Robust Verification of Controllers under State Uncertainty via Hamilton-Jacobi Reachability Analysis

Albert Lin ALBERTKL@STANFORD.EDU
Stanford University, CA, US
Alessandro Pinto ALESSANDRO.PINTO@JPL.NASA.GOV
NASA Jet Propulsion Laboratory, California Institute of Technology, CA, US
Somil Bansal SOMIL@STANFORD.EDU
Stanford University, CA, US

Editors: G. Sukhatme, L. Lindemann, S. Tu, A. Wierman, N. Atanasov

Abstract

As perception-based controllers for autonomous systems become increasingly popular in the real world, it is important that we can formally verify their safety and performance despite *perceptual uncertainty*. Unfortunately, the verification of such systems remains challenging, largely due to the complexity of the controllers, which are often nonlinear, nonconvex, learning-based, and/or black-box. Prior works propose verification algorithms that are based on approximate reachability methods, but they often restrict the class of controllers and systems that can be handled or result in overly conservative analyses. Hamilton-Jacobi (HJ) reachability analysis is a popular formal verification tool for general nonlinear systems that can compute optimal reachable sets under worst-case system uncertainties; however, its application to perception-based systems is currently underexplored. In this work, we propose RoVer-CoRe, a framework for the **Ro**bust **Ver**ification of **Co**ntrollers via HJ **Re**achability. To the best of our knowledge, RoVer-CoRe is the first HJ reachability-based framework for the verification of perception-based systems under perceptual uncertainty. Our key insight is to concatenate the system controller, observation function, and the state estimation modules to obtain an equivalent closed-loop system that is readily compatible with existing reachability frameworks. Within RoVer-CoRe, we propose novel methods for formal safety verification and robust controller design. We demonstrate the efficacy of the framework in case studies involving aircraft taxiing and NN-based rover navigation. Code is available at the link in the footnote[1].

Keywords: Hamilton-Jacobi reachability analysis, perceptual uncertainty, formal verification

1. Introduction

As perception-based controllers for autonomous systems become increasingly popular in the real world, it is important that we can formally verify their safety and performance despite *perceptual uncertainty*. Perceptual uncertainty can result from inherent noise in the robot's sensors, inevitable learning errors in learning-based modules, as well as environmental factors, e.g., when a tunnel blocks the reception of a GPS signal. In this work, we are interested in computing formal guarantees for such systems despite worst-case perceptual errors. For example, we would like to verify that a rover will not enter safety-critical keep-out zones even if it encounters adversarial errors in a vision-based state estimation module. Unfortunately, verifying perception-based systems remains challenging, largely due to the complexity of the controllers, which are often nonlinear, nonconvex, learning-based, and/or black-box (Edwards et al., 2024). These qualities make the formal analysis of controllers a difficult problem on its own, even without perceptual uncertainty.

1. https://github.com/albertklin/rover-core

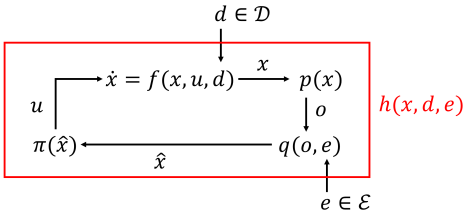

Figure 1: (Black) A perception-based system under perceptual uncertainty e, which result in noisy state estimates $\hat{x}$. (Red) The closed-loop system abstraction used for safety analysis.

Since many perception-based controllers are implemented with neural networks (NN) due to their ability to process general inputs, a substantial body of prior works focuses on verifying such systems by combining NN verification tools with control-theoretic verification methods (Arjomand-Bigdeli et al., 2024; Trapiello et al., 2023; Wang et al., 2024; Everett, 2021; Clavière et al., 2021; Zhao et al., 2022; Rossi et al., 2024; Everett, 2021). The most common approaches leverage approximate reachable set methods to verify the closed-loop system as a whole (Clavière et al., 2021; Zhang et al., 2024; Xiang and Shao, 2022; Dutta et al., 2018; Schilling et al., 2022; Yang et al., 2020; Xiang and Johnson, 2018; Rossi et al., 2024; Zhang et al., 2023; Tran et al., 2020; Ivanov et al., 2019; Fan et al., 2020; Everett et al., 2021; Newton and Papachristodoulou, 2022). However, such methods are heavily constrained by the conservatism of the chosen propagation tools, which can result in a failure to certify system safety. We propose to overcome this limitation by using Hamilton-Jacobi (HJ) reachability analysis for the reachability computation (Bansal et al., 2017). Our key insight is that HJ reachability can provide a less conservative result than existing approaches because it computes the optimal reachable sets for general nonlinear dynamical systems.

Building on this insight, we propose RoVer-CoRe, a framework for the **Ro**bust **Ver**ification of **Co**ntrollers via HJ **Re**achability. RoVer-CoRe concatenates the system controller, observation function, and the state estimation modules to obtain an equivalent closed-loop system, where the only external input is perceptual uncertainty. This equivalent closed-loop system is readily handled under existing HJ reachability tools after appropriate adjustments. Within RoVer-CoRe, we show that the main challenge in applying HJ reachability tools comes from computing the closed-loop system Hamiltonian. To overcome this challenge, we propose different methods to bound the Hamiltonian depending on the form of the controller, resulting in potentially conservative yet sound analyses. Finally, we demonstrate the efficacy of RoVer-CoRe for formal safety verification and robust controller design on case studies involving aircraft taxiing and NN-based rover navigation. To our knowledge, RoVer-CoRe is the first HJ reachability-based framework for verifying perception-based systems. In summary, our contributions include:

- RoVer-CoRe, an HJ reachability-based framework for verifying perception-based systems,
- methods for formal safety verification and robust controller design under uncertainty, and
- case studies demonstrating the efficacy of the proposed framework, with code open-sourced[1].

2. Related Works

2.1. Approximate Reachable Set Methods

Previous works combine NN verification tools with approximate reachable set methods to verify NN-controlled systems under perceptual uncertainty (Everett, 2021). In such works, NN verification tools bound the closed-loop system dynamics subject to bounded perturbations in the controller's observations. Then, approximate reachable sets can be computed using the bounded dynamics.

To efficiently propagate sets for nonlinear systems, approximate methods employ various finite set representations, such as hyper-rectangles (Clavière et al., 2021; Xiang and Johnson, 2018; Rossi et al., 2024), polytopes (Dutta et al., 2018; Yang et al., 2020; Zhang et al., 2023; Everett et al., 2021; Newton and Papachristodoulou, 2022), ellipsiods, zonotopes (Zhang et al., 2024; Trapiello et al., 2023; Tran et al., 2020), support functions, and Taylor models (Schilling et al., 2022; Ivanov et al., 2019), or make linear modeling approximations (ArjomandBigdeli et al., 2024). However, these representations and the methods used to propagate them can incur significant approximation costs, resulting in conservative analyses that can fail to certify safety. In Appendix A, we empirically quantify this conservatism by comparing RoVer-CoRe against NNV 2.0 (Lopez et al., 2023), a popular verification toolbox for NN control systems (NNCS). Alternative approaches have been proposed to compute tighter set approximations, but they are restricted to certain classes of system dynamics and controllers. We propose to overcome these limitations by replacing approximate set propagation tools with HJ reachability analysis, which aims to compute optimal reachable sets.

2.2. Certificate-Based Methods

Recent works extend certificate-based methods, particularly control barrier functions (CBFs), to verify safety under state uncertainty. For example, Measurement-Robust CBFs (MR-CBFs) strengthen the standard CBF inequality using bounds on the estimation error to guarantee safety for all states consistent with that bound (Dean et al., 2021; Cosner et al., 2021). However, an MR-CBF is not always guaranteed to exist, depending on the magnitude of the error bound. Robust CBFs (R-CBFs) improve upon MR-CBFs by introducing a robustifying term that handles estimation errors without requiring a fixed error bound a priori (Nanayakkara et al., 2025). Most recently, an adaptive extension to R-CBFs has been proposed to reduce conservatism and infeasibility (Das et al., 2025). Despite this progress, existing CBF-based methods suffer from common limitations, including key challenges with global feasibility and barrier construction, which can lead to safety violations in practice. In contrast, our proposed framework provides a globally constructive mechanism that can handle general nonlinear systems and failure set geometries.

2.3. HJ Reachability Analysis

Our proposed framework builds upon HJ reachability analysis, which computes the optimal reachable set and associated controller under worst-case disturbances that capture model mismatches, external forces, or other adversarial effects (Bansal et al., 2017). At a first glance, it may appear natural to treat perceptual uncertainty as another such adversarial disturbance. However, the standard HJ formulation assumes that the controller has perfect state information, optimizing the control input based on the true system state. Relaxing this assumption fundamentally alters the structure of the underlying differential game, requiring a reformulation of the value function to accomodate partial observability, which is a direction we leave for future work. In this paper, we instead focus on verifying closed-loop systems induced by fixed perception-based controllers, which, perhaps surprisingly, aligns naturally with the existing HJ reachability framework after appropriate adjustments. To the best of our knowledge, RoVer-CoRe is the first framework to use HJ reachability to provide formal guarantees for systems operating under perceptual uncertainty.

3. Problem Setup

We model a perception-based system with state $x \in \mathcal{X}$, control $u \in \mathcal{U}$, disturbance $d \in \mathcal{D}$, and dynamics $\dot{x} = f(x, u, d)$ governing how x evolves over time until a final time horizon T.

The system observation (or output) $o \in \mathcal{O}$ is given by a general observation map $p : o = p(x)$ which is a function of the underlying state x. For example, o can correspond to an image taken by an onboard camera sensor or the output of a motion capture system. We assume that the system obtains a state estimate $\hat{x} \in \mathcal{X}$ by a general estimator $q : \hat{x} = q(o, e)$ which is a function of the observation o and a perceptual error input $e \in \mathcal{E}$. For example, q can represent a noisy state-estimation filter or an NN-based state-estimation module that contains learning errors. Let $\pi : \mathcal{X} \to \mathcal{U}$ denote a state-based controller which maps the state estimate to a control input, i.e., $u = \pi(\hat{x})$. See Figure 1 for an illustration of the overall system model.

We denote the set of failure states as $\mathcal{F} \subseteq \mathcal{X}$ (e.g., collision states) which the system is not allowed to enter. The failure set can be represented by the zero-sublevel set of a Lipschitz-continuous function $l : \mathcal{X} \to \mathbb{R}$, i.e., $x \in \mathcal{F} \Leftrightarrow l(x) \leq 0$. Let $\xi_{x_0,t_0}^{\pi \circ q(p(\cdot),e(\cdot)),d(\cdot)}(\tau)$ denote the state achieved at time $\tau \in [t_0, T]$ by starting at initial state x_0 at time t_0 and applying the control policy π over $[t_0, \tau]$ under the error signal $e(\cdot)$ and disturbance signal $d(\cdot)$. We are interested in verifying the safety of the system starting from state x_0 and time t_0 under π in the presence of the worst-case error signal $e(\cdot)$ and the worst-case disturbance signal $d(\cdot)$ until the finite time horizon T. In other words, we want to verify that $\forall \tau \in [t_0, T], \forall e(\cdot), \forall d(\cdot), \xi_{x_0,t_0}^{\pi \circ q(p(\cdot),e(\cdot)),d(\cdot)}(\tau) \notin \mathcal{F}$.

4. Background

4.1. HJ Reachability Analysis

In this section, we explain HJ reachability analysis in the context of a traditional system verification problem, where there is no perception-based controller. In Section 5, we show how we can extend the framework to verify perception-based controllers.

HJ reachability analysis is concerned with computing the system's Backward Reachable Tube, which we denote as BRT (Lygeros, 2004; Mitchell et al., 2005). We define BRT as the set of all initial states $x \in \mathcal{X}$ starting from which, for all control signals $\mathbf{u}(\cdot)$, there exists a disturbance signal $\mathbf{d}(\cdot)$ such that the system will enter the failure set $\mathcal{F}$ within the time horizon $[t_0, T]$:

$$\text{BRT} := \{x \in \mathcal{X} : \forall \mathbf{u}(\cdot), \exists \mathbf{d}(\cdot), \exists \tau \in [t_0, T], \xi_{x,t_0}^{\mathbf{u}(\cdot),\mathbf{d}(\cdot)}(\tau) \in \mathcal{F}\}. \tag{1}$$

The BRT complement precisely captures the set of states for which we can guarantee system safety.

In HJ reachability, computing BRT is formulated as a robust optimal control problem. First, we implicitly represent the failure set $\mathcal{F}$ by a failure function $l(x)$ whose zero-sublevel set yields $\mathcal{F}$: $\mathcal{F} = \{x \in \mathcal{X} : l(x) \leq 0\}$. $l(x)$ is commonly the signed distance function to $\mathcal{F}$. Next, we define the cost function corresponding to a control signal $\mathbf{u}(\cdot)$ and disturbance signal $\mathbf{d}(\cdot)$ to be the minimum of $l(x)$ over the trajectory starting from a state x and time t:

$$J_{\mathbf{u}(\cdot),\mathbf{d}(\cdot)}(x, t) := \min_{\tau \in [t,T]} l(\xi_{x,t}^{\mathbf{u}(\cdot),\mathbf{d}(\cdot)}(\tau)). \tag{2}$$

Since the control aims to avoid $\mathcal{F}$ under worst-case disturbance, the value function corresponding to this robust optimal control problem is:

$$V(x, t) := \max_{\mathbf{u}(\cdot)} \min_{\mathbf{d}(\cdot)} J_{\mathbf{u}(\cdot),\mathbf{d}(\cdot)}(x, t). \tag{3}$$

By defining our optimal control problem in this way, we can recover BRT using the value function. The value function being non-positive implies that the failure function is non-positive somewhere along the optimal trajectory, or in other words, that the system will inevitably enter $\mathcal{F}$. Conversely, the value function being positive implies that there exists a control signal that will prevent the system from entering $\mathcal{F}$ even under the worst-case disturbance signal. Thus, BRT is computed as the zero-sublevel set of the value function:

$$\text{BRT} = \{x \in \mathcal{X} : V(x, t_0) \leq 0\}. \tag{4}$$

Using the principles of optimality and dynamic programming, it can be shown that the value function in Equation (3) can be computed as the solution to the following final value Hamilton-Jacobi-Isaacs Variational Inequality (HJI-VI):

$$\begin{aligned} \min\{D_t V(x,t) + H(x,t,\nabla V(x,t)), l(x) - V(x,t)\} = 0, \\ V(x,T) = l(x), \quad \forall t \in [t_0, T]. \end{aligned} \tag{5}$$

$D_t V(x,t)$ and $\nabla V(x,t)$ represent the temporal derivative and spatial gradient of the value function $V(x,t)$, respectively. The Hamiltonian $H(x,t,\nabla V(x,t))$ encodes how the control and disturbance interact with the system dynamics:

$$H(x,t,\nabla V(x,t)) \coloneqq \max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} \nabla V(x,t) \cdot f(x,u,d). \tag{6}$$

The value function in Equation (3) also induces the optimal safety controller:

$$\mathbf{u}^*(x,t) \coloneqq \arg\max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} \nabla V(x,t) \cdot f(x,u,d). \tag{7}$$

Intuitively, the optimal safety controller aligns the system dynamics in the direction of the value function's gradients, thus steering the system towards higher-value states, i.e., away from $\mathcal{F}$. It can be shown that safety is guaranteed despite worst-case disturbances if the system starts outside of BRT and applies the control in Equation (7) at the BRT boundary (Borquez et al., 2024).

Unfortunately, the analysis above assumes that the controller has access to the true system state. When perceptual uncertainty is present, this assumption is violated, so the classical reachability formulation cannot be applied directly. This motivates our use of a different closed-loop system abstraction induced by a perception-based controller, which we introduce next for the safety analysis.

5. Approach

As discussed above, traditional HJ reachability frameworks are not equipped to handle perceptual uncertainty directly. To address this limitation, we introduce RoVer-CoRe. Section 5.1 presents the key closed-loop system abstraction that enables the use of HJ reachability tools, under which the main technical challenge becomes optimizing the closed-loop Hamiltonian. Section 5.2 proposes methods to efficiently compute or bound this Hamiltonian and examines the resulting safety guarantees. Finally, Section 5.3 discusses how the verification results support robust controller design.

5.1. Closed-Loop System Abstraction

To begin our analysis, we observe that for a fixed controller, the only mutable inputs to the system are the disturbance signal $d(\cdot)$ and error signal $e(\cdot)$. Treating both as adversarial, we aim to apply HJ

reachability tools to the resulting closed-loop behavior. Our *key idea* is to concatenate the controller, observation map, and state estimator to form the closed-loop dynamics:

$$h(x, d, e) \coloneqq f(x, \pi(q(p(x), e)), d), \tag{8}$$

whose only external inputs are d and e. This abstraction is illustrated in Figure 1.

Under this representation, the system interface aligns with traditional HJ reachability methods. The main difficulty, however, lies in evaluating the closed-loop Hamiltonian, which becomes:

$$H(x, t, \nabla V(x, t)) = \min_{d \in \mathcal{D}, e \in \mathcal{E}} \nabla V(x, t) \cdot h(x, d, e). \tag{9}$$

Due to the complexity of the controller $\pi(\cdot)$, observation map $p(\cdot)$, and estimator $q(\cdot)$, the Hamiltonian can be a non-convex function of d and e, making Equation (9) challenging to solve. This difficulty is the central obstacle in verifying closed-loop systems under perceptual uncertainty. In the next section, we propose methods to address this challenge.

5.2. Computing or Bounding the Closed-Loop Hamiltonian

In this section, we propose methods to efficiently compute or bound the Hamiltonian in Equation (9) and discuss their implications on the computed guarantees. Users of RoVer-CoRe should determine which of the proposed methods best suits their particular verification problem.

As a first step, we propose to assume that the state estimate $\hat{x}$ is the result of a bounded additive perturbation to the underlying state, i.e., $\hat{x} = q(p(x), e) = x + e$. This step can be taken without loss of generality, since the noise bounds can always be chosen large enough to contain all perceptual errors encountered during deployment, albeit at the cost of conservatism. This assumption greatly simplifies the application of the proposed methods described next. We defer the treatment of more general observation maps and state estimators, such as via generative NNs, to future work.

5.2.1. EXACTLY COMPUTING THE HAMILTONIAN

In some cases, particularly when the controller is enumerable or has a simple structure, e.g., a linear feedback controller, the optimization problem in Equation (9) can be solved exactly. In such settings, we can apply HJ reachability tools directly to the closed-loop abstraction with no conservatism. The verification process also yields the corresponding worst-case disturbance and error signals, which serve as concrete counterexamples when safety is violated. We illustrate this procedure for the taxiing controller in Section 6.1 and the grid-based MPC in Section 6.2.

5.2.2. LOWER BOUNDING THE HAMILTONIAN

When it is not possible to exactly solve the optimization problem in Equation (9) due to the complexity of the controller, we propose to instead compute a lower bound on the closed-loop Hamiltonian, which will result in a conservative yet sound verification of the system (Choi et al., 2025). Specifically, let $\underline{\mathcal{U}}(x)$ and $\overline{\mathcal{U}}(x)$ represent element-wise lower and upper bounds on the controller output at state x under perceptual uncertainty, i.e., $\underline{\mathcal{U}}(x) \leq \pi\left(q\left(p\left(x\right), e\right)\right) \leq \overline{\mathcal{U}}(x), \forall e \in \mathcal{E}$. Let $\mathcal{U}(x)$ denote the hyperrectangle defined by $\underline{\mathcal{U}}(x)$ and $\overline{\mathcal{U}}(x)$. We can lower-bound Equation (9):

$$H(x, t, \nabla V(x, t)) \geq \min_{d \in \mathcal{D}, u \in \mathcal{U}(x)} \nabla V(x, t) \cdot f(x, u, d), \tag{10}$$

which takes a form similar to the open-loop Hamiltonian in Equation (6) and thus aligns with existing HJ reachability tools. Intuitively, we robustly handle perceptual uncertainty by capturing all possibilities with $\mathcal{U}(x)$. $\mathcal{U}(x)$ can be computed efficiently depending on the form of the controller. For example, if the controller is represented on a grid, we can compute $\mathcal{U}(x)$ by enumeration. If the controller is an NN, we can compute $\mathcal{U}(x)$ using state-of-the-art NN verification tools. We illustrate this procedure for the NN-based controller in Section 6.2.

Remark 1 *The conservatism of the analysis will scale with the degree with which $\mathcal{U}(x)$ overapproximates the true controller output space. Although we do not explore it here, we suggest that users can trade off conservatism with computational complexity depending on the characteristics of the system, e.g., by representing $\mathcal{U}(x)$ as a convex hull rather than a hyperrectangle, or by budgeting more computational resources for tighter NN verification.*

5.3. Robust Controller Design

As described above, RoVer-CoRe evaluates the safety margins of fixed perception-based controllers under perceptual uncertainty. Here, we discuss how these verification results can be used in practice to improve controller robustness. In particular, we propose using RoVer-CoRe for safety-guided hyperparameter optimization. Consider a family of controllers π_α parameterized by tunable hyperparameters α. For each π_α, RoVer-CoRe computes a robust, parameter-conditioned value function V_α. By evaluating V_α over a range of α, we can select α^* which maximizes a desired safety objective, such as the safety margin at a given initial state x_0, or the volume of the safe set. This enables safety-guaranteed controller design, which we demonstrate in the next section.

6. Case Studies

6.1. Aircraft Taxiing

For our first case study, we analyze a lane-following aircraft taxiing controller (Katz et al., 2021). The aircraft evolves according to:

$$\dot{p_x} = v \sin\theta \qquad \dot{p_y} = v \cos\theta \qquad \dot{\theta} = u, \tag{11}$$

where p_x is the crosstrack error, p_y is the downtrack position, and θ is the heading error relative to the centerline. v is the linear velocity fixed at 5 m/s, and u is the commanded angular velocity. The aircraft starts at $x_0 = (0, 100, 0)$ and must follow the centerline as closely as possible using images from a wing-mounted camera. A deep neural network (DNN) module processes the images to predict the crosstrack error $\hat{p}_x$ and the heading error $\hat{\theta}$. Using these estimates, the controller computes $u := \tan(a \cdot \hat{p}_x + b \cdot \hat{\theta})$, where $a, b \leq 0$ are proportional gains on $\hat{p}_x$ and $\hat{\theta}$, respectively. The aircraft violates safety if it leaves the runway, corresponding to the failure set $\mathcal{F} = \{x : |p_x| \geq 10\}$.

The DNN predictions can include errors $e_{\hat{p}_x} = \hat{p}_x - p_x,\ e_{\hat{\theta}} = \hat{\theta} - \theta$ that induce system failure. Thus, we would like to formally verify the system under worst-case perceptual uncertainty. In Figure 2, we plot the error distribution over a test dataset of runway images generated in the X-Plane simulator (Laminar Research, 2016). Using the error distribution, we compute element-wise norm bounds $\bar{e}_{\hat{p}_x}, \bar{e}_{\hat{\theta}}$ that correspond to different coverages $0 \leq c \leq 1$ of the observed errors.

Next, we demonstrate how RoVer-CoRe can be used to verify system safety. We consider the controller $a = -0.013, b = -0.44$, as suggested in Chakraborty and Bansal (2023). Since the controller is sufficiently simple, we can compute the closed-loop Hamiltonian and conduct an exact

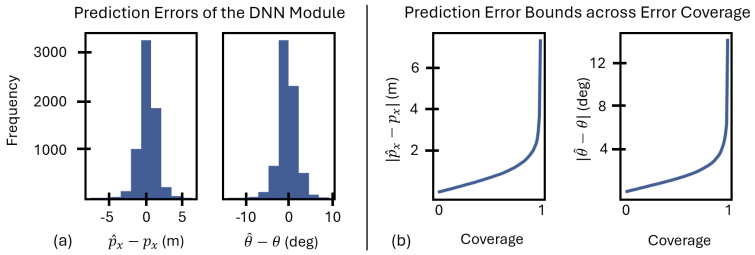

Figure 2: (a) Histograms of the prediction errors for $\hat{p}_x$ and $\hat{\theta}$ by the aircraft DNN module. (b) Prediction error bounds for different error coverages.

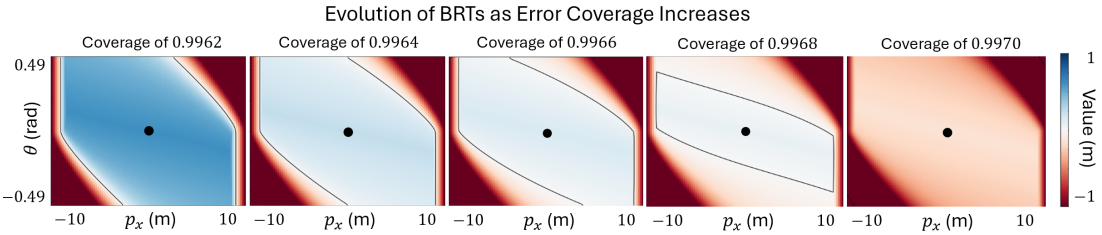

Figure 3: Evolution of the BRTs for the taxiing controller as the error coverage increases. Safety is guaranteed in the blue region. (Black) The BRT boundary. (Dot) The initial state.

verification: $H(x,t,\nabla V(x,t)) = \beta_1 v \sin\theta + \beta_2 v \cos\theta + \beta_3 \tan\left(a\left(p_x + e^*_{\hat{p}_x}\right) + b\left(\theta + e^*_{\hat{\theta}}\right)\right)$, where $\nabla V(x,t) = [\beta_1, \beta_2, \beta_3]^T$, and the worst-case errors are given by $e^*_{\hat{p}_x} = -\operatorname{sgn}(\beta_3 a)\bar{e}_{\hat{p}_x}$, $e^*_{\hat{\theta}} = -\operatorname{sgn}(\beta_3 b)\bar{e}_{\hat{\theta}}$. We compute the robust value functions across different coverages of the observed test errors using the grid-based `hj_reachability` Python toolbox (Edward Schmerling) over the state space $[-11, 11]$ m $\times$ $[100, 250]$ m $\times$ $[-0.49, 0.49]$ rad with a grid shape of $[101, 101, 101]$ for a time horizon of 20 s. Each value function takes $\leq$ 2 s to compute on an NVIDIA 3090 Ti GPU. In Figure 3, we plot the evolution of the BRTs as the error coverage increases. The plots show that we can formally verify the safety of the initial state of the system up to an error coverage of 0.9968.

Finally, we demonstrate how RoVer-CoRe can guide robust controller design. Figure 4 (a) shows the safety value of the initial state under different controllers across the 2D controller-hyperparameter space for an error coverage of 0.997. The original controller (black dot) is unsafe, whereas controllers such as the orange point within the blue region remain robust to the desired level of perceptual uncertainty. Figure 4 (b) visualizes the corresponding actual (solid) and perceived (dashed) trajectories: the orange controller preserves safety under worst-case perception errors.

To illustrate how RoVer-CoRe exposes failure modes, we also analyze the unsafe controllers marked in green and purple. The green controller's high cross-track gain causes over-correction when the aircraft misperceives its lateral position, driving it outside the runway limits. The purple controller's large heading-error gain leads the aircraft to believe it is aligned with the centerline while actually drifting off course. These results highlight how RoVer-CoRe enables both formal safety guarantees and diagnostic insights for principled controller refinement.

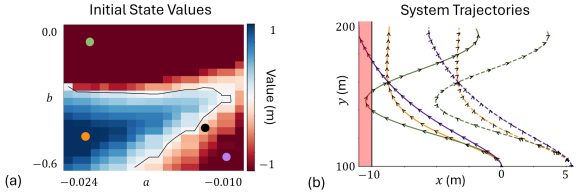

Figure 4: (a) Value function at the initial state under an error coverage of 0.997 for different controllers. (b) Actual (solid) and perceived (dashed) trajectories under worst-case uncertainty.

6.2. NN-Based Rover Navigation

Our second case study considers a rover navigating around obstacles under perceptual uncertainty, inspired by NASA's *Endurance* concept for long-range lunar night exploration (Baker et al., 2024). The rover uses visual odometry whose error grows in low-light conditions. Uncertainty can be reset to zero by turning on headlights, but this comes at a high energy cost. We model the error with an element-wise norm bound that grows linearly with the time t' since the lights were last activated: $\bar{e}^{(t')} = (\bar{e}^{(t')}_{\hat{p}_x}, \bar{e}^{(t')}_{\hat{p}_y}, \bar{e}^{(t')}_{\hat{\theta}}) = (0.1t', 0.1t', 0.02t')$. The rover follows Equation (11) with $v = 1$ m/s and violates safety upon collision. Since safety depends on the closed-loop evolution of the time-varying perceptual uncertainty, the problem is a natural fit for RoVer-CoRe.

Next, we use RoVer-CoRe to verify the rover's safety following Section 5.2.2. To show the applicability of our method, we evaluate two controllers: an expert MPC defined on a grid, for which control bounds are obtained directly via enumeration, and an NN trained to imitate the MPC, for which bounds are computed using the α, β-CROWN verifier (Wang et al., 2021). We compute bounds assuming the lights are off on a state-time grid of $[0, 20]$ m $\times$ $[-5, 5]$ m $\times$ $[-\pi, \pi]$ rad $\times$ $[0, 5]$ s with shape $4 \times [100]$, which takes $\approx$ 30 minutes on an NVIDIA 5090 GPU. The NN bounds at ($\theta = 0$ rad, $t = 1.5$ s), along with rollouts under perfect state estimation, are shown in Figure 5.

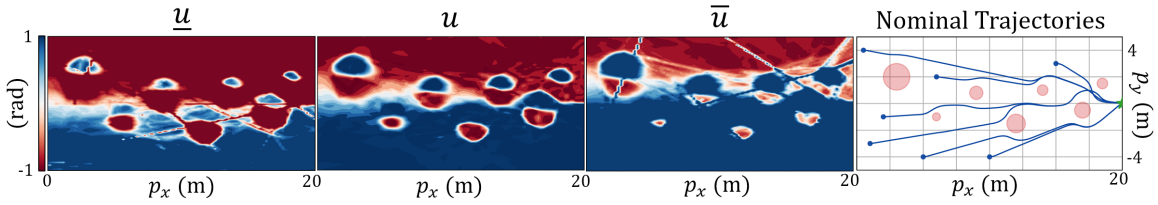

Figure 5: From left to right: the lower bounds, nominal outputs, and upper bounds of the NN at ($\theta = 0$ rad, $t = 1.5$ s). Rightmost: Rollouts by the NN-based controller successfully reach a goal (green star) while avoiding obstacles (red circles) under perfect state estimation.

After obtaining the controller bounds, we compute the corresponding robust value function using the grid-based `hj_reachability` Python toolbox (Edward Schmerling) on the same grid as above, which takes $\approx$ 20 s on an NVIDIA 5090 GPU. The BRTs for increasing time horizons T for the NN (solid) and MPC (dashed) appear in the left plot of Figure 6. The NN BRTs are consistently more inflated than those of the MPC, reflecting both imitation error and the looseness of the α, β-CROWN bounds. This yields a conservative but sound guarantee; starting from any state outside these BRTs is guaranteed to be safe under worst-case perceptual uncertainty.

To evaluate efficiency and completeness, we compare RoVer-CoRe against a naive Monte Carlo baseline for verifying the MPC. For each grid state, we roll out 10^4 trajectories under uniformly sampled state uncertainties and record the minimum safety value observed. The estimates stabilize within collecting $\frac{1}{10}$ of the samples. The resulting BRT estimate (purple, right plot of Figure 6) misses many unsafe states that RoVer-CoRe correctly identifies. Since we compute the Hamiltonian for the MPC exactly via enumeration, RoVer-CoRe also produces counterexamples, several of which are shown in the same plot. These results underscore the rigor and completeness of our method. In Appendix A, we further compare against the popular set-based NNV 2.0 tool (Lopez et al., 2023).

Finally, we show how RoVer-CoRe enables robust controller design. If the rover kept its lights on continuously, safety would be governed by the zero-uncertainty BRT. Leveraging this, we treat the zero-uncertainty BRT as a surrogate failure set and, for each state, use RoVer-CoRe to compute

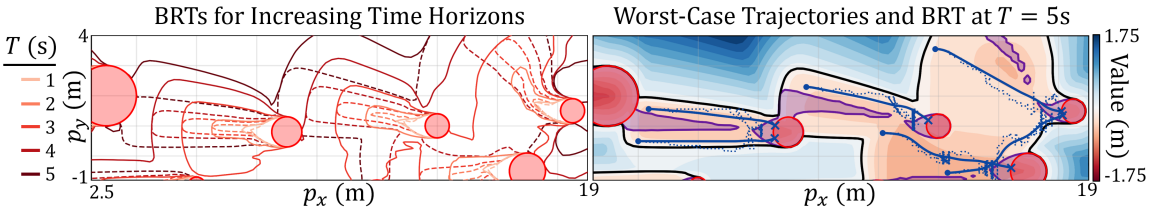

Figure 6: Left: BRT boundaries computed by RoVer-CoRe when the lights are off for the original MPC (dashed) and the NN-based controller (solid) as the time horizon T increases (color). Right: The initial-time value function for $T = 5$s for the original MPC as computed by RoVer-CoRe (color) compared with the Monte Carlo baseline (purple). True (solid) and perceived (dotted) worst-case rollouts that lead to collision are shown in blue.

the earliest time at which the growing-uncertainty dynamics would drive the rover into this set. This time corresponds exactly to the maximum duration the rover can safely operate with the lights off. Using these durations, we construct a light-activation policy that guarantees safety while reducing unnecessary energy use. Figure 7 shows the resulting policy applied to the same initial states that previously led to failure: the rover remains safe, and the lights activate only when needed. As expected, the policy triggers the lights slightly earlier for the NN-based controller, reflecting its weaker robustness. For both controllers, the rover is able to navigate in the dark for significant periods while retaining formal safety guarantees. These results highlight how RoVer-CoRe enables both rigorous safety verification and practical controller design in challenging uncertainty settings.

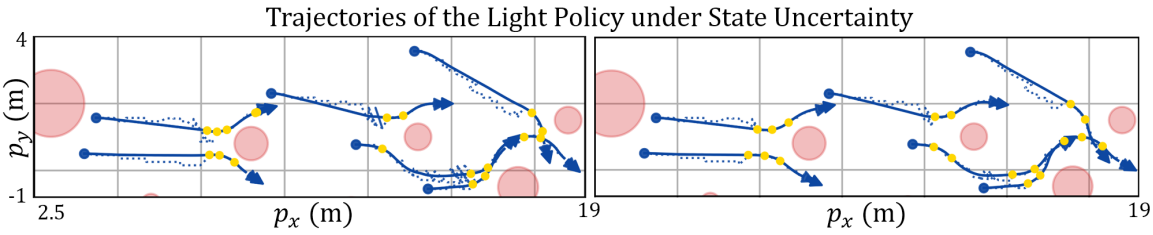

Figure 7: True (solid) and perceived (dotted) worst-case rollouts starting from the same initial states as in Figure 6, under the original MPC (left) and NN-based controller (right), which would be unsafe without lights. The light policies derived by RoVer-CoRe activate the lights (yellow dots) only when needed to guarantee safety under worst-case uncertainty.

7. Conclusion

We introduced RoVer-CoRe, the first HJ reachability-based framework to verify perception-based controllers operating under bounded perceptual uncertainty by formulating a closed-loop abstraction that unifies perception, estimation, and control. A central insight is the characterization of the closed-loop Hamiltonian, for which we develop exact and conservative bounding methods that enable safety verification for different forms of controllers. Through case studies, with code open-sourced[1], we show that RoVer-CoRe not only provides formal safety guarantees but also produces significantly tighter guarantees than set-based NNCS verification tools (Appendix A), exposes perception-induced failure modes, and supports robust controller design. Future directions include incorporating probabilistic uncertainty models, improving scalability via learning-based reachability tools, and extending the framework to richer perceptual pipelines and controller synthesis.

Acknowledgments

This work was supported in part by a NASA Space Technology Graduate Research Opportunity, the NSF CAREER Program under award 2240163, and the DARPA ANSR program. Part of the research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

Ali ArjomandBigdeli, Andrew Mata, and Stanley Bak. Verification of neural network control systems in continuous time. In Guy Avni, Mirco Giacobbe, Taylor T. Johnson, Guy Katz, Anna Lukina, Nina Narodytska, and Christian Schilling, editors, *AI Verification*, pages 100–115, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-65112-0.

John D Baker, John O Elliott, James T Keane, Nadia R Khan, Richard P Kornfeld, Hari D Nayar, and Issa A Nesnas. The endurance lunar rover sample return mission. In *2024 IEEE Aerospace Conference*, pages 1–13. IEEE, 2024.

Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J Tomlin. Hamilton-Jacobi Reachability: A brief overview and recent advances. In *IEEE Conference on Decision and Control (CDC)*, 2017.

Javier Borquez, Kaustav Chakraborty, Hao Wang, and Somil Bansal. On safety and liveness filtering using hamilton–jacobi reachability analysis. *IEEE Transactions on Robotics*, 40:4235–4251, 2024. doi: 10.1109/TRO.2024.3454470.

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. doi: 10.1109/LRA.2023.3258719.

Jason J Choi, Christopher A Strong, Koushil Sreenath, Namhoon Cho, and Claire J Tomlin. Data-driven hamiltonian for direct construction of safe set from trajectory data. *arXiv preprint arXiv:2504.03233*, 2025.

Arthur Clavière, Eric Asselin, Christophe Garion, and Claire Pagetti. Safety verification of neural network controlled systems. In *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pages 47–54, 2021. doi: 10.1109/DSN-W52860.2021.00019.

Ryan K Cosner, Andrew W Singletary, Andrew J Taylor, Tamas G Molnar, Katherine L Bouman, and Aaron D Ames. Measurement-robust control barrier functions: Certainty in safety with uncertainty in state. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6286–6291. IEEE, 2021.

Ersin Das, Rahal Nanayakkara, Xiao Tan, Ryan M Bena, Joel W Burdick, Paulo Tabuada, and Aaron D Ames. Safe navigation under state uncertainty: Online adaptation for robust control barrier functions. *arXiv preprint arXiv:2508.19159*, 2025.

Sarah Dean, Andrew Taylor, Ryan Cosner, Benjamin Recht, and Aaron Ames. Guaranteeing safety of learned perception modules via measurement-robust control barrier functions. In *Conference on Robot Learning*, pages 654–670. PMLR, 2021.

Souradeep Dutta, Susmit Jha, Sriram Sankaranarayanan, and Ashish Tiwari. Learning and verification of feedback control systems using feedforward neural networks. *IFAC-PapersOnLine*, 51(16):151–156, 2018. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2018.08.026. URL https://www.sciencedirect.com/science/article/pii/S240589631831139X. 6th IFAC Conference on Analysis and Design of Hybrid Systems ADHS 2018.

Edward Schmerling. hj_reachability: Hamilton-jacobi reachability analysis in jax. URL https://github.com/StanfordASL/hj_reachability.

Alec Edwards, Andrea Peruffo, and Alessandro Abate. A general framework for verification and control of dynamical models via certificate synthesis, 2024. URL https://arxiv.org/abs/2309.06090.

Michael Everett. Neural network verification in control. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 6326–6340, 2021. doi: 10.1109/CDC45484.2021.9683154.

Michael Everett, Golnaz Habibi, and Jonathan P. How. Efficient reachability analysis of closed-loop systems with neural network controllers. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4384–4390, 2021. doi: 10.1109/ICRA48506.2021.9561348.

Jiameng Fan, Chao Huang, Xin Chen, Wenchao Li, and Qi Zhu. Reachnn*: A tool for reachability analysis of neural-network controlled systems. In Dang Van Hung and Oleg Sokolsky, editors, *Automated Technology for Verification and Analysis*, pages 537–542, Cham, 2020. Springer International Publishing. ISBN 978-3-030-59152-6.

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*, HSCC '19, page 169–178, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362825. doi: 10.1145/3302504.3311806. URL https://doi.org/10.1145/3302504.3311806.

Sydney Katz, Anthony Corso, Sandeep Chinchali, Amine Elhafsi, Apoorva Sharma, Mykel Kochenderfer, and Marco Pavone. Nasa uli aircraft taxi dataset, 2021. URL https://purl.stanford.edu/zz143mb4347.

Laminar Research, 2016. URL https://www.x-plane.com/.

Diego Manzanas Lopez, Sung Woo Choi, Hoang-Dung Tran, and Taylor T. Johnson. Nnv 2.0: The neural network verification tool. In *Computer Aided Verification: 35th International Conference, CAV 2023, Paris, France, July 17–22, 2023, Proceedings, Part II*, pages 397–412, Berlin, Heidelberg, 2023. Springer-Verlag. ISBN 978-3-031-37702-0. doi: 10.1007/978-3-031-37703-7_19. URL https://doi.org/10.1007/978-3-031-37703-7_19.

John Lygeros. On reachability and minimum cost optimal control. *Automatica*, 40(6):917–927, 2004.

Ian Mitchell, Alex Bayen, and Claire J. Tomlin. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control (TAC)*, 50(7):947–957, 2005.

Rahal Nanayakkara, Aaron D Ames, and Paulo Tabuada. Safety under state uncertainty: Robustifying control barrier functions. *arXiv preprint arXiv:2508.17226*, 2025.

Matthew Newton and Antonis Papachristodoulou. Reachability analysis of neural feedback loops using sparse polynomial optimisation. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 2745–2750, 2022. doi: 10.1109/CDC51059.2022.9992719.

Federico Rossi, Cinzia Bernardeschi, and Marco Cococcioni. Neural networks in closed-loop systems: Verification using interval arithmetic and formal prover. *Engineering Applications of Artificial Intelligence*, 137:109238, 2024. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2024.109238. URL https://www.sciencedirect.com/science/article/pii/S0952197624013964.

Christian Schilling, Marcelo Forets, and Sebastián Guadalupe. Verification of neural-network control systems by integrating taylor models and zonotopes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36:8169–8177, 06 2022. doi: 10.1609/aaai.v36i7.20790.

Hoang-Dung Tran, Xiaodong Yang, Diego Manzanas Lopez, Patrick Musau, Luan Viet Nguyen, Weiming Xiang, Stanley Bak, and Taylor T. Johnson. Nnv: The neural network verification tool for deep neural networks and learning-enabled cyber-physical systems. In Shuvendu K. Lahiri and Chao Wang, editors, *Computer Aided Verification*, pages 3–17, Cham, 2020. Springer International Publishing. ISBN 978-3-030-53288-8.

Carlos Trapiello, Christophe Combastel, and Ali Zolghadri. Verification of neural network control systems using symbolic zonotopes and polynotopes, 2023. URL https://arxiv.org/abs/2306.14619.

Han Wang, Zuxun Xiong, Liqun Zhao, and Antonis Papachristodoulou. Model-free verification for neural network controlled systems, 2024. URL https://arxiv.org/abs/2312.08293.

Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and Zico Kolter. Beta-crown: efficient bound propagation with per-neuron split constraints for neural network robustness verification. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Weiming Xiang and Taylor T Johnson. Reachability analysis and safety verification for neural network control systems. *arXiv preprint arXiv:1805.09944*, 2018.

Weiming Xiang and Zhongzhu Shao. Safety verification of neural network control systems using guaranteed neural network model reduction. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 1521–1526, 2022. doi: 10.1109/CDC51059.2022.9992984.

Guoqing Yang, Guangyi Qian, Pan Lv, and Hong Li. Efficient verification of control systems with neural network controllers. In *Proceedings of the 3rd International Conference on Vision, Image*

and Signal Processing, ICVISP 2019, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450376259. doi: 10.1145/3387168.3387244. URL https://doi.org/10.1145/3387168.3387244.

Chi Zhang, Wenjie Ruan, and Peipei Xu. Reachability analysis of neural network control systems. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i12.26783. URL https://doi.org/10.1609/aaai.v37i12.26783.

Yuhao Zhang, Hang Zhang, and Xiangru Xu. Reachability analysis of neural network control systems with tunable accuracy and efficiency. *IEEE Control Systems Letters*, 8:1697–1702, 2024. doi: 10.1109/LCSYS.2024.3415471.

Qingye Zhao, Xin Chen, Zhuoyu Zhao, Yifan Zhang, Enyi Tang, and Xuandong Li. Verifying neural network controlled systems using neural networks. In *Proceedings of the 25th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391962. doi: 10.1145/3501710.3519511. URL https://doi.org/10.1145/3501710.3519511.

Appendix A. Comparison with Set-Based NNCS Verification

To empirically validate the conservatism reduction discussed in Section 2, we compare RoVer-CoRe against NNV 2.0 (Lopez et al., 2023; Tran et al., 2020), a popular set-based neural network control system (NNCS) verification tool. We design a simplified experiment to enable a fair comparison between the two methods.

Setup. Both methods verify a rover following Equation (11) with $v = 1$ m/s and control $u \in [-1, 1]$ rad/s over the state space $[0, 20]$ m $\times$ $[-5, 5]$ m $\times$ $[-\pi, \pi]$ rad with a grid shape of $[100]^3$. A single circular obstacle of radius 1 m is placed at $(p_x, p_y) = (16, 0)$ on the rover's nominal path toward the goal at $(20, 0)$. A pure ReLU MLP controller ($3 \to 128 \to 128 \to 1$) is trained via supervised imitation of an MPC and shared identically by both methods. Perception uncertainty is bounded by $\bar{e} = (\bar{e}_{\hat{p}_x}, \bar{e}_{\hat{p}_y}, \bar{e}_{\hat{\theta}}) = (0.5\text{ m}, 0.5\text{ m}, 0.1\text{ rad})$; the NN observes the perturbed state $\hat{x} = x + e$ with $|e| \leq \bar{e}$, while the dynamics evolve under the true state x.

NNV method. NNV computes the BRT via forward reachability under time-reversed dynamics, which is mathematically equivalent to backward reachability in continuous time under the single-player (fixed controller, worst-case disturbance) setting. The NN controller is analyzed using NNV's approx-star method, which propagates Star sets through the ReLU layers to bound the control output (Tran et al., 2020). The nonlinear plant dynamics are propagated using CORA's zonotope-based reachability. Following NNV's documented pipeline, a per-step interval hull is applied to maintain a single set representation at each time step. The initial obstacle set is partitioned into 100 slices along θ to keep trigonometric intervals tight.

Results. Table 1 compares BRT volumes across time horizons. NNV produces BRTs that are 1.9–3.2× more conservative than RoVer-CoRe's, with the ratio growing as the overapproximation errors from interval-hull set propagation accumulate through the nonlinear dynamics, commonly known as the wrapping effect. Figure 8 visualizes the BRT boundaries at $\theta = 0$ for both methods. RoVer-CoRe's BRTs exhibit smooth boundaries that closely follow the nonlinear dynamics, whereas NNV's BRTs are compositionally rectangular and visibly bloated. The full NNV computation (all time horizons) takes $\approx$ 15 minutes on 24 CPU cores; the RoVer-CoRe pipeline takes $\approx$ 17 s for α, β-CROWN bounds (Wang et al., 2021) and $\approx$ 10 s for the HJ PDE solve (Edward Schmerling) on a single GPU.

Table 1: BRT volume comparison. NNV uses 100 θ-partitions with per-step interval hull.

T (s)	RoVer-CoRe (m^2·rad)	NNV v2.0 (m^2·rad)	Ratio
0.5	27.4	50.9	1.86×
1.0	39.9	94.9	2.38×
1.5	58.1	162.2	2.79×
2.0	82.8	260.9	3.15×
2.5	115.3	365.3	3.17×

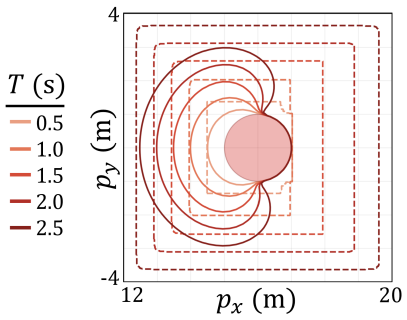

Figure 8: BRT boundaries at $\theta = 0$ for increasing time horizons T (color). RoVer-CoRe (solid) produces smooth, tight boundaries, while NNV v2.0 (dashed) produces compositionally rectangular, bloated boundaries due to the wrapping effect. Circle: obstacle.

Discussion. The conservatism gap reflects a fundamental difference between set-based propagation and grid-based HJ reachability. Set-based methods must represent the reachable set at each time step using a finite geometric object (e.g., a box or zonotope), and each such overapproximation introduces error that compounds through subsequent steps of the nonlinear dynamics, commonly known as the wrapping effect. This explains the growing ratio in Table 1: a small per-step overapproximation accumulates into a large conservatism gap over many steps. HJ reachability, by contrast, solves the value function PDE on a fixed grid without maintaining an explicit set representation, and thus avoids this compounding error entirely. We also tested a tighter NNV configuration that avoids the per-step bounding box by propagating each set element independently; even in this best case, the BRT volume ratio remains 2.04× at $T = 1$ s, though this mode is intractable beyond $T \approx 1$ s due to exponential growth in the number of set elements (Tran et al., 2020). Full experimental details and reproduction scripts are available in the open-source repository[1].