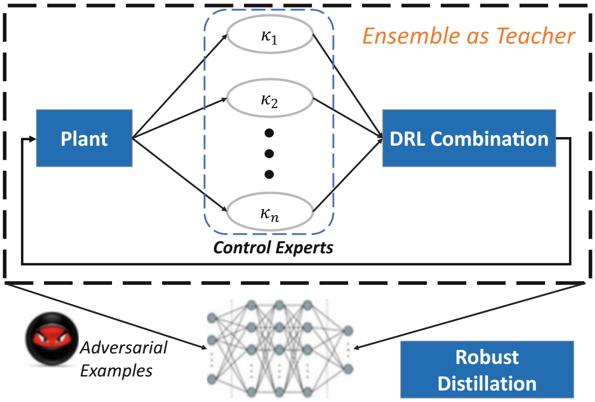


Fig. 10 Overview of the Cocktail framework to learn a better neural network controller from multiple existing control experts via system-level ensemble from reinforcement learning and robust distillation with probabilistic adversarial training

of the learning-based components. For instance, many recent neural network-based planners demonstrate significant performance improvement and accident rate reduction in average over traditional model-based methods. Some of those learn a single neural network for planning via reinforcement learning, imitation learning, supervised learning, etc., while others employ a hierarchical planner design, which usually consists of low-level planners for different modes and a high-level planner that is responsible for selecting the mode. However, even though safety improvement is often considered and demonstrated empirically through experiments in those works, formal system safety verification remains a challenging problem.

In [42], we propose a hierarchical neural network based planner that analyzes the underlying physical scenarios of the system and learns a system-level behavior planning scheme with multiple scenario-specific motion-planning strategies, as shown in Fig. 11. We develop an efficient verification method that incorporates overapproximation of the system state reachable set and novel partition and union techniques for formally ensuring system safety under our physics-aware planner. With theoretical analysis, we show that considering the different physical scenarios and building a hierarchical planner based on such analysis may improve system safety and verifiability. We also empirically demonstrate the effectiveness of our approach and its advantage over other baselines in practical case studies of unprotected left turn and highway merging, two common challenging safety-critical tasks in autonomous driving.

4 Conclusion and Future Directions

Safety is a critical challenge to the widespread adoption of CAVs. In this book chapter, we have outlined some specific technical problems and proposed solutions for verifying and improving the safety of CAVs, especially aiming at those challenges brought by the increasing usage of learning-based components. The road

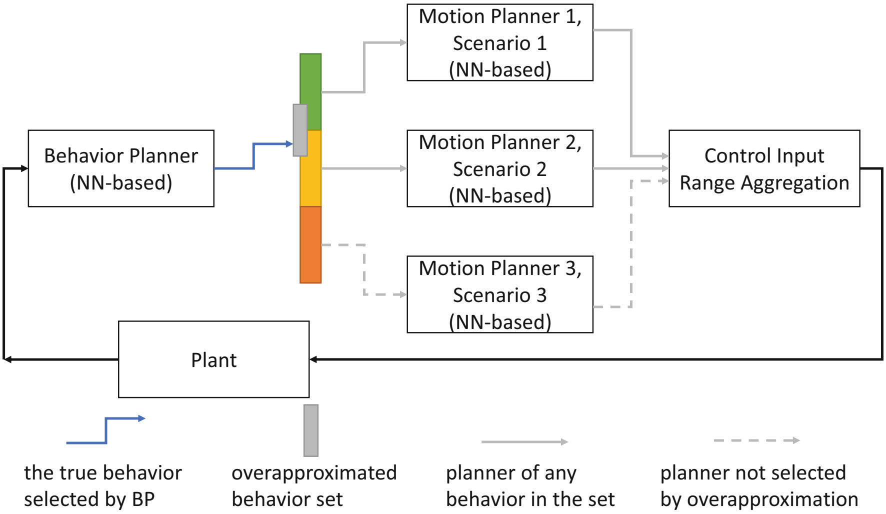


Fig. 11 Design of a hierarchical neural network-based planner that consists of one behavior planner μ and N motion planners $\{\kappa_1, \kappa_2, \ldots, \kappa_N\}$ [42]. In the figure, we have $N = 3$ for example. The behavior planner decides the most appropriate behavior given the system state x, and then the corresponding motion planner is enabled to control the system. To compute an overapproximation of the reachable set of the system under such hierarchical planner, we first compute an overapproximated behavior set, which is illustrated by the grey rectangle in the figure. Then for each behavior in the overapproximated behavior set, the corresponding motion planner's output range can be aggregated as the possible control input range, thus computing an overapproximation of the system state reachable set under all possible behaviors

to safe autonomy, however, still requires clearing major roadblocks in perception, control, and connectivity, and we discuss some of those below.

On the verification side, developing more efficient and rigorous techniques especially for CAVs with neural network-based perception modules will be a primary focus. The high dimensionality of the problem may necessitate sacrificing deterministic guarantees and adopting statistical or probabilistic analysis. In particular, for probabilistic safety verification of neural network-controlled systems, existing statistic model checking approach often requires a large number of system simulations and costs a lot of time. This may be relieved by approximately tracking the propagation of the probabilistic distributions of reachable states. Another possible direction is to perform property-directed reachability analysis for neural network-controlled systems. Existing reachability algorithms explore all state space that is possible to reach, and it is often unnecessary to do so when a safety property is simply defined by very few constraints. A property-directed reachability technique may exclude the state space that is not relevant to the safety condition and reduce a great amount of time in computing the reachable sets. For connected vehicles, abstract modeling of inter-vehicle information exchange and interactions and compositional analysis will be the key to leapfrogging the complexity challenge of verifying the safety of large-scale multi-agent systems.

On design and adaptation of CAVs, we believe that the key is to develop more end-to-end approaches that can address CAV safety across sensing, perception, planning and control stages, and more cross-layer approaches that can consider functional safety, software and hardware execution correctness, and even inter-vehicle communication reliability in a holistic manner. For instance, effectively addressing adversarial attacks to neural network-based perception modules will require quantitative analysis of their impact on downstream planning/control modules and ultimately on system-level safety, and will need end-to-end mitigation strategies that are developed based on such analysis. Runtime adaptation to mitigate component failures will need techniques to assess the impact of those failures across system layers, explore adaptation solutions that address the bottlenecks, and ensure the changed configurations meet various constraints across functional, software and hardware layers.

Acknowledgments We gratefully acknowledge the support from the US National Science Foundation (NSF) awards CCF-1646497, CCF-1834324, CNS-1834701, CNS-1839511, IIS-1724341, CNS-2038853, the US Office of Naval Research (ONR) grant N00014-19-1-2496, and the US Air Force Research Laboratory (AFRL) under contract number FA8650-16-C-2642.

References

1. Althoff, M.: An introduction to cora 2015. In: Proceedings of ARCH'15. EPiC Series in Computer Science, vol. 34, pp. 120–151. EasyChair (2015)
2. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P.-H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: The algorithmic analysis of hybrid systems. Theor. Comput. Sci. **138**(1), 3–34 (1995)
3. Balunovic, M., Vechev, M.: Adversarial training and provable defenses: Bridging the gap. In: International Conference on Learning Representations (2020)
4. Bernat, G., Cayssials, R.: Guaranteed on-line weakly-hard real-time systems. In: IEEE Real-Time Systems Symposium (RTSS) (2001)
5. Castane, R., Marti, P., Velasco, M., Cervin, A., Henriksson D.: Resource management for control tasks based on the transient dynamics of closed-loop systems. In: 18th Euromicro Conference on Real-Time Systems (ECRTS'06) (2006)
6. Cervin, A., Eker, J., Bernhardsson, B., Årzén, K.E.: Feedback–feedforward scheduling of control tasks. Real-Time Syst. **23**(1), 25–53 (2002)
7. Cervin, A., Velasco, M., Marti, P., Camacho, A.: Optimal online sampling period assignment: theory and experiments. IEEE Trans. Control Syst. Technol. **19**(4), 902–910 (2011)
8. Chen, X., Ábrahám, E., Sankaranarayanan, S.: Flow*: an analyzer for non-linear hybrid systems. In: Proceedings of CAV'13. LNCS, vol. 8044, pp. 258–263. Springer (2013)
9. Chen, X., Sankaranarayanan, S.: Reachability analysis for cyber-physical systems: are we there yet? In: Proceedings of NFM'22. LNCS, vol. 13260, pp. 109–130. Springer (2022)
10. Chisci, L., Rossiter, J.A., Zappa, G.: Systems with persistent disturbances: predictive control with restricted constraints. Automatica **37**(7) (2001)
11. Dai, X., Chang, W., Zhao, S., Burns, A.: A dual-mode strategy for performance-maximisation and resource-efficient cps design. ACM Trans. Embed. Comput. Syst. **18**(5s) (2019)

12. Davare, A., Zhu, Q., Di Natale, M., Pinello, C., Kanajan, S., Sangiovanni-Vincentelli, A.: Period optimization for hard real-time distributed automotive systems. In: Design Automation Conference (DAC'07) (2007)
13. Deng, P., Zhu, Q., Davare, A., Mourikis, A., Liu, X., Natale, M.D.: An efficient control-driven period optimization algorithm for distributed real-time systems. IEEE Trans. Comput. **65**(12), 3552–3566 (2016)
14. Dutta, S., Jha, S., Sankaranarayanan, S., Tiwari, A.: Output range analysis for deep feedforward neural networks. In: NASA Formal Methods Symposium, pp. 121–138. Springer (2018)
15. Dutta, S., Jha, S., Sankaranarayanan, S., Tiwari, A.: Output range analysis for deep feedforward neural networks. In: Proceedings of NFM'18. LNCS, vol. 10811, pp. 121–138. Springer (2018)
16. Dutta, S., Chen, X., Sankaranarayanan, S.: Reachability analysis for neural feedback systems using regressive polynomial rule inference. In: 22nd ACM International Conference on Hybrid Systems: Computation and Control (HSCC), pp. 157–168 (2019)
17. Dvijotham, K., Stanforth, R., Gowal, S., Mann, T.A., Kohli, P.: A dual approach to scalable verification of deep networks. In: UAI, vol. 1, p. 2 (2018)
18. Fan, J., Li, W.: Adversarial training and provable robustness: a tale of two objectives. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 7367–7376 (2021)
19. Fan, J., Huang, C., Li, W., Chen, X., Zhu, Q.: Towards verification-aware knowledge distillation for neural-network controlled systems. In: 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 1–8. IEEE (2019)
20. Fan, J., Huang, C., Chen, X., Li, W., Zhu, Q.: Reachnn*: a tool for reachability analysis of neural-network controlled systems. In: International Symposium on Automated Technology for Verification and Analysis (2020)
21. Fawzi, A., Moosavi-Dezfooli, S.-M., Frossard, P.: The robustness of deep networks: a geometrical perspective. IEEE Signal Process. Mag. **34**(6), 50–62 (2017)
22. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conferences on Learning Representations (2015)
23. Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T., Kohli, P.: On the effectiveness of interval bound propagation for training verifiably robust models. Preprint (2018). arXiv:1810.12715
24. Henzinger, T.A., Kopke, P.W., Puri, A., Varaiya, P.: What's decidable about hybrid automata? In: Proceedings of the 27th Annual ACM Symposium on Theory of Computing (STOC'95), pp. 373–382. ACM (1995)
25. Huang, X., Kwiatkowska, M., Wang, S., Wu, M.: Safety verification of deep neural networks. In: International Conference on Computer Aided Verification, pp. 3–29. Springer (2017)
26. Huang, C., Fan, J., Li, W., Chen, X., Zhu, Q.: Reachnn: reachability analysis of neural-network controlled systems. ACM Trans. Embedd. Comput. Syst. **18**(5s), 1–22 (2019)
27. Huang, C., Li, W., Zhu, Q.: Formal verification of weakly-hard systems. In: The 22nd ACM International Conference on Hybrid Systems: Computation and Control (HSCC) (2019)
28. Huang, C., Chang, K.-C., Lin, C.-W., Zhu, Q.: Saw: a tool for safety analysis of weakly-hard systems. In: 32nd International Conference on Computer-Aided Verification (CAV'20) (2020)
29. Huang, C., Fan, J., Chen, X., Li, W., Zhu, Q.: Divide and slide: layer-wise refinement for output range analysis of deep neural networks. In: International Conference on Embedded Software (EMSOFT) (2020)
30. Huang, C., Xu, S., Wang, Z., Lan, S., Li, W., Zhu, Q.: Opportunistic intermittent control with safety guarantees for autonomous systems. Proccedings of the Design Automation Conference (DAC'20) (2020)
31. Huang, C., Fan, J., Chen, X., Li, W., Zhu, Q.: Polar: a polynomial arithmetic framework for verifying neural-network controlled systems. Preprint (2021). arXiv:2106.13867
32. Ivanov, R., Weimer, J., Alur, R., Pappas, G.J., Lee, I.: Verisig: verifying safety properties of hybrid systems with neural network controllers. In: 22nd ACM International Conference on Hybrid Systems: Computation and Control (HSCC), pp. 169–178 (2019)
33. Ivanov, R., Carpenter, T.J., Weimer, J., Alur, R., Pappas, G.J., Lee, I.: Verifying the safety of autonomous systems with neural network controllers. ACM Trans. Embedd. Comput. Syst. (TECS) **20**(1), 1–26 (2020)

34. Ivanov, R., Carpenter, T., Weimer, J., Alur, R., Pappas, G., Lee, I.: Verisig 2.0: verification of neural network controllers using taylor model preconditioning. In: Silva, A., Rustan, K., Leino, M. (eds.) Computer Aided Verification, pp. 249–262. Springer International Publishing, Cham (2021)
35. Jiao, R., Liang, H., Sato, T., Shen, J., Chen, Q.A., Zhu, Q.: End-to-end uncertainty-based mitigation of adversarial attacks to automated lane centering. In: 2021 IEEE Intelligent Vehicles Symposium (IV), pp. 266–273 (2021)
36. Katz, G., Barrett, C., Dill, D.L., Julian, K., Kochenderfer, M.J.: Reluplex: an efficient smt solver for verifying deep neural networks. In: International Conference on Computer Aided Verification (CAV), pp. 97–117. Springer (2017)
37. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations (2016)
38. Lee, D., Hess, D.J.: Public concerns and connected and automated vehicles: safety, privacy, and data security. Hum. Soc. Sci. Commun. **9**(1), 1–13 (2022)
39. Lee, K., Lee, K., Lee, H., Shin, J.: A simple unified framework for detecting out-of-distribution samples and adversarial attacks. Adv. Neural Inf. Process. Syst. **31** (2018)
40. Liang, H., Wang, Z., Roy, D., Dey, S., Chakraborty, S., Zhu, Q.: Security-driven codesign with weakly-hard constraints for real-time embedded systems. In: 37th IEEE International Conference on Computer Design (ICCD'19) (2019)
41. Liang, H., Wang, Z., Jiao, R., Zhu, Q.: Leveraging weakly-hard constraints for improving system fault tolerance with functional and timing guarantees. In: 2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD), pp. 1–9 (2020)
42. Liu, X., Huang, C., Wang, Y., Zheng, B., Zhu, Q.: Physics-aware safety-assured design of hierarchical neural network based planner. In: 2022 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS) (2022)
43. Löfberg, J: Minimax Approaches to Robust Model Predictive Control, vol. 812. University Electronic Press, Linköping (2003)
44. Lu, J., Issaranon, T., Forsyth, D.: Safetynet: detecting and rejecting adversarial examples robustly. In: Proceedings of the IEEE international conference on computer vision, pp. 446–454 (2017)
45. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. Preprint (2017). arXiv:1706.06083
46. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (2018)
47. Makino, K., Berz, M.: Taylor models and other validated functional inclusion methods. J. Pure Appl. Math. **4**(4), 379–456 (2003)
48. Mirman, M., Gehr, T., Vechev, M.: Differentiable abstract interpretation for provably robust neural networks. In: International Conference on Machine Learning, pp. 3578–3586 (2018)
49. Mundhenk, P., Paverd, A., Mrowca, A., Steinhorst, S., Lukasiewycz, M., Fahmy, S.A., Chakraborty, S.: Security in automotive networks: lightweight authentication and authorization. ACM Trans. Des. Autom. Electron. Syst. **22**(2), 25:1–25:27 (2017)
50. Nedialkov, N.S.: Implementing a rigorous ode solver through literate programming. In: Rauh, A., Auer, E. (eds.) Modeling, Design, and Simulation of Systems with Uncertainties. Mathematical Engineering, vol. 3, pp. 3–19. Springer, Berlin/Heidelberg (2011)
51. NHTSA Media.: U.S. transportation secretary elaine l. chao announces first participants in new automated vehicle initiative web pilot to improve safety, testing, public engagement. NHTSA (2020)
52. Phillips, G.M.: Interpolation and Approximation by Polynomials. Springer, Berlin (2003)
53. Prabhakar, P., Afzal, Z.R.: Abstraction based output range analysis for neural networks. In: Advances in Neural Information Processing Systems, pp. 15788–15798 (2019)
54. Raghunathan, A., Steinhardt, J., Liang, P.S.: Semidefinite relaxations for certifying robustness to adversarial examples. In: Advances in Neural Information Processing Systems, pp. 10877–10887 (2018)

55. Ramanathan, P.: Overload management in real-time control applications using (m, k)-firm guarantee. IEEE Trans. Parallel Distrib. Syst. **10**(6), 549–559 (1999)
56. Richards, A.G.: Robust constrained model predictive control. Ph.D Thesis, Massachusetts Institute of Technology, 2005
57. Roy, D., Chang, W., Mitter, S.K., Chakraborty, S.: Tighter dimensioning of heterogeneous multi-resource autonomous cps with control performance guarantees. In: ACM/IEEE Design Automation Conference (DAC), pp. 1–6 (2019)
58. Ruan, W., Huang, X., Kwiatkowska, M.: Reachability analysis of deep neural networks with provable guarantees. In: International Joint Conferences on Artificial Intelligence (2018)
59. Sato, T., Shen, J., Wang, N., Jia, Y., Lin, X., Chen, Q.A.: Dirty road can attack: Security of deep learning based automated lane centering under {Physical-World} attack. In: 30th USENIX Security Symposium (USENIX Security 21), pp. 3309–3326 (2021)
60. Seshia, S.A., Hu, S., Li, W., Zhu, Q.: Design automation of cyber-physical systems: challenges, advances, and opportunities. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **36**(9), 1421–1434 (2017)
61. Seto, D., Krogh, B., Sha, L., Chutinan, A.: The simplex architecture for safe online control system upgrades. In: American Control Conference (ACC), vol. 6, pp. 3504–3508 (1998)
62. Siddiqui, F., Lerman, R., Merrill, J.B.: Teslas running autopilot involved in 273 crashes reported since last year. The Washington Post (2022)
63. Singh, G., Gehr, T., Mirman, M., Püschel, M., Vechev, M.: Fast and effective robustness certification. In: Advances in Neural Information Processing Systems, pp. 10802–10813 (2018)
64. Singh, G., Gehr, T., Püschel, M., Vechev, M.: Boosting robustness certification of neural networks. In: International Conference on Learning Representations (2019)
65. Summary Report: Standing general order on crash reporting for automated driving systems. Technical Report, NHTSA, 2022
66. Summary Report: Standing general order on crash reporting for level 2 advanced driver assistance systems. Technical Report, NHTSA, 2022
67. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. International Conferences on Learning Representations (2014)
68. Tjeng, V., Xiao, K.Y., Tedrake, R.: Evaluating robustness of neural networks with mixed integer programming. In: International Conference on Learning Representations (2019)
69. Tran, H.-D., Bak, S., Xiang, W., Johnson, T.T.: Verification of deep convolutional neural networks using imagestars. In: International Conference on Computer-Aided Verification (2020)
70. U.S. Department of Transportation: Using connected vehicle technologies to solve real-world operational problems. USDOT ITS Research - Connected Vehicle Pilot Deployment Program (2022)
71. Wang, S., Pei, K., Whitehouse, J., Yang, J., Jana, S.: Formal security analysis of neural networks using symbolic intervals. In: 27th {USENIX} Security Symposium ({USENIX} Security 18), pp. 1599–1614 (2018)
72. Wang, Y., Huang, C., Zhu, Q.: Energy-efficient control adaptation with safety guarantees for learning-enabled cyber-physical systems. In: Proceedings of the 39th International Conference on Computer-Aided Design, ICCAD '20, New York, NY, USA. Association for Computing Machinery (2020)
73. Wang, S., Zhang, H., Xu, K., Lin, X., Jana, S., Hsieh, C.-J., Kolter, J.Z.: Beta-crown: efficient bound propagation with per-neuron split constraints for neural network robustness verification. In: Proceedings of NeurIPS'21, vol. 34 (2021)
74. Wang, Y., Huang, C., Wang, Z., Xu, S., Wang, Z., Zhu, Q.: Cocktail: learn a better neural network controller from multiple experts via adaptive mixing and robust distillation. In: 2021 58th ACM/IEEE Design Automation Conference (DAC), pp. 397–402. IEEE (2021)
75. Wang, Z., Huang, C., Kim, H., Li, W., Zhu, Q.: Cross-layer adaptation with safety-assured proactive task job skipping. ACM Trans. Embed. Comput. Syst. **20**(5s) (2021)

76. Wang, Y., Huang, C., Wang, Z., Wang, Z., Zhu, Q.: Design-while-verify: correct-by-construction control learning with verification in the loop. In: 59th ACM/IEEE Design Automation Conference, DAC 2022, San Francisco, CA, USA, July 10–14 (2022)
77. Wang, Z., Huang, C., Zhu, Q.: Efficient global robustness certification of neural networks via interleaving twin-network encoding. In: DATE'22: Proceedings of the Conference on Design, Automation and Test in Europe (2022)
78. Wiggers, K.: Waymo's driverless cars were involved in 18 accidents over 20 months. VentureBeat (2020)
79. Wong, E., Kolter, Z.: Provable defenses against adversarial examples via the convex outer adversarial polytope. In: International Conference on Machine Learning, pp. 5286–5295 (2018)
80. Wong, E., Rice, L., Kolter, J.Z.: Fast is better than free: revisiting adversarial training. In: International Conferences on Learning Representations (2020)
81. Zheng, B., Gao, Y., Zhu, Q., Gupta, S.: Analysis and optimization of soft error tolerance strategies for real-time systems. In: 2015 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), pp. 55–64 (2015)
82. Zhang, H., Weng, T.-W., Chen, P.-Y., Hsieh, C.-J., Daniel, L.: Efficient neural network robustness certification with general activation functions. In: Advances in Neural Information Processing Systems, pp. 4939–4948 (2018)
83. Zhang, H., Chen, H., Xiao, C., Li, B., Boning, D., Hsieh, C.-J.: Towards stable and efficient training of verifiably robust neural networks. In: International Conference on Learning Representations (2020)
84. Zhou, H., Li, W., Kong, Z., Guo, J., Zhang, Y., Yu, B., Zhang, L., Liu, C.: Deepbillboard: Systematic physical-world testing of autonomous driving systems. In: 2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE), pp. 347–358. IEEE (2020)
85. Zhou, W., Gao, R., Kim, B., Kang, E., Li, W.: Runtime-safety-guided policy repair. In: Deshmukh, J., Ničković, D. (eds.) Runtime Verification, pp. 131–150. Springer International Publishing, Cham (2020)
86. Zhu, Q., Sangiovanni-Vincentelli, A.: Codesign methodologies and tools for cyber–physical systems. In: Proceedings of the IEEE **106**(9), 1484–1500 (2018)