State-Dependent Conformal Perception Bounds for Neuro-Symbolic Verification of Autonomous Systems

Thomas Waite WAITET@RPI.EDU
Rensselaer Polytechnic Institute, Troy, New York
Yuang Geng YUANG.GENG@UFL.EDU
Trevor Turnquist TREVOR.TURNQUIST@UFL.EDU
Ivan Ruchkin* IRUCHKIN@ECE.UFL.EDU
University of Florida, Gainesville, Florida
Radoslav Ivanov* IVANOR@RPI.EDU
Rensselaer Polytechnic Institute, Troy, New York

Editors: G. Pappas, P. Ravikumar, S. A. Seshia

Abstract

It remains a challenge to provide safety guarantees for autonomous systems with neural perception and control. A typical approach obtains symbolic bounds on perception error (e.g., using conformal prediction) and performs verification under these bounds. However, these bounds can lead to drastic conservatism in the resulting end-to-end safety guarantee. This paper proposes an approach to synthesize symbolic perception error bounds that serve as an optimal interface between perception performance and control verification. The key idea is to consider our error bounds to be heteroskedastic with respect to the system's state — not time like in previous approaches. These bounds can be obtained with two gradient-free optimization algorithms. We demonstrate that our bounds lead to tighter safety guarantees than the state-of-the-art in a case study on a mountain car.
Keywords: Neural network verification, conformal prediction, gradient-free optimization

1. Introduction

Modern autonomous systems such as Waymo's self-driving cars and VoloCity's air taxis show impressive capabilities of neural perception and control, but providing safety guarantees on such systems remains difficult. The primary obstacle in verifying safety is that the agents operate in complex, stochastic environments perceived through high-dimensional measurements. Purely formal (symbolic) verification techniques require realistic environmental models to make the verification result meaningful in practice. Environmental models can be constructed from first principles (e.g. via a pinhole camera model or by tracing LiDAR rays), but such models cannot easily account for unexpected stochastic complexities of real systems such as LiDAR reflections (Ivanov et al. (2020a)). Alternatively, environmental models could be learned via a generative network (Katz et al. (2022)), but verification tools are difficult to scale to the individual pixel level, particularly for closed-loop systems (Everett (2021)). Besides, the real-world validity of such neural models remains in question. At the other extreme, purely statistical verification approaches are appealing because they capture statistical uncertainty — but cannot exploit the knowledge of the underlying system dynamics.

Many popular methods combine statistical and symbolic safety verification of systems with neural perception. The general approach follows two steps: first, obtain high-confidence bounds

* Co-last authors: equal contribution in supervision.

on uncertain quantities (e.g., neural perception error) using a statistical tool such as conformal prediction (CP); second, compute high-confidence reachable sets using a symbolic description of the dynamics (Lin and Bansal (2024), Muthali et al. (2023), Jafarpour et al. (2024), Geng et al. (2024)). A persistent challenge for these approaches, however, is *overly conservative reachable sets*, particularly over long time horizons. In response, many methods aim to reduce conservatism in CP bounds for a variety of settings (Romano et al. (2019), Sharma et al. (2024), Kiyani et al. (2024), Tumu et al. (2024)). One particular approach by Cleaveland et al. (2024) reformulates the weighted conformity scores to optimize for tighter bounds than the point-wise ones by Lindemann et al. (2023). In effect, it exploits the fact that the conformal errors are *heteroskedastic* over time. We observe that in practice, however, error is often highly correlated with state (e.g., motion blur is increased at higher speeds). Our key insight is that neural perception errors are *heteroskedastic with respect to state*, and this heteroskedasticity can be utilized to reduce conservatism in symbolic reachability analysis.

In this work, we introduce **state-dependent conformal bounds** for neural perception error as an "interface" between neural perception and symbolic verification. To do this, we propose two methods to partition the state space into regions via gradient-free optimization methods. The regions are optimized such that the regional perception errors contribute minimally to over-approximation error in the symbolic reachability calculation. Our approach balances the number of regions (which can decay our guarantee due to the union bound) and the size of the error in each region.

The proposed neuro-symbolic verification method opens the path to a new level of assurance for autonomous systems. The high-level approach is to use symbolic techniques for well-understood parts of the system (e.g., dynamics) and data-driven methods for high-dimensional and hard-to-model aspects (e.g., perception). Specifically, we abstract the perception model and obtain high-confidence data-driven bounds on the abstracted system, which are then used to construct tight, high-confidence reachable sets using a symbolic verification tool by leveraging the system dynamics. The ultimate output of our approach is a safety guarantee that provably holds with a user-specified probability. Our contributions are as follows:

- A framework for providing statistical safety guarantees on neural perception and control systems that exploits heteroskedasticity in perception error over the state space.
- A method for finding state-dependent neural perception error bounds via conformal prediction. The bounds and regions are selected with gradient-free optimization methods to reduce overapproximation error in symbolic high-confidence reachability computations.
- A case study on mountain car demonstrating our conformal bounds lead to significantly smaller reachable sets than the state-of-the-art time-based conformal prediction methods.

Related Work Conformal prediction, originally introduced by Vovk et al. (2005) and Shafer and Vovk (2008), is an increasingly popular method for obtaining data-driven uncertainty bounds. As conformal prediction has expanded to a variety of applications, safety of autonomous systems has gained particular attention with recent examples including safe motion planning (Lindemann et al. (2023)), safe controller design (Yang et al. (2023)), online safety monitoring (Zhang et al. (2024), Zhao et al. (2024)), and integration into safety decision-making frameworks (Lekeufack et al. (2024)). We refer the reader to Lindemann et al. (2024) and Angelopoulos et al. (2023b) for detailed tutorials and a broader overview of the conformal prediction field.

Deterministic reachability for neural control systems is a well-developed area. Open-loop methods focus on verifying input-output properties of networks (Wang et al. (2021); Dutta et al. (2018);

Katz et al. (2017); Tran et al. (2020)), while closed-loop methods interleave neural networks with symbolic dynamics to calculate reachable sets (Ivanov et al. (2019); Dutta et al. (2019); Huang et al. (2019); Fan et al. (2020); Wang et al. (2023)).Recently, Chakraborty and Bansal (2023) used reachability analysis on image-controlled systems to discover unsafe initial sets, but it requires exhaustive querying of a simulator's perception map for all states, preventing analysis of real systems.

In this work, we combine conformal prediction and neural network reachability to consider high-probability reachable sets for neural perception and control systems with known dynamics. While methods exist for reachability of stochastic systems with known or unknown dynamics (Abate et al. (2007, 2008); Lin and Bansal (2023); Alanwar et al. (2023); Bortolussi and Sanguinetti (2014)), they do not consider neural components for perception or control. One notable approach from Hashemi et al. (2024) combines neural network reachability and conformal predictions but focuses on high-confidence reachability for systems with *unknown dynamics*.

2. Problem Formulation

We consider dynamical systems with perception of the form

$$x_{k+1} = f(x_k, u_k); \quad z_k = p(x_k); \quad y_k = nn(z_k) := g(x_k) + v_k; \quad u_k = h(y_k), \tag{1}$$

where $x_k \in \mathcal{X} \subset \mathbb{R}^n$ are the system states (e.g., position, velocity); $z_k \in \mathbb{R}^{m_z}$ are the measurements (e.g., camera images) generated from an unknown perception map p; $y_k \in \mathbb{R}^{m_y}$ are the outputs of a neural component nn trained to extract a desired function g of the states from images (e.g., state estimates); v_k is the unknown random noise introduced by the neural component nn; f is the known plant dynamics model, and h is a known controller.[1]

Our reasoning for this model choice is as follows. First, note that (1) models a standard system with neural perception where the neural component is trained to extract a low-dimensional symbolic representation of the measurements (e.g., car location within the lane). As discussed in prior work by Dean et al. (2020), this formulation enables a separation-principle-like control design where the controller can be developed specifically for g (e.g., a linear measurement), while being robust to high-probability bounds on v_k (and thus abstracting away the unknown and complex map p). Similarly, this formulation enables a high-confidence verification approach that abstracts away p and verifies safety for the entire system, as long as high-confidence bounds on v_k are known. Thus, in the remainder of the paper, we will only focus on the following *abstracted system*:

$$x_{k+1} = f(x_k, u_k); \quad y_k = g(x_k) + v_k; \quad u_k = h(y_k). \tag{2}$$

Background: reachability analysis. Consider a system such as the one in (2), where we are given a known initial set $\mathcal{X}_0$ and known bounds on the noise $\|v_k\| \leq b$. Reachability analysis aims to calculate reachable sets $\mathcal{X}_1, \ldots, \mathcal{X}_T$ that are guaranteed to contain the state x_k at each time k (e.g., so as to verify that no unsafe states are reached). The reachable sets are typically conservatively approximated using computationally convenient shapes such as ellipsoids (Althoff (2015)) or Taylor models (Chen et al. (2012)). Unfortunately, worst-case bounds on v_k in (2) may be impossible to obtain without strong and often unrealistic assumptions. Thus, the problem considered in this paper is to compute reachable sets that hold *with high probability* over random initial conditions and

1. While the proposed framework can handle the more general setting with the dynamics noise, for simplicity we assume no dynamics noise in the problem statement.

noise trajectories. These reachable sets are useful in a number of ways, e.g., for high-confidence pre-deployment guarantees, online monitoring, or planning around other agents.

To obtain high-confidence reachable sets, we assume we are given a dataset of N trajectories $D = \{(x_{1,0:T}, y_{1,0:T}) \ldots, (x_{N,0:T}, y_{N,0:T})\}$, where $x_{i,0:T} = (x_{i,0}, \ldots, x_{i,T})$ is the full trajectory i of states (same for measurements). To keep our notation simple, we assume all trajectories have the same length. We also assume all trajectories are generated using controller h to ensure that they are on-policy and independently identically distributed (IID), such that each $x_0 \sim \mathcal{D}_0$ is sampled from an unknown distribution $\mathcal{D}_0$ over a known set $\mathcal{X}_0$. Finally, we also assume $v_k \sim \mathcal{V}_{k|k-1}$, i.e., the noise at each step is sampled from an unknown conditional distribution, $\mathcal{V}_{k|k-1}$, given the previous noise values and the initial state. As part of future work discussed in Section 5, we will investigate the off-policy problem where trajectories are generated using an exploration controller.

Problem 1 (High-Confidence Reachability) *Given the system in* (2)*, a confidence level* α*, and a calibration dataset of trajectories* D*, the goal is to construct a* sequence of reachable sets $\mathcal{X}_1, \ldots, \mathcal{X}_T$ *such that* $\mathbb{P}_{x_0 \sim \mathcal{D}_0, v_k \sim \mathcal{V}_{k|k-1}}[\forall k = 0..T : x_k \in \mathcal{X}_k] \geq 1 - \alpha$.

While Problem 1 can be solved using existing (purely statistical) time-series conformal prediction, e.g., by Lindemann et al. (2023), the resulting sets would inevitably be conservative without any knowledge of system dynamics. The main benefit of knowing the dynamics model f is that it allows us to use *reachability analysis* to solve Problem 1, e.g., the authors' tool Verisig (Ivanov et al. (2019)) — as long as high-confidence bounds on perception noise v_k are available.

Background: scalar conformal prediction. In the scalar CP setting by Angelopoulos et al. (2023b), we are given a calibration dataset $D = \{z_1, \ldots, z_N\}$, where the z_i are realizations of exchangeable random variables $Z_1, \ldots, Z_N$, i.e., $\mathbb{P}[Z_{q(1)} \leq \cdots \leq Z_{q(N)}] = \mathbb{P}[Z_{r(1)} \leq \cdots \leq Z_{r(N)}]$ for any two re-ordering functions q and r. Consider a new random variable Z_{test} that is also exchangeable with the Z_i. Assuming the z_i are sorted in increasing order, one can show that $\mathbb{P}[Z_{test} \leq z_{\lceil (N+1)(1-\alpha) \rceil}] \geq 1 - \alpha$. In other words, the (normalized) $1 - \alpha$ quantile, denoted by $\text{Quantile}(D, 1 - \alpha)$, is a high-confidence upper bound on a new exchangeable sample.

Background: conformal prediction for time-series data. The time-series CP setting, e.g., as considered by Cleaveland et al. (2024), is more challenging. Here, $D = \{z_{1,0:T}, \ldots, z_{N,0:T}\}$, and the problem is to design a bound function η such that $\mathbb{P}[\forall k = 0..T : Z_{test,k} \leq \eta(k)] \geq 1 - \alpha$, i.e., the probability that the entire trajectory Z_{test} is within the η bounds is at least $1 - \alpha$. A straightforward solution is to apply the scalar bounds for each time step and then obtain trajectory-wide guarantees using the probability union bound; however, this approach results in overly conservative confidence. To overcome this challenge, researchers, e.g., Cleaveland et al. (2024) and Angelopoulos et al. (2023a), have proposed to re-weigh the bounds $\eta(k)$ at each time step k to tighten up the bounds.

Novel setting: state-dependent conformal prediction. As noted above, we aim to obtain high-confidence bounds on perception noise v_k that would enable reachability analysis as a solution to Problem 1. Although such bounds on v_k can be directly obtained using time-series conformal prediction, they tend to be conservative: existing works reduce conservativeness by exploiting heteroskedasticity (i.e., varied uncertainty in v_k) *over time*. In contrast, we put forward a more effective approach to exploit heteroskedasticity *over the state space*. Since perception error likely varies drastically within the state space, we expect state-dependent bounds to separate high-noise from low-noise regions and result in much tighter reachable sets.

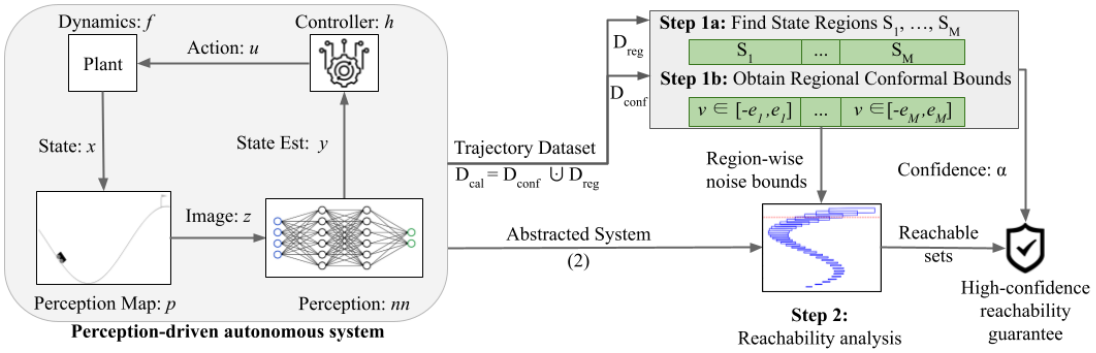


Figure 1: An overview of our approach to high-confidence reachability.

To be precise, we propose to partition the state space into M disjoint regions: $\mathcal{X} = \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_M$. Each state x will correspond to a piecewise-constant perception error bound $\eta(x)$ determined by the region $\mathcal{S}_i \ni x$, where η will be chosen to satisfy a high-confidence guarantee on the noise within each region: $\mathbb{P}[\forall k = 0..T : (x_k \in \mathcal{S}_i \Rightarrow \|v_k\| \leq \eta(x_k))] \geq 1 - (\alpha/M)$. The per-region guarantees would allow us to maintain the overall high-confidence guarantee from Problem 1 (proved in Section 3), with the added benefit that the final reachable sets, $\mathcal{X}_i$, may be much tighter than those obtained through time-series CP. Of course, this approach requires a suitable partitioning of the state space, which is the main problem of this paper.

Problem 2 (State-Dependent Conformal Prediction) *Given the system in (2), a confidence level α, and a calibration dataset D, the goal is to* partition the state space *into M disjoint regions, i.e., $\mathcal{X} = \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_M$ and compute a corresponding* noise bound function $\eta(x)$*, so as to minimize a loss function $\mathcal{L}(D, \mathcal{S}_1, \ldots, \mathcal{S}_M)$ correlated with tighter reachable sets $\mathcal{X}_i$, as defined in Problem 1.*

Remark *Problem 2 has two parts: 1) identifying a suitable loss function $\mathcal{L}$ and 2) solving the resulting optimization problem. Both of these parts are the main contributions of this paper.*

3. Approach

This section presents the proposed approach, starting with the theoretical results that demonstrate its soundness and followed by the algorithms to partition the state space and compute reachable tubes.

3.1. Approach Overview

At a high level, the proposed approach consists of two steps (corresponding to Problems 2 and 1, respectively), illustrated in Figure 1. Step 1 is to partition the state space $\mathcal{X} = \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_M$ and design a region-based function $\eta(x)$ such that $\mathbb{P}_{x_0 \sim \mathcal{D}_0, v_k \sim \mathcal{V}_{k|k-1}}[\forall k = 0..T : \|y_k - g(x_k)\| \leq \eta(x_k)] \geq 1 - \alpha$, where $\mathcal{D}_0$ and $\mathcal{V}_{k|k-1}$ are unknown, but $\mathcal{D}_0$ is assumed to have support over a known set $\mathcal{X}_0$. Step 2 is to perform worst-case (deterministic) reachability analysis of the abstracted system in (2) from initial set $\mathcal{X}_0$. Here, v_k is treated as bounded noise with state-dependent bounds $\eta(x)$. The rest of this subsection shows that the condition in Step 1 ensures the reachable sets in Step 2 solve Problem 1. The following subsections provide a specific approach for each step, leading to tight reachable sets.

Step 1: region-based perception noise bounds. Before we discuss how to partition the state space (in Section 3.2), we first outline the requirements for this partition. In particular, we aim to

apply a union bound over all regions, so each region must satisfy the following upper bound on perception error: $\mathbb{P}_{x_0\sim\mathcal{D}_0,v_k\sim\mathcal{V}_{k|k-1}}[\exists k = 0..T : (x_k \in \mathcal{S}_i \wedge \|y_k - g(x_k)\| > \eta(x_k))] \leq (\alpha/M)$.[2] The next proposition shows how this guarantee leads to the overall guarantee over the entire state space. All proofs are provided in the appendix.

Proposition 1 *Consider the abstracted system in (2) with the state space partitioned into M regions, $\mathcal{X} = \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_M$, and assume a high-confidence bound within each region:*

$$\mathbb{P}_{x_0\sim\mathcal{D}_0,v_k\sim\mathcal{V}_{k|k-1}}[\exists k = 0..T : (\|y_k - g(x_k)\| > \eta(x_k) \wedge x_k \in \mathcal{S}_i)] \leq \frac{\alpha}{M}.$$

Then the trajectory-wide bound holds: $\mathbb{P}_{x_0\sim\mathcal{D}_0,v_k\sim\mathcal{V}_{k|k-1}}[\exists k = 0..T : \|y_k - g(x_k)\| > \eta(x_k)] \leq \alpha$.

Step 2: High-confidence reachability analysis. Given the trajectory-wide guarantee on v_k, the following theorem shows that performing worst-case reachability analysis using η bounds on v_k will produce reachable sets that satisfy the condition in Problem 1.

Theorem 2 *Consider the abstracted system in (2), with x_0 sampled from an unknown distribution $\mathcal{D}_0$ with support over a known set $\mathcal{X}_0$. Suppose we are given a bound function η such that $\mathbb{P}_{x_0\sim\mathcal{D}_0,v_k\sim\mathcal{V}_{k|k-1}}[\exists k = 0..T : \|y_k - g(x_k)\| > \eta(x_k)] \leq \alpha$. Suppose worst-case reachable sets $\mathcal{X}_1, \ldots, \mathcal{X}_T$ are computed for (2), with initial set $\mathcal{X}_0$ and noise bounds $\|v_k\| \leq \eta(x_k)$. Then:*

$$\mathbb{P}_{x_0\sim\mathcal{D}_0,v_k\sim\mathcal{V}_{k|k-1}}[\forall k = 0..T : x_k \in \mathcal{X}_k] \geq 1 - \alpha.$$

3.2. State-Dependent Conformal Prediction

This subsection presents an optimization-based approach for partitioning the state space into regions $\mathcal{X} = \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_M$ that 1) satisfy the condition in Proposition 1 and 2) reduce the approximation error incurred by the subsequent reachability task. We first describe how to calculate trajectory-wide guarantees per region for *any* partition. We define *region-specific trajectory subsets*:

$$D_{\mathcal{S}_i} = \{(x_{1,m_1:n_1}, y_{1,m_1:n_1}), \ldots, (x_{N,m_N:n_N}, y_{N,m_N:n_N}) \mid x_{i,j} \in \mathcal{S}_i\},$$

where we consider the sub-trajectory of each $x_{i,0:T}$ which is contained in $\mathcal{S}_i$; note that the time steps in each sub-trajectory are the same as in the full one. Given $D_{\mathcal{S}_i}$, the *sub-trajectory non-conformity scores* are defined as the maximum sub-trajectory-wide perception error within each region,

$$\delta^j_{\mathcal{S}_i} = \max_{t=m_j..n_j} \|g(x_{j,t}) - y_{j,t}\| \text{ for } j = 1..N; \text{ and } \delta^{N+1}_{\mathcal{S}_i} = \infty \tag{3}$$

Next, we apply scalar conformal prediction to these non-conformity scores to obtain the corresponding perception error confident bound for each region for Proposition 1.

Proposition 3 (High-confidence region-based perception error bound) *Given a confidence level α and sub-trajectory error dataset $\Delta_i = \{\delta^1_{\mathcal{S}_i}, \ldots, \delta^{N+1}_{\mathcal{S}_i}\}$ for each region $\mathcal{S}_i$, the following* perception error bound $\eta(x)$ *satisfies the conditions in Proposition 1:*

$$\eta(x) = \text{Quantile}\left(\Delta_i, 1 - \frac{\alpha}{M}\right) \text{ if } x \in \mathcal{S}_i.$$

2. It is possible to require that different regions have different confidence bounds, as long as the overall guarantee of $1 - \alpha$ is reached. For simplicity, however, we require all regions to have the same $1 - (\alpha/M)$ guarantee.

For conformal perception bounds in each region, two loss functions are applied to optimize the partitioning: *Experience Loss (EL)* and *Experience Time-Decay Loss (ETDL)*. EL is designed to prioritize frequently visited regions by assigning higher weights. This strategy tightens conformal error bounds in these areas, under the assumption that they are of greater significance for the reachability analysis since we need to inflate reachable sets more frequently. Less frequently visited regions receive lower weights to balance the optimization process. EL is defined as:

$$\mathcal{L}_{EL} = \sum_{i=1}^{M} \sum_{x_{j,t} \in D_{\mathcal{S}_i}} w_i \eta(x_{j,t}), \text{ where the weights are } w_i = \frac{|D_{\mathcal{S}_i}|}{\sum_{j=1}^{M} |D_{\mathcal{S}_j}|}. \tag{4}$$

ETDL extends EL by incorporating a time-decay weighting strategy. Since over-approximation errors tend to accumulate over time, ETDL assigns larger weights to states earlier in the trajectory. This helps avoid accumulating error early during verification. ETDL is defined below:

$$\mathcal{L}_{ETDL} = \sum_{i=1}^{M} \sum_{x_{j,t} \in D_{\mathcal{S}_i}} w_i \lambda_t \eta(x_{j,t}), \tag{5}$$

where λ_t is a decreasing, time-dependent weight function; we use exponential decay in our experiments: $\lambda_t = 0.9^t$. We are now ready to state the region-based optimization problem considered in this paper.

Definition 4 (Reachability-Informed Region Optimization) *Given a calibration dataset of trajectories D and a confidence bound α,* the reachability-informed region optimization problem *is to select M regions that:*

$$\begin{aligned} \min_{\mathcal{S}_1,\ldots,\mathcal{S}_M} \ & \mathcal{L}, \text{ where } \mathcal{L} = \begin{cases} \mathcal{L}_{EL}, & \text{if EL is chosen} \\ \mathcal{L}_{ETDL}, & \text{if ETDL is chosen} \end{cases} \\ \text{s.t. } & \mathcal{X} = \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_M, \\ & \Delta_i = \{\delta_{\mathcal{S}_i}^1, \ldots, \delta_{\mathcal{S}_i}^{N+1}\}, i = 1, \ldots, M, \text{ and} \\ & \eta(x) = \text{Quantile}\left(\Delta_i, 1 - \frac{\alpha}{M}\right) \text{ if } x \in \mathcal{S}_i, i = 1, \ldots, M. \end{aligned} \tag{6}$$

Solving the optimization problem. The problem in (6) is ill-defined for arbitrary shapes for $\mathcal{S}_i$. As a first step, we consider boxes for the regions and two *gradient-free global search algorithms*: Genetic Algorithm (GA) (Mirjalili (2019)) and Simulated Annealing (SA) (van Laarhoven (1987)). Both methods are well-suited for this problem because they do not require gradient information and can effectively search for globally optimal partitions of the state space. GA explores the solution through evolutionary computation: selection, crossover, and mutation to refine candidate region solutions iteratively. In contrast, SA operates through stochastic perturbations of regions, using a probabilistic criterion to escape local minima while gradually converging on an optimal solution.

3.3. Reachable Tube Computation

Given the state region partitions $\mathcal{S}_1, \ldots, \mathcal{S}_M$ and perception bound function η, we have now obtained high-confidence bounds for the unknown random noise v_k in the abstracted system from (2). For each region, we define the *conformal error bound* explicitly: $e_i = \eta(x_k)$ if $x_k \in \mathcal{S}_i$ for $i = 1, \ldots, M$. From Theorem 2, in the case of L_∞ norm bounds on v_k, we have that $v_k \in [-e_i, e_i]$. Thus, the system for which we need to compute reachable sets becomes:

$$x_{k+1} = f(x_k, u_k); \quad y_k = g(x_k) + [-e_i, e_i] \quad \text{if } x_k \in S_i; \quad u_k = h(y_k). \tag{7}$$

To compute high-confidence reachable set analysis for our abstracted dynamical system as defined in Problem 1, we can use any reachability tool for hybrid systems. One such tool is the authors' tool Verisig (Ivanov et al. (2021)). To encode the regional perception bounds in a hybrid system, we add transitions between the plant f and controller h with guards determined by the regions ($x_k \in \mathcal{S}_i$) and resets that inflate the measurement model $g(x_k)$ with the corresponding error interval $[-e_i, e_i]$.

Note that introducing additional transitions to a system can make scalability challenging: reachable sets that intersect with multiple regions must be considered separately. This leads to longer verification time as each "branch" must be verified separately. In highly-branching verification tasks, individual branches are often strict subsets of other branches with larger reachable sets, leading to redundant verification. Next, we describe our method to remove this redundancy in Verisig.

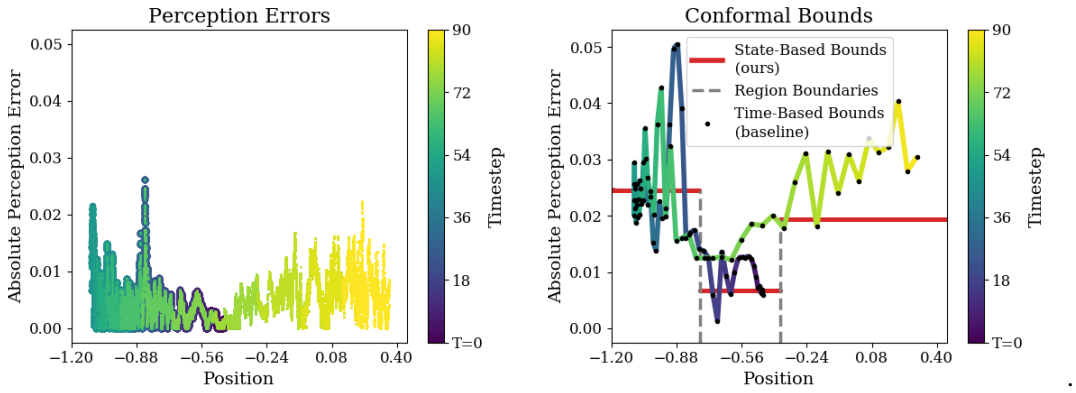
In Verisig, reachable sets are represented with *Taylor Models* (TMs), introduced by Makino and Berz (2003). Informally, a TM encloses a function f over a specified domain. Formally, a TM for a function f is an over-approximation for f containing a polynomial p_f and worst-case error bound I_f for a given domain D, such that $f(x) \in \{p_f(x) + e \mid e \in I_f\}\ \forall x \in D$. To remove redundant branches, we aim to identify when one TM "parent" branch encloses another. In general, this is not trivial because TM ranges are evaluated via interval arithmetic and produce boxes from their symbolic and error components. Thus, a conservative method to check for inclusion is to transform the "parent" into a box and check whether a conservative approximation of the "child" is fully within this box. While this method may introduce additional error due to transforming "parent" TMs, Verisig already implements *shrink-wrapping* of TMs to reduce long-term over-approximation whenever remainders grow large (Ivanov et al. (2021)). Shrink-wrapping resets TMs to be fully-symbolic and contain their original range with no remainder. Thus, we opportunistically check for redundant subset branches whenever a branch is shrink-wrapped. Any subset branches of a newly shrink-wrapped branch are removed from the verification, thus enhancing its scalability.

4. Case Study: Mountain Car

We evaluate the proposed neuro-symbolic verification method on Mountain Car (MC), a popular reinforcement learning benchmark from OpenAI's Gymnasium. Consistently throughout, we measure our results against the time-series approach from Cleaveland et al. (2024) and refer to this approach as the "baseline". As the baseline requires a fixed time horizon, we use $T = 90$. In all methods, we set $\alpha = 0.05$ so that computed reachable sets contain the real trajectory with 95% confidence.

Control and Perception Models. Per Section 2, we consider a modular control pipeline with a low dimensional (i.e. state-based) neural controller and a perception model that extracts low-dimensional representations from high-dimensional observations. In particular, we use a controller h adopted from Ivanov et al. (2020b) that was pre-trained and pre-verified to be safe (i.e., reaching the top of the hill with a reward of at least 90) when observing the ground-truth position and velocity starting from the initial set $p_0 \in [-0.55, -0.45]$. For the perception model, we use a state estimator nn that predicts the position x of the car using a gray-scaled image from the MC simulator. Since we cannot produce velocity estimates from single images, we provide the controller with ground truth velocity and leave multi-image perception for future work. See Appendix B for details.

Data Collection. We generated a dataset D of 4,000 trajectories by simulating MC with the perception model nn and controller h described above. The initial states $\mathcal{X}_0$ are sampled uniformly from $[-0.55, -0.45]$, and trajectories are terminated after reaching the goal state $x = 0.45$. We emphasize that while the perception model was trained on pre-deployment data with contrast noise, the



(*a*) Perception Errors for all 4,000 trajectories in D showing heteroskedasticity over time and state.

(*b*) State-based regional conformal bounds for $M = 3$ regions vs. the baseline bounds.

Figure 2: Perception errors and their respective bounds under our method and time-based baseline.

dataset D was generated by adding blur noise to image observations so as to demonstrate that the proposed method can handle out-of-distribution deployment noise on the perception model (details in Appendix B). The perception errors for this dataset are shown in Figure 2(*a*) – note the drastic heteroskedasticity over the state space exposed by the added blur noise. We split D evenly into 2,000-trajectory calibration and test sets, D_{cal} and D_{test}, respectively. D_{cal} is further split into two disjoint sets: D_{reg} for determining region edges via (6) and D_{conf} for finding the regional conformal bounds. D_{test} is reserved for testing the conservativeness of the probabilistic guarantees.

Conformal Bound Computations. We compute regions and regional conformal bounds in three ways. First, we use our state-based method with all combinations of optimization algorithms {SA, GA}, loss functions {EL, ETDL}, and regions $M \in \{2, \ldots, 7\}$. To solve for the regions via (6), we randomly select 500 trajectories from D_{cal} and use the remaining 1,500 trajectories to find conformal bounds. Second, as an ablation, we compute conformal bounds based on partitioning the state space *uniformly* into $M = \{1, \ldots, 7\}$ equally sized regions, using all 2,000 trajectories in D_{cal} for conformal bounds (as we do not need to synthesize regions). Third, we compute time-based conformal bounds for the baseline comparison. Using the algorithm described by Cleaveland et al. (2024), we randomly select 100 trajectories from D_{cal} to set the α values and use the remaining 1,900 to set the conformal bounds. Figure 2(*b*) illustrates regions and conformal bounds for GA+ETDL ($M = 3$) and the time-based conformal bounds for an example trajectory.

Reachable Set Size Evaluation. Table 1 summarizes the average verification time and maximal reachable set sizes under each experiment. To compute reachable sets for our state-based methods, we follow the approach described in Section 3.3 and encode the regional perception errors in Verisig with M discrete jumps between the dynamics and controller, corresponding to each region. For the time-based method, a different perception error bound is used at each time step, as per the bounds shown in Figure 2(*b*). We compute reachable sets under each experimental condition from a restricted initial position set of $\mathcal{X}_0 = [-0.51, -0.49]$. For each experiment, the verification for the initial set $\mathcal{X}_0$ was carried out in parallel with 200 sub-intervals of size 0.0001. The average time to compute reach sets for each of the 200 initial subsets is shown in Table 1. For further evaluation, Figure 3 provides a visual comparison between the reachable sets produced by our best method (GA + ETDL, $M = 7$) and by the baseline for the entire initial set $\mathcal{X}_0 = [-0.55, -0.45]$.

	Average Time to Compute 90 Steps [s]							Max Reachable Set Size over 90 Steps						
Algorithm \ M	1	2	3	4	5	6	7	1	2	3	4	5	6	7
Uniform	1,066	2,308	4,167	3,386	19,824	9,059	9,745	0.939	0.883	0.406	0.230	0.520	0.251	0.203
SA + EL	-	2,356	2,674	3,855	5,318	6,021	7,195	-	0.458	0.225	0.226	0.213	0.218	0.210
SA + ETDL	-	2,401	2,075	2,371	2,963	4,054	4,737	-	0.448	**0.200**	0.205	0.165	0.164	0.162
GA + EL	-	2,490	2,871	3,970	9,580	12,231	9,601	-	0.458	0.225	0.207	0.167	0.167	0.145
GA + ETDL	-	2,405	2,084	2,261	2,729	3,323	5,249	-	0.456	0.203	**0.168**	**0.163**	**0.154**	**0.115**
Time-based baseline Cleaveland et al. (2024)	**1,044**							0.225						

Table 1: Max reachable set size and average computation time per initial subset for our state-based conformal bounds compared to the baseline time-series bounds from Cleaveland et al. (2024).

Results & Discussion. Overall, the genetic algorithm finds the smallest reachable set sizes. The most notable improvements come from our timed-decayed loss function ETDL, which greatly improves verification time and reduces reachable sets as compared to EL alone. This confirms the intuition that incurring error early in the verification process disproportionately impacts the resulting reachable sets. As compared to the baseline, our best-performing algorithm and loss combination GA+ETDL produces smaller reachable set sizes for all $M \geq 3$, though computing the reachable sets is much slower due to reachable sets potentially intersecting multiple regions at the same time, as noted in Section 3.3. See Appendix C for additional analysis of the subset merging optimization to handle this scalability challenge.

5. Future Work and Conclusion

In this paper, we presented a novel approach to finding state-dependent conformal bounds for a neural perception model and utilize knowledge of the system dynamics to produce high-confidence reachable sets via a symbolic verification tool. Our case study demonstrates our methods can produce dramatically tighter reachable sets than the state-of-the-art conformal method based on time series.

In future work, we plan to extend our methods to partition more state dimensions. These additional dimensions will motivate scalability improvements for optimization methods, data usage, and verification complexity. We also intend to investigate the case where the perception data is collected **off-policy**, i.e., using an exploratory controller. This would enable us to use an adaptive controller at run-time which may navigate dynamical environments with high confidence. Furthermore, we will investigate the **off-model** problem, where guarantees from simulations or a lab-tested system are extended to the deployed system. Such methods would allow us to bridge the simulation-to-reality gap and enable the rapid and high-confidence development of safe autonomous systems.

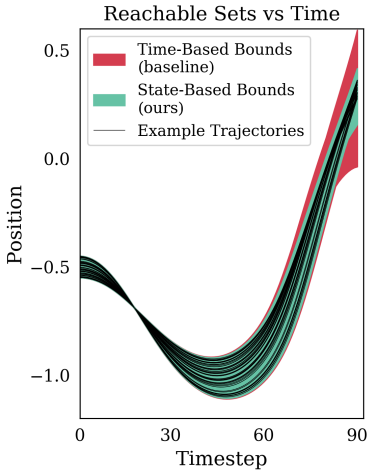


Figure 3: Reachable Sets for our state-based conformal perception bounds (GA+ETDL, $M = 7$) vs. the time-based baseline bounds. Our reachable sets are tighter than the time-based method *at all timesteps*.

Acknowledgments

This work was supported by the NSF Grants CCF-2403615 and CCF-2403616. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF) or the US Government.

References

Alessandro Abate, Saurabh Amin, Maria Prandini, John Lygeros, and Shankar Sastry. Computational approaches to reachability analysis of stochastic hybrid systems. In *International Workshop on Hybrid Systems: Computation and Control*, pages 4–17. Springer, 2007.

Alessandro Abate, Maria Prandini, John Lygeros, and Shankar Sastry. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Automatica*, 44(11):2724–2734, 2008.

Amr Alanwar, Anne Koch, Frank Allgöwer, and Karl Henrik Johansson. Data-driven reachability analysis from noisy data. *IEEE Transactions on Automatic Control*, 68(5):3054–3069, 2023.

Matthias Althoff. An introduction to cora 2015. *ARCH@ CPSWeek*, 34:120–151, 2015.

Anastasios Angelopoulos, Emmanuel Candes, and Ryan J Tibshirani. Conformal pid control for time series prediction. *Advances in neural information processing systems*, 36:23047–23074, 2023a.

Anastasios N Angelopoulos, Stephen Bates, et al. Conformal prediction: A gentle introduction. *Foundations and Trends® in Machine Learning*, 16(4):494–591, 2023b.

Luca Bortolussi and Guido Sanguinetti. A statistical approach for computing reachability of nonlinear and stochastic dynamical systems. In *International Conference on Quantitative Evaluation of Systems*, pages 41–56. Springer, 2014.

Kaustav Chakraborty and Somil Bansal. Discovering closed-loop failures of vision-based controllers via reachability analysis. *IEEE Robotics and Automation Letters*, 8(5):2692–2699, 2023.

Xin Chen, Erika Abraham, and Sriram Sankaranarayanan. Taylor model flowpipe construction for non-linear hybrid systems. In *2012 IEEE 33rd Real-Time Systems Symposium*, pages 183–192. IEEE, 2012.

Alex Clark. Pillow (pil fork) documentation, 2015. https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf.

Matthew Cleaveland, Insup Lee, George J Pappas, and Lars Lindemann. Conformal prediction regions for time series using linear complementarity programming. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 20984–20992, 2024.

Sarah Dean, Nikolai Matni, Benjamin Recht, and Vickie Ye. Robust guarantees for perception-based control. In *Learning for Dynamics and Control*, pages 350–360. PMLR, 2020.

S. Dutta, S. Jha, S. Sankaranarayanan, and A. Tiwari. Output range analysis for deep feedforward neural networks. In *NASA Formal Methods Symposium*, pages 121–138. Springer, 2018.

Souradeep Dutta, Xin Chen, and Sriram Sankaranarayanan. Reachability analysis for neural feedback systems using regressive polynomial rule inference. In *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, pages 157–168, 2019.

Michael Everett. Neural network verification in control. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 6326–6340. IEEE, 2021.

Jiameng Fan, Chao Huang, Xin Chen, Wenchao Li, and Qi Zhu. Reachnn*: A tool for reachability analysis of neural-network controlled systems. In *International Symposium on Automated Technology for Verification and Analysis*, pages 537–542. Springer, 2020.

Yuang Geng, Jake Brandon Baldauf, Souradeep Dutta, Chao Huang, and Ivan Ruchkin. Bridging dimensions: Confident reachability for high-dimensional controllers. In *International Symposium on Formal Methods*, pages 381–402. Springer, 2024.

Gymnasium. Mountain car. https://gymnasium.farama.org/environments/classic_control/mountain_car_continuous/.

Navid Hashemi, Lars Lindemann, and Jyotirmoy V Deshmukh. Statistical reachability analysis of stochastic cyber-physical systems under distribution shift. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(11):4250–4261, 2024.

Chao Huang, Jiameng Fan, Wenchao Li, Xin Chen, and Qi Zhu. Reachnn: Reachability analysis of neural-network controlled systems. *ACM Transactions on Embedded Computing Systems (TECS)*, 18(5s):1–22, 2019.

R. Ivanov, T. Carpenter, J. Weimer, R. Alur, G. J. Pappas, and I. Lee. Case study: Verifying the safety of an autonomous racing car with a neural network controller. In *International Conference on Hybrid Systems: Computation and Control*, 2020a.

Radoslav Ivanov, James Weimer, Rajeev Alur, George J Pappas, and Insup Lee. Verisig: verifying safety properties of hybrid systems with neural network controllers. In *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, pages 169–178, 2019.

Radoslav Ivanov, Taylor J Carpenter, James Weimer, Rajeev Alur, George J Pappas, and Insup Lee. Verifying the safety of autonomous systems with neural network controllers. *ACM Transactions on Embedded Computing Systems (TECS)*, 20(1):1–26, 2020b.

Radoslav Ivanov, Taylor Carpenter, James Weimer, Rajeev Alur, George Pappas, and Insup Lee. Verisig 2.0: Verification of neural network controllers using taylor model preconditioning. In *Computer Aided Verification: 33rd International Conference, CAV 2021, Virtual Event, July 20–23, 2021, Proceedings, Part I*, pages 249–262. Springer, 2021.

Saber Jafarpour, Zishun Liu, and Yongxin Chen. Probabilistic reachability analysis of stochastic control systems. *arXiv preprint arXiv:2407.12225*, 2024.

G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017.

Sydney M Katz, Anthony L Corso, Christopher A Strong, and Mykel J Kochenderfer. Verification of image-based neural network controllers using generative models. *Journal of Aerospace Information Systems*, 19(9):574–584, 2022.

Shayan Kiyani, George Pappas, and Hamed Hassani. Length optimization in conformal prediction. *arXiv preprint arXiv:2406.18814*, 2024.

Jordan Lekeufack, Anastasios N Angelopoulos, Andrea Bajcsy, Michael I Jordan, and Jitendra Malik. Conformal decision theory: Safe autonomous decisions from imperfect predictions. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11668–11675. IEEE, 2024.

Albert Lin and Somil Bansal. Generating formal safety assurances for high-dimensional reachability. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10525–10531. IEEE, 2023.

Albert Lin and Somil Bansal. Verification of neural reachable tubes via scenario optimization and conformal prediction. In *6th Annual Learning for Dynamics & Control Conference*, pages 719–731. PMLR, 2024.

Lars Lindemann, Matthew Cleaveland, Gihyun Shim, and George J Pappas. Safe planning in dynamic environments using conformal prediction. *IEEE Robotics and Automation Letters*, 2023.

Lars Lindemann, Yiqi Zhao, Xinyi Yu, George J Pappas, and Jyotirmoy V Deshmukh. Formal verification and control with conformal prediction. *arXiv preprint arXiv:2409.00536*, 2024.

Kyoko Makino and Martin Berz. Taylor models and other validated functional inclusion methods. *International Journal of Pure and Applied Mathematics*, 6:239–316, 2003.

Seyedali Mirjalili. *Genetic Algorithm*. Springer, 2019. doi: 10.1007/978-3-319-93025-1_4.

Anish Muthali, Haotian Shen, Sampada Deglurkar, Michael H Lim, Rebecca Roelofs, Aleksandra Faust, and Claire Tomlin. Multi-agent reachability calibration with conformal prediction. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 6596–6603. IEEE, 2023.

Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. *Advances in neural information processing systems*, 32, 2019.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

Apoorva Sharma, Sushant Veer, Asher Hancock, Heng Yang, Marco Pavone, and Anirudha Majumdar. Pac-bayes generalization certificates for learned inductive conformal prediction. *Advances in Neural Information Processing Systems*, 36, 2024.

H. Tran, S. Bak, W. Xiang, and T. T. Johnson. Verification of deep convolutional neural networks using imagestars. In *32nd International Conference on Computer-Aided Verification (CAV)*. Springer, July 2020.

Renukanandan Tumu, Matthew Cleaveland, Rahul Mangharam, George Pappas, and Lars Lindemann. Multi-modal conformal prediction regions by optimizing convex shape templates. In *6th Annual Learning for Dynamics & Control Conference*, pages 1343–1356. PMLR, 2024.

Peter J. M. van Laarhoven. *Simulated annealing*. Springer, 1987. doi: 10.1007/978-94-015-7744-1_2.

VoloCity. VoloCity:The air taxi that's a cut above. https://www.volocopter.com/en/solutions/volocity.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J Zico Kolter. Beta-CROWN: Efficient bound propagation with per-neuron split constraints for complete and incomplete neural network verification. *Advances in Neural Information Processing Systems*, 34, 2021.

Yixuan Wang, Weichao Zhou, Jiameng Fan, Zhilu Wang, Jiajun Li, Xin Chen, Chao Huang, Wenchao Li, and Qi Zhu. Polar-express: Efficient and precise formal reachability analysis of neural-network controlled systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(3):994–1007, 2023.

Waymo. Waymo: The World's Most Experienced Driver. https://waymo.com/.

Shuo Yang, George J Pappas, Rahul Mangharam, and Lars Lindemann. Safe perception-based control under stochastic sensor uncertainty using conformal prediction. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 6072–6078. IEEE, 2023.

Yunchuan Zhang, Sangwoo Park, and Osvaldo Simeone. Bayesian optimization with formal safety guarantees via online conformal prediction. *IEEE Journal of Selected Topics in Signal Processing*, 2024.

Yiqi Zhao, Bardh Hoxha, Georgios Fainekos, Jyotirmoy V Deshmukh, and Lars Lindemann. Robust conformal prediction for stl runtime verification under distribution shift. In *2024 ACM/IEEE 15th International Conference on Cyber-Physical Systems (ICCPS)*, pages 169–179. IEEE, 2024.

Appendix A. Proofs

A.1. Proof of Proposition 1

Consider the event $A = \{\exists k = 0..T : \|y_k - g(x_k)\| > \eta(x_k)\}$. Since the $\mathcal{S}_i$ are disjoint, we can bound the probability of A as follows:

$$\mathbb{P}_{x_0 \sim \mathcal{D}_0, v_k \sim \mathcal{V}_{k|k-1}}[A] = \mathbb{P}_{x_0 \sim \mathcal{D}_0, v_k \sim \mathcal{V}_{k|k-1}}\left[\bigcup_{i=1}^{M} \{\exists k = 0..T : (\|y_k - g(x_k)\| > \eta(x_k) \wedge x_k \in \mathcal{S}_i)\}\right] \leq \alpha,$$

where the inequality follows from the union bound.

A.2. Proof of Theorem 2

Consider the event $A = \{\exists k = 0..T : x_k \notin \mathcal{X}_k\}$. Since the sets $\mathcal{X}_i$ are worst-case reachable sets, then it must be the case that $\|v_l\| > \eta(x_l)$ for some $l \leq k$., i.e.,

$$A = \{\exists k = 0..T : (x_k \notin \mathcal{X}_k \wedge \exists l \leq k : \|v_l\| > \eta(x_l))\}.$$

However, we know that the noise bounds hold with probability $1 - \alpha$ over the entire trajectory, so $\mathbb{P}_{x_0 \sim \mathcal{D}_0, v_k \sim \mathcal{V}_{k|k-1}}[A] \leq \alpha$, and the result follows.

A.3. Proof of Proposition 3

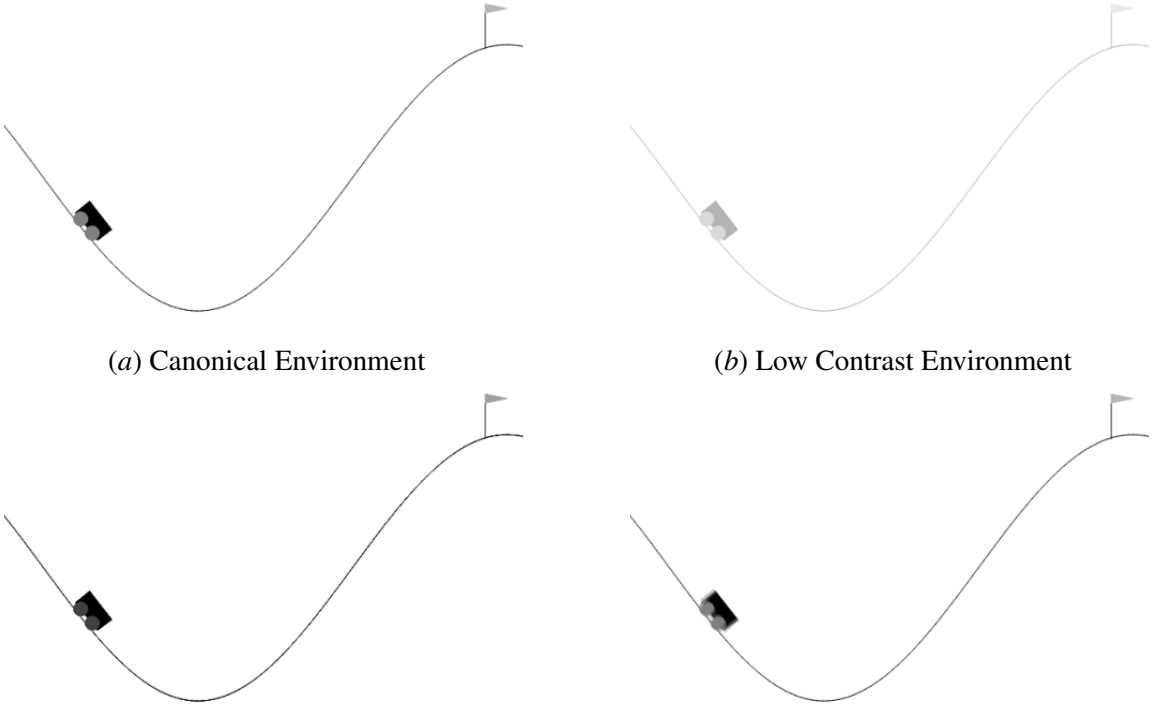
Since each set Δ_i contains the maximal errors per trajectory within the corresponding region $\mathcal{S}_i$, the (normallized) $1 - (\alpha/M)$ quantile provides a high-confidence bound on the trajectory-wide error within $\mathcal{S}_i$.

Appendix B. Case Study Details

MC Background. Mountain Car is a common yet challenging reinforcement learning benchmark in which an underpowered car must reach the top of the right hill as shown in Figure 4(*a*). Because the car is underpowered, a successful controller must first utilize the left hill to gain momentum before reaching the goal on the right side. The car dynamics are shown in (8) where $p \in [-1.2, 0.6]$ is the position, $v \in [-0.07, 0.07]$ is the velocity, and $u \in [-1.0, 1.0]$ is the control thrust. The initial position is the bottom of the mountain with $p_0 \in [-0.55, -0.45]$ and at rest with $v_0 = 0$. The dynamics for the standard system are as follows:

$$\begin{aligned} p_{k+1} &= p_k + v_k \\ v_{k+1} &= v_k + 0.0015u_k - 0.0025\cos(3p_k) \end{aligned} \tag{8}$$

Case Study System & Environment. For our case study, we use neural networks for both the state-based controller $h : p \times v \mapsto u$ and an image-based perception model $nn : z \mapsto \hat{p}$ that observes images z of the MC environment and produces a state estimate $\hat{p}$. We use the simulator as the canonical perception map $s : p \mapsto z$ to map positions to 400×600 pixel gray-scaled images of the



(*a*) Canonical Environment

(*b*) Low Contrast Environment

(*c*) High Contrast Environment

(*d*) Blurred Environment

Figure 4: Images of the mountain car environment

environment. Thus, our case study system is as follows:

$$\begin{aligned} z_k &= s(p_k) \\ \hat{p}_k &= nn(z_k) \\ u_k &= h(\hat{p}_k, v_k) \\ v_{k+1} &= v_k + 0.0015u_k - 0.0025\cos(3p_k) \\ p_{k+1} &= p_k + v_k \end{aligned} \tag{9}$$

We add contrast and blur noise to images for training and deployment, respectively, of the perception model. In particular, we consider modified perception maps and noise parameters α and δ. For contrast, the modified perception map $s_c : p \times \alpha \mapsto z_c$ creates a contrasted image z_c. Contrast is added using the Python Image Library (PIL) ImageEnhance module where $\alpha = 0$ produces a solid gray image, $\alpha = 1$ produces the original image, and $\alpha > 1$ produces a higher contrast version of the original image (Clark (2015)). The blur perception map $s_b : p \times \delta \mapsto z_b$ creates a blurred image z_b. Blur is added as follows: $z_b = s_b(p, \delta) = 0.5s(p - \delta) + s(p) + 0.5s(p + \delta)$, i.e., a canonical image with a lighter overlay of left and right shifted images. Blurred images are then normalized to [0, 1].

Controller. The controller, h, is a neural network that takes position and velocity as inputs, has two hidden layers of 16 neurons with sigmoid activations, and has one output neuron with tanh activation. This controller was pre-trained and pre-verified to be safe by Ivanov et al. (2019), i.e., it reaches the goal with a reward of at least 90 when starting in the initial set $p_0 \in [-0.59, -0.4]$

when observing ground truth position and velocity. For this case study, we consider the initial set $p_0 \in [-0.55, -0.45]$ for which the controller is more robust.

Perception Model. The state estimator nn is a convolutional neural network (CNN). The input is a single channel (gray-scaled) 400×600 pixel image followed by 2 (convolutional + max pooling) layers with 16 internal channels followed by 2 hidden linear layers or 100 neurons and a single output neuron. The convolutional layers have kernels of 32 and 24, and the pooling kernel is size 16. Stride is 2 for both convolution and pooling. All internal activations functions are ReLU, and the output is a scaled and shifted Tanh such that outputs are in the range of the MC position: $[-1.2, 0.6]$. The model is trained on contrasted and canonical images (see Figures 4(*a*)-4(*c*)) generated from 100 equally-spaced positions and 9 contrast levels from $\alpha \in [0.1, 2.0]$ for a total of 1000 samples. The model was trained for 1000 epochs with MSE Loss.

Trajectory Data Collection. During data collection, we added out-of-distribution blur noise to images with $\delta = 0.005$. See Figure 4(*b*) for for an example image.

Appendix C. Reachability Computation Optimizations

	Average Time to Compute 90 Steps [s]							**Max Reachable Set Size over 90 Steps**						
M / **Algo**	1	2	3	4	5	6	7	1	2	3	4	5	6	7
G + ETDL (Greedy Merge)	-	**1,471**	**1,815**	**1,617**	**1,819**	**2,130**	**3,096**	-	0.606	0.287	0.246	0.237	0.219	0.154
G + ETDL (OPP Merge)	-	2,405	2,084	2,261	2,729	3,323	5,249	-	**0.456**	**0.203**	**0.168**	**0.163**	**0.154**	**0.115**
G + ETDL (No Merge)	-	3,260	2,508	2,822	4,463	5,687	11,605	-	**0.456**	**0.203**	**0.168**	**0.163**	**0.154**	**0.115**

Table 2: Comparison of time to compute reachable sets and reachable sets sizes based for different subset merging algorithms.

As described in Section 3.3, shrink-wrapping in the verification tool allows to remove or *merge* redundant branches with existing branches opportunistically. Table 2 shows how this opportunistic method (OPP Merge) reduces verification time while not introducing additional over-approximation error as compared to not removing redundant branches (No Merge). As an extension, we could additionally allow increased approximation error in the interest of time. One way is to greedily shrink-wrap branches whenever they *would* have children to be removed. This method (Greedy Merge) is shown in Table 2, and the results demonstrate the the corresponding time decrease and over-approximation increase. Future work will consider other optimizations for this scalability challenge, particularly as the number of regions increase with additional state dimensions.