GoTube: Scalable Stochastic Verification of Continuous-Depth Models

Sophie Gruenbacher [1]*, **Mathias Lechner** [2], **Ramin Hasani** [3],
Daniela Rus [3], **Thomas A. Henzinger** [2], **Scott Smolka** [4], **Radu Grosu** [1]

Abstract

We introduce a new stochastic verification algorithm that formally quantifies the behavioural robustness of any time-continuous process formulated as a continuous-depth model. The algorithm solves a set of global optimization (Go) problems over a given time horizon to construct a tight enclosure (Tube) of the set of all process executions starting from a ball of initial states. We call our algorithm GoTube. Through its construction, GoTube ensures that the bounding tube is conservative up to a desired probability. GoTube is implemented in JAX and optimized to scale to complex continuous-depth models. Compared to advanced reachability analysis tools for time-continuous neural networks, GoTube provably does not accumulate over-approximation errors between time steps, and avoids the infamous wrapping effect inherent in symbolic techniques. We show that GoTube substantially outperforms state-of-the-art verification tools in terms of the size of the initial ball, speed, time-horizon, task completion, and scalability, on a large set of experiments. GoTube is stable and sets the state-of-the-art for its ability to scale up to time-horizons well-beyond what has been possible before.

1 Introduction

The use of deep-learning systems powered by continuous-depth models continues to grow, especially due to the revival of neural ordinary differential equations (Neural ODEs) (Chen et al., 2018). Ensuring their safety and robustness is thus becoming a major imperative, particularly in high-stakes decision-making applications, such as medicine, automation, and finance. A particularly appealing approach is to construct a tight over-approximation of the set of states reached over time according to the neural network's dynamics (a bounding tube), and provide deterministic or stochastic guarantees for the conservativeness of the tube's bounds.

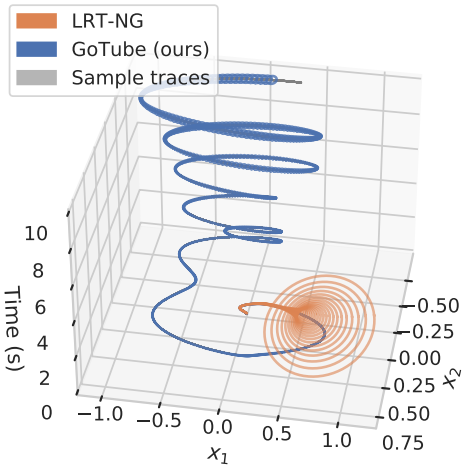

Figure 1: Reachtubes of LRT-NG (Gruenbacher et al., 2020) and GoTube for a CT-RNN controlling CartPole-v1 environment. CAPD (Kapela et al., 2020) and Flow* (Chen et al., 2013a) failed.

Deterministic verification approaches ensure conservative bounds (Chen et al., 2013a; Gowal et al., 2018; Mirman et al., 2018; Bunel et al., 2020a; Kapela et al., 2020; Gruenbacher et al., 2020), but often sacrifice speed and accuracy

*[1]TU Wien, [2]IST Austria, [3]CSAIL MIT, [4]Stony Brook University.
Correspondence to: sophie.gruenbacher@tuwien.ac.at
Code: https://github.com/DatenVorsprung/GoTube

Preprint. Under review.

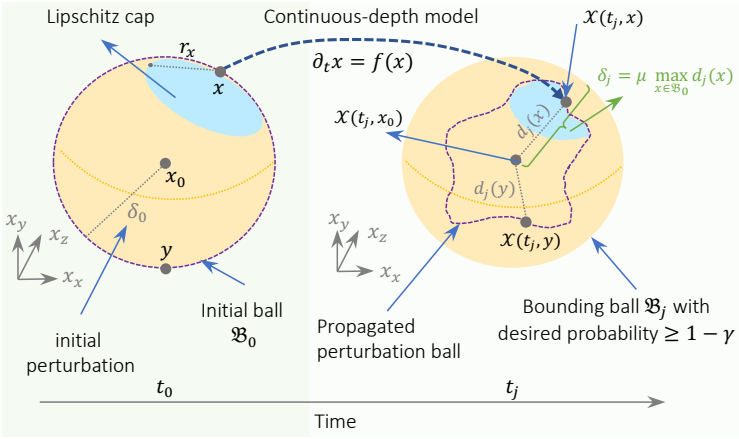

Figure 2: GoTube in a nutshell. The center x_0 of ball $\mathcal{B}_0 = B(x_0, \delta_0)$, with δ_0 the initial perturbation, and samples x drawn uniformly from $\mathcal{B}_0$'s surface, are numerically integrated in time to $\chi(t_j, x_0)$ and $\chi(t_j, x)$, respectively. The Lipschitz constant of $\chi(t_j, x)$ and their distance $d_j(x)$ to $\chi(t_j, x_0)$ are then used to compute Lipschitz caps around samples x, and the radius δ_j of bounding ball $\mathcal{B}_j$. The ratio between the caps' surfaces and $\mathcal{B}_0$'s surface are correlated to the desired confidence $1 - \gamma$.

(Ehlers, 2017), and thus scalability (see CAPD, Flow*, and LRTNG in Fig. 1). Stochastic methods on the other hand, only ensure a weaker notion of conservativeness in form of confidence intervals (stochastic bounds), but this allows them to design much more accurate and faster algorithms that scale up to much larger continuous-depth models (Shmarov and Zuliani, 2015a; Bortolussi and Sanguinetti, 2014; Grunbacher et al., 2021) (see this paper's GoTube in Fig. 1).

It was recently shown that stochastic verification approaches based on Lagrangian reachability can provably guarantee confidence intervals for a given continuous-depth model (Grunbacher et al., 2021). The proposed SLR algorithm, performs both stochastic global optimization and local differential optimization (Zhigljavsky and Zilinskas, 2008; Pontryagin, 2018), to construct in every time step a bounding ball of the reachable states, and thus over time, a tight bounding Tube.

Interval arithmetic is used in this process to symbolically bound the magnitude of each ball's Lipschitz constant, and find reasonably tight spherical Lipschitz (don't care) caps around the initial-state samples taken from the initial ball. The ratio between the surface covered by all Lipschitz caps and the surface of the initial ball is correlated to the desired probability. The radius of the initial ball can be understood as quantifying the magnitude of a perturbation of its center.

Although this theoretical result suggested an elegant way to avoid compounding errors, and to relax the computational overhead of multiple forward-propagations (standard in deterministic approaches (De Palma et al., 2021)), its practical implementation faces three fundamental problems on its way towards scalability: 1) The efficient sampling and propagation of tens of thousands of initial states. 2) The use of local gradient descent to search for local maxima. 3) The use of interval arithmetic to compute a conservative upper bound for the Lipschitz constant.

Each of these problems renders a naive implementation intractable. While the implications of Problem 1 are obvious, those for Problems 2-3 are more insidious. As one propagates all initial-state samples in time according to the neural-network's dynamics, the length of gradient descent through backpropagation in Problem 2 increases with each time step. This leads to increasingly longer computation times and to vanishing gradients that have to be curated.

Problem 3 is even more perfidious. The longer one propagates the initial states in time, the longer it takes to compute the Lipschitz constant through interval arithmetic, and the more conservative its approximation becomes due to wrapping effects. This in turn leads to increasingly smaller Lipschitz caps. Ensuring the desired stochastic bounds thus requires increasingly more samples.

Motivated by these theoretical and practical problems, we introduce in this paper **GoTube**, an algorithm and its associated tool for solving these problems in an elegant and nontrivial fashion. On a large set of experiments with continuous-depth models, GoTube substantially outperforms

state-of-the-art verification tools in terms of the size of the initial ball, speed, time-horizon, task completion, and scalability. In particular, the main contributions of this paper are as follows:

- *We solve Problem 1* by sampling and propagating the initial states in parallel, according to the neural network's dynamics. To this end we carefully and completely tensorized the naive implementation, which allowed us to take advantage of state-of-the-art ML tools such as JAX. As a result, we were able to easily work with tens of thousands of samples.
- *We solve Problem 2* by getting rid of local differential optimization altogether. Once we were able to run very large numbers of samples in parallel, the advantage of gradient-based search was compensated for by using additional samples. This dramatically sped up computation time, as it removed the propagation-horizon dependence. Hence the name GoTube.
- *We solve Problem 3* by replacing the conservative interval-based computation of the Lipschitz constant, with a statistical estimation of its value, based on the propagated samples. This eliminated the dependence on the propagation horizon and the blow-up in the number of samples. However, this required a new theory, ensuring desired stochastic bounds.

Compared to existing reachability analysis tools, GoTube does not accumulate over-approximation errors between time steps; rather it bounds these errors by a user-defined parameter μ. Moreover, GoTube avoids over-approximation errors of the Lipschitz constant inherent in symbolic techniques by using mean-value statistics, with confidence $1 - \lambda$, for the conservativeness bounds.

We perform a variety of experiments with GoTube on continuous-time neural networks and benchmark it against state-of-the-art baselines. We find that GoTube provides stochastic bounds up to arbitrary time-horizons, a capability that was not possible to achieve with the other reachability-analysis methods. We also observe that GoTube can use significantly larger perturbation radius for its initial ball, compared to other contemporary verification methods. The volume of the computed bounding tubes is also significantly tighter for continuous-depth models involving neural networks then the one of other tools, a reason of why they do not blow up.

2 Related Work

Global Optimization. Efficient local optimization methods such as gradient descent cannot be used for global optimization since such problems are typically non-convex. Thus, many advanced verification algorithms tend to use global optimization schemes (Bunel et al., 2018, 2020a). Depending on the properties of the objective function such as smoothness, various types of global optimization techniques exist. For instance, interval-based branch-and-bound (BaB) algorithms (Neumaier, 2004; Hansen and Walster, 2003) work well on differentiable objectives up to a certain scale, which has recently been improved (De Palma et al., 2021). There are also Lipschitz-global optimization methods for satisfying Lipschitz conditions (Piyavskii, 1972; Shubert, 1972; Malherbe and Vayatis, 2017; Kvasov and Sergeyev, 2013). For example, a method for computing the Lipschitz constant of deep neural networks to assist with their robustness and verification analyses was recently proposed in (Fazlyab et al., 2019) and (Bhowmick et al., 2021). Additionally, there are evolutionary strategies for global optimization using the covariance matrix computation (Hansen and Ostermeier, 2001; Igel et al., 2007). In our approach, for global optimization, we use random sampling and compute neighbourhoods (Lipschitz caps) of the samples, where we have probabilistic knowledge about the values, such that we are able to correspondingly estimate the stochastic global optimum with high confidence. (Zhigljavsky and Zilinskas, 2008).

Verification of Neural Networks. A large body of work tried to enhance the robustness of neural networks against adversarial examples (Goodfellow et al., 2014; Szegedy et al., 2013). There are efforts that show how to break the many defense mechanisms proposed (Athalye et al., 2018; Uesato et al., 2018; Lechner et al., 2020b, 2021; Babaiee et al., 2021), until the arrival of methods for formally verifying robustness to adversarial attacks around neighborhoods of data (Tjeng et al., 2018; Wong and Kolter, 2018; Henzinger et al., 2021). The majority of these complete verification algorithms for neural networks work on piece-wise linear structures of small- to-medium-size feedforward networks (Salman et al., 2019). For instance, (Bunel et al., 2020b) has recently introduced a BaB method that outperforms state-of-the-art verification methods (Katz et al., 2017; Wang et al., 2018; Tjeng et al., 2018; Tjandraatmadja et al., 2020). A more scalable approach for rectified linear unit (ReLU) networks (Nair and Hinton, 2010) was recently proposed based on Lagrangian decomposition; this

approach significantly improves the speed and tightness of the bounds (De Palma et al., 2021). The proposed approach not only improves the tightness of the bounds but also performs a novel branching that matches the performance of the learning-based methods (Lu and Mudigonda, 2020) and outperforms state-of-the-art methods (Zhang et al., 2018; Singh et al., 2020; Bak et al., 2020; Henriksen and Lomuscio, 2020). While these verification approaches work well for feed-forward networks with growing complexity, they are not suitable for recurrent and continuous neural network instances (Hasani, 2020; Hasani et al., 2017a; Lechner and Hasani, 2020), which we address in this work.

Verification of Continuous-time Systems. Reachabilty analysis is a verification approach that provides safety guarantees for a given continuous dynamical system (Henzinger and Rusu, 1998; Alur et al., 2000; Islam et al., 2016; Ruan et al., 2018; Gurung et al., 2019; Vinod and Oishi, 2021). Most dynamical systems in safety-critical applications (Hasani et al., 2016; Wang et al., 2017; Hasani et al., 2017b; Wang et al., 2019; Hasani et al., 2019b; Brunnbauer et al., 2021) are highly nonlinear and uncertain in nature (Hasani et al., 2019a; DelPreto et al., 2018; Lechner et al., 2020a). The uncertainty can be in the system's parameters (Wang et al., 2015; Fränzle et al., 2010; Shmarov and Zuliani, 2015a; Enszer and Stadtherr, 2011), or their initial state (Enszer and Stadtherr, 2011; Huang et al., 2017). This is often handled by considering balls of a certain radius around them. Nonlinearity might be inherent in the system dynamics, or due to discrete mode-jumps (Fränzle et al., 2011), We provide a summary of methods developed for the reachability analysis of continuous-time ODEs in Table 1.

A fundamental shortcoming of the majority of the methods described in Table 1 is their lack of scalability while providing conservative bounds. Even approaches such as Stochastic Lagrangian Reachability (SLR) (Grunbacher et al., 2021) which guarantees convergence of their verification algorithm on large-scale systems, in practice fail to complete the verification of ODE-based neural networks given a larger time horizon. In this paper, we show that GoTube establishes the state-of-the art for the verification of ODE-based systems in terms of speed, time-horizon, task completion, and scalability on a large set of experiments.

Table 1: Related work on the reachability analysis of continuous-time systems. Determ.= Deterministic. "No" indicates a stochastic method. Table content is partially reproduced from Grunbacher et al. (2021).

Technique	**Determ.**	**Parallel**	**wrapping effect**	**Arbitrary Time-horizon**
LRT (Cyranka et al., 2017) with Infinitesimal strain theory	yes	no	yes	no
CAPD (Kapela et al., 2020) implements Lohner algorithm	yes	no	yes	no
Flow-star (Chen et al., 2013b) with Taylor models	yes	no	yes	no
δ-reachability (Gao et al., 2013) with approximate satisfiability	yes	no	yes	no
C2E2 (Duggirala et al., 2015) with discrepancy functions	yes	no	yes	no
LDFM (Fan et al., 2017) by simulation, matrix measures	yes	yes	no	no
TIRA (Meyer et al., 2019) with second-order sensitivity	yes	yes	no	no
Isabelle/HOL (Immler, 2015) with proof-assistant	yes	no	yes	no
Breach (Donzé, 2010; Donzé and Maler, 2007) by simulation	yes	yes	no	no
PIRK (Devonport et al., 2020) with contraction bounds	yes	yes	no	no
HR (Li et al., 2020a) with hybridization	yes	no	yes	no
ProbReach (Shmarov and Zuliani, 2015b) with δ-reachability,	no	no	yes	no
VSPODE (Enszer and Stadtherr, 2011) using p-boxes	no	no	yes	no
Gaussian process (GP) (Bortolussi and Sanguinetti, 2014)	no	no	no	no
Stochastic Lagrangian reachability SLR (Grunbacher et al., 2021)	no	yes	no	no
GoTube (Ours)	no	yes	no	**yes**

3 Setup

In this section, we introduce our notation, preliminary concepts, and definitions required to state and prove the stochastic bounds that GoTube guarantees for time-continuous process models.

Continuous-depth models. These are a special case of nonlinear ordinary differential equations (ODEs), where the model is defined by the derivative of the unknown states x computed by a vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^n$, which is assumed to be Lipschitz-continuous and forward-complete:

$$\partial_t x = f(x), \quad x(t_0) \in \mathcal{B}_0 = B(x_0, \delta_0), \tag{1}$$

where $\mathcal{B}_0$ defines the initial ball (a region of initial states, whose radius quantifies the magnitude δ_0 of a perturbation of its center x_0) equivalent to (Grunbacher et al., 2021). Time dependence can be incorporated by an additional variable x with $\delta_t x = 1$. Thus this definition naturally extends to time-varying ODEs. Nonlinear ODEs do not have in general closed-form solutions, and therefore one can not compute symbolically the solution $\chi(t_j, x)$ for all $x \in \mathcal{B}_0$. For a sequence of k timesteps from time t_0 until time horizon T: $t_0 < \ldots < t_k = T$, we use numerical ODE solvers to compute $\chi(t_j, x)$ of the initial value problem (IVP) in Eq. (1) at time t_j starting at different points $x(t_0) = x$.

We extend this computation to the entire ball, by numerically integrating the center x_0 and a set of points $x \in \mathcal{V}$, uniformly sampled from the surface of the ball, and using this information to compute stochastic upper bounds for the propagated perturbation δ_0 of the center x_0 at every time t_j. Building up on the setup definitions in (Grunbacher et al., 2021), we define the following:

Definition 1 (Bounding ball) *Given an initial ball $\mathcal{B}_0 = B(x_0, \delta_0)$, we call $B(\chi(t_j, x_0), \delta_j(\mathcal{B}_0))$ a* bounding ball *at time t_j, if it stochastically bounds the reachable states x at time t_j for all initial points around x_0 having the maximal initial perturbation δ_0.*

According to the notation at (Grunbacher et al., 2021), we refer to the bounding ball at time t_j simply as $\mathcal{B}_j = B(x_j, \delta_j)$, whenever the initial values x_0 and δ_0 are known from the context.

Definition 2 (Bounding Tube) *Given an initial ball $\mathcal{B}_0 = B(x_0, \delta_0)$ and bounding balls for $t_0 < \ldots < t_k = T$, we call the series of bounding balls $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_k$ a* bounding tube.

Maximum perturbation at time t_j. To compute a bounding tube, we have to compute at every timestep t_j the maximum perturbation δ_j, which is defined as a solution of the optimization problem:

$$\delta_j \geq \max_{x \in \mathcal{B}_0} \|\chi(t_j, x) - \chi(t_j, x_0)\| = \max_{x \in \mathcal{B}_0} d(t_j, x), \tag{2}$$

where $d_j(x) = d(t_j, x)$ denotes the *distance* at time t_j, if the initial center x_0 is known from the context. As stated in (Grunbacher et al., 2021), the radius at time t_j can be over-approximated by solving a global optimization problem on the surface of the initial ball $\mathcal{B}_0^S$: as we require Lipschitz-continuity and forward-completeness of the ODE in Eq. (1), the map $x \mapsto \chi(t_j, x)$ is a homeomorphism and commutes with closure and interior operators. In particular, the image of the boundary of the set $\mathcal{B}_0$ is equal to the boundary of the image $\chi(t_j, \mathcal{B}_0)$. Thus, Eq. (2) has its optimum on the surface of the initial ball $\mathcal{B}_0^S = \text{surface}(\mathcal{B}_0)$, and we will only consider points on the surface.

Definition 3 (Lipschitz cap) *Let $\mathcal{V}$ be the set of all sampled points, $x \in \mathcal{V}$ be a sample point on the surface of the initial ball, $\bar{m}_{j,\mathcal{V}} = \max_{x \in \mathcal{V}} d_j(x)$ be the sample maximum and $B(x, r_x)^S = B(x, r_x) \cap \mathcal{B}_0^S$ be a spherical cap around that point. We call the cap $B(x, r_x)^S$ a γ, t_j-Lipschitz cap, if it holds that $\Pr\left(d_j(y) \leq \mu \cdot \bar{m}_{j,\mathcal{V}}\right) \geq 1 - \gamma$ for all $y \in B(x, r_x)^S$.*

Lipschitz caps around the samples enable us to calculate a probability of having found an upper bound of the true maximum $m_j^\star = d_j(x_j^\star) = \max_{x \in \mathcal{B}_0} d_j(x)$ of the optimization problem in Eq. (2), as it follows from the definition that if the Lipschitz cap of a sample $x \in \mathcal{B}_0^S$ covers $x_j^\star$, then it follows that $\Pr(m_j^\star \leq \mu \cdot d_j(x)) \geq 1 - \gamma$. Intuitively, the points within a cap do not have to be explored.

4 Main Results

Our GoTube algorithm and its theory is inspired by the stochastic lagrangian reachability (SLR) algorithm (Grunbacher et al., 2021). However, it solves SLR's fundamental scalability problems.

On the one hand, we replaced gradient descent by implementing tensorization and substantially increasing the number of random samples used. On the other hand, we gave up interval arithmetic for computing a conservative upper bound for the Lipschitz constant, thus replacing deterministic caps with stochastic Lipschitz caps. To be able to do that, we formulated Theorems on: 1) How to choose the radius of a Lipschitz cap using the local Lipschitz constants of the samples together with the expected difference quotients. 2) Convergence guarantees using these new stochastic caps, as they are used by GoTube to compute the probability of δ_j being an upper bound of the biggest perturbation.

We start by describing the GoTube Algorithm. This facilitates the comprehension of the different computation and theory steps. Given a continuous-depth model as in Eq. (1), an initial ball $\mathcal{B}_0$ defined

Algorithm 1 GoTube

Require: initial ball $\mathcal{B}_0 = B(x_0, \delta_0)$, time horizon T, sequence of timesteps t_j ($t_0 < \cdots < t_k = T$), tolerance $\mu > 1$, confidence level $\gamma \in (0,1)$, batch size b, distance function d

$\mathcal{V} \leftarrow \{\}$ (list of visited random points)
sample batch $x^B \in \mathcal{B}_0^S$
for $(j = 1; j \leq k; j = j + 1)$ **do**
  $\bar{p} \leftarrow 0$
  while $\bar{p} < 1 - \gamma$ **do**
    $\mathcal{V} \leftarrow \mathcal{V} \cup \{x^B\}$
    $x_j \leftarrow \chi(t_j, x_0)$ (integrate initial center point)
    $\bar{m}_{j,\mathcal{V}} \leftarrow \max_{x \in \mathcal{V}} d(t_j, x)$
    compute local Lipschitz constants λ_x for $x \in \mathcal{V}$
    compute expected local difference quotient $\Delta\lambda_{x,\mathcal{V}}$ for $x \in \mathcal{V}$
    compute cap radii $r_x(\lambda_x, \Delta\lambda_{x,\mathcal{V}})$ (Thm. 1) for $x \in \mathcal{V}$
    $\mathcal{S} \leftarrow \bigcup_{x \in \mathcal{V}} B(x, r_x)^S$ (total covered area)
    $\bar{p} \leftarrow \Pr(\mu \cdot \bar{m}_{j,\mathcal{V}} \geq m^\star)$
    sample batch $x^B \in \mathcal{B}_0$
  end while
  $\delta_j \leftarrow \mu \cdot \bar{m}_{j,\mathcal{V}}$
  $\mathcal{B}_j \leftarrow B(x_j, \delta_j)$
end for
return $(\mathcal{B}_1, \ldots, \mathcal{B}_k)$

by a center point x_0 and the maximum initial perturbation δ_0, a time horizon T with a sequence of timesteps t_j ($t_0 < \ldots < t_k = T$), a confidence level $\gamma \in (0,1)$, a maximum multiplicative tolerance of over-approximation $\mu > 1$, a batch size b, and a distance function d. The output of the GoTube algorithm is a bounding tube that stochastically over-approximates at most by μ the propagated initial perturbation from the center x_0 with a probability higher than $1 - \gamma$. Although the input and output is similar to (Grunbacher et al., 2021), we had to significantly change the algorithm by creating new theorems, such that GoTube is scalable and works also on continuous-depth models.

GoTube starts by sampling a batch (tensor) $x^B \in \mathcal{B}_0^S$. It then iterates for the k steps of the time horizon T the following. After initializing the probability ensured to zero, and the visited states to the empty set, it loops until it reaches the desired confidence (probability) $1 - \gamma$, by increasingly taking additional batches. In each iteration, it integrates the center and the already available samples from their previous time step, and the possibly new batches from their initial state (for simplicity, the pseudocode does not make this distinction explicit). GoTube then computes the maximum distance from the integrated samples to the integrated center, their local Lipschitz constant, as in (Grunbacher et al., 2021), according to the variational equation of Eq. (1). Based on this information GoTube then computes the mean Lipschitz statistics and the cap radii accordingly. The total surface of the caps is then employed to compute and update the achieved confidence (probability). Once the desired confidence is achieved, GoTube exits the inner loop, and computes the bounding ball in terms of its center and radius. After exiting the outer loop GoTube returns the bounding tube.

Theorem 1 (Radius of Lipschitz Caps) *Given a continuous-depth model f from Eq. (1), $\gamma \in (0,1)$, $\mu > 1$, target time t_j, the set of all sampled points $\mathcal{V}$, the number of sampled points $N = |\mathcal{V}|$, the sample maximum $\bar{m}_{j,\mathcal{V}} = \max_{x \in \mathcal{V}} d_j(x)$, the IVP solutions $\chi(t_j, x)$, and the corresponding stretching factors $\lambda_x = \|\partial_x \chi(t_j, x)\|$ for all $x \in \mathcal{V}$. For $x \in \mathcal{V}$, let $\nu_x = |\lambda_x - \lambda_X| / \|x - X\|$ be a new random variable, where $X \in \mathcal{B}_0^S$ is the random variable which is thrown by random sampling on the surface of the initial ball. Let the upper bound $\Delta\lambda_{x,\mathcal{V}}$ of the confidence interval of $\mathbb{E}\nu_x$ be defined as follows:*

$$\Delta\lambda_{x,\mathcal{V}}(\gamma) = \overline{\nu_x} + t^*_{\gamma/2}(N-2)\frac{s(\nu_x)}{\sqrt{N-1}}, \tag{3}$$

with $\overline{\nu_x}$ and $s(\nu_x)$ being the sample mean and sample standard deviation of ν_x, and t^* being the Student's t-distribution. Let r_x be defined as:

$$r_x = (2 \cdot \Delta\lambda_{x,\mathcal{V}})^{-1} \left(-\lambda_x + \sqrt{\lambda_x^2 + 4 \cdot \Delta\lambda_{x,\mathcal{V}} \cdot (\mu \cdot \bar{m}_{j,\mathcal{V}} - d_j(x))} \right), \tag{4}$$

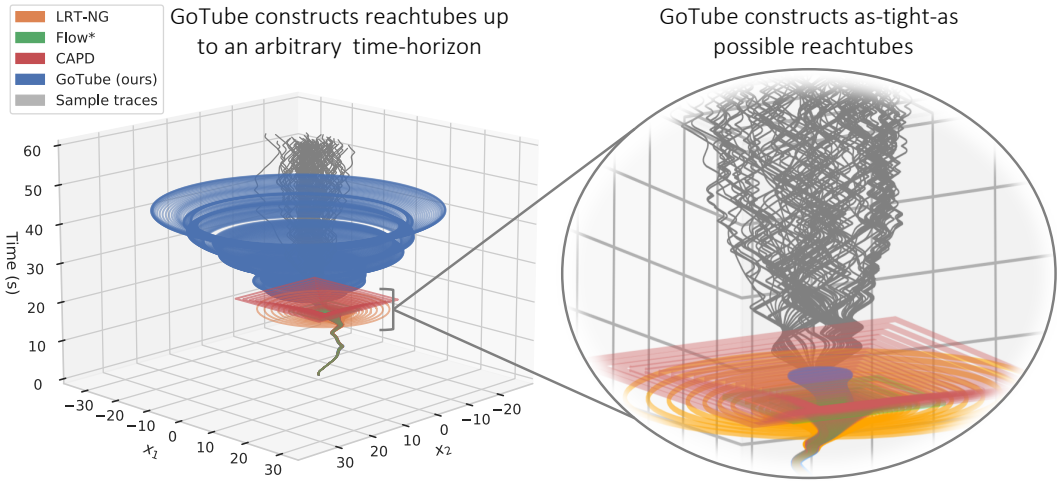

Figure 3: Visualization of the reachtubes constructed for the Dubin's car model with various reachability methods. While the tubes computed by existing methods (LRT-NG, Flow* and CAPD) explode at $t \approx 20s$ due to the accumulation of over-approximation errors (the infamous wrapping effect), GoTube can keep tight bounds beyond $t > 40s$. Note also the chaotic nature of 100 executions.

then it holds that:

$$\Pr\left(d_j(y) \leq \mu \cdot \bar{m}_{j,\mathcal{V}}\right) \geq 1 - \gamma \quad \forall y \in B(x, r_x)^S, \tag{5}$$

and thus that $B(x, r_x)^S$ is a γ, t_j-Lipschitz cap.

The full proof is provided in the Appendix. *Proof sketch:* As $\Delta\lambda_{x,\mathcal{V}}$ is the upper bound of the confidence interval of $\mathbb{E}\nu_x$, it holds that $\Pr(\lambda_y \leq \lambda_x + \Delta\lambda_{x,\mathcal{V}} \cdot \|x - y\|) \geq 1 - \gamma$. Therefore Eq. (4) follows by solving the following equation: $(\mu \cdot \bar{m}_{j,\mathcal{V}} - d_j(x)) = \lambda_x r_x + \Delta\lambda_{x,\mathcal{V}} r_x^2$ and using a similar proof to the one of (Grunbacher et al., 2021)[Theorem 1].

We now state that the convergence guarantee holds for the GoTube Algorithm, to ensure that the Algorithm terminates in finite time.

Theorem 2 (Convergence Guarantee using Lipschitz caps) *Given the over-approximation factor $\mu > 1$, the set of all sampled points $\mathcal{V}$ and the sample maximum $\bar{m}_{j,\mathcal{V}} = \max_{x \in \mathcal{V}} d_j(x)$. Let $m_j^\star = \max_{x \in \mathcal{B}_0} d_j(x)$ be the initial ball maximum. Then:*

$$\forall \gamma \in (0,1), \exists N \in \mathbb{N} \text{ s.t. } \Pr(\mu \cdot \bar{m}_{j,\mathcal{V}} \geq m_j^\star) \geq 1 - \gamma, \tag{6}$$

where $N = |\mathcal{V}|$ is the number of sampled points.

The full proof is provided in the Appendix. *Proof sketch:* Let $x_j^\star$ be a point such that $d_j(x_j^\star) = m_j^\star$. Given $\gamma \in (0,1)$ and cap radii r_x, we know from the proof of (Grunbacher et al., 2021)[Theorem 2] that $\exists N \in \mathbb{N}$: $\Pr(\exists x \in \mathcal{V}\colon B(x, r_x)^S \ni x_j^\star) \geq \sqrt{1-\gamma}$. Using a set of sampled points $\mathcal{V}$ with cardinality N and using $1 - \sqrt{1-\gamma}$ instead of γ in Eq. (3) and Theorem 1, the resulting probability is larger than $\sqrt{1-\gamma}$. From the definition of a Lipschitz cap it follows that $\Pr(d_j(x^\star) \leq \mu \cdot \bar{m}_{j,\mathcal{V}} | \exists x \in \mathcal{V}\colon B(x, r_x)^S \ni x^\star) \geq \sqrt{1-\gamma}$. For any sets A, B it holds that $\Pr(A) \geq \Pr(A \cap B) = \Pr(A|B) \cdot \Pr(B)$, thus we multiply both probabilities and therefore Eq. (6) holds.

5 Experimental Evaluation

We perform a diverse set of experiments with GoTube to evaluate its performance and identify its characteristics and limits in verifying continuous-time systems with increasing complexity. We run our evaluations on a standard workstation machine setup (12 vCPUs, 64GB memory) equipped with single GPU, for a per-run timeout of 1 hour.

Table 2: Comparison of GoTube to existing reachability methods. Benchmark models and results of other methods from (Gruenbacher et al., 2020). The first six benchmarks concern classical dynamical systems, whereas the two bottom rows correspond to time-continuous RNN models (LTC= liquid time-constant networks) in a closed feedback loop with an RL environment (Lechner et al., 2019; Hasani et al., 2020, 2021b; Vorbach et al., 2021). The numbers show the volume of the constructed tube. Lower is better, best number in bold.

Benchmark	LRT-NG	Flow*	CAPD	LRT	GoTube (50%)	GoTube (90%)	GoTube (99%)
Brusselator	1.5e-4	9.8e-5	3.6e-4	6.1e-4	8.6e-5	8.6e-5	**8.6e-5**
Van Der Pol	4.2e-4	**3.5e-4**	1.5e-3	3.7e-3	5.0e-4	5.0e-4	5.0e-4
Robotarm	**7.9e-11**	8.7e-10	1.1e-9	Fail	2.5e-10	2.5e-10	2.5e-10
Dubins Car	0.131	4.5e-2	0.1181	385	1.5e-2	2.5e-2	**2.6e-2**
Cardiac-cell	**3.7e-9**	1.5e-8	4.4e-8	3.2e-8	4.2e-8	4.2e-8	4.5e-8
CartPole-v1+Linear	**7.2e-17**	7e-13	2.6e-13	Fail	3.2e-8	3.2e-8	3.2e-8
CartPole-v1+LTC	4.49e-33	Fail	Fail	Fail	1.3e-37	2.6e-37	**4.9e-37**
CartPole-v1+CTRNN	3.9e-27	Fail	Fail	Fail	5.4e-34	9.9e-34	**1.2e-33**

5.1 On the volume of the bounding balls with GoTube

Our first experimental evaluation concerns the over-approximation errors of the constructed bounding tubes. An ideal reachability tool should be able to output an as tight as possible tube that encloses the system's executions. Consequently, as our comparison metric we will report the average volume of the bounding balls, with less volume is better. We use the benchmarks and settings of (Gruenbacher et al., 2020) (same radii, time horizons, and models) as basis of our evaluation. In particular, we compare GoTube to the deterministic, state-of-the-art reachablity tools LRT-NG, Flow*, CAPD and LRT. We measure the volume of GoTube's balls at the confidence levels of 50%, 90% and 99%.

The results are shown in Table 2. For the first six benchmarks, which are classical dynamical systems, we use the small time horizons T and small initial radii δ_0, which the other tools could handle. GoTube with a 99% confidence achieves a competitive performance to the other tools, outperforming them in 2 out of 6 benchmarks. The specific reachtubes and the chaotic nature of hundred executions of the Dubin's car are shown in Figure 3. As one can see, the GoTube reachtube extends to a much longer time horizon, which we fixed at 40s. All other tools blew up before 20s. For the two problems involving neural networks, GoTube produces significantly tighter reachtubes.

5.2 GoTube provides safety bounds up an arbitrary time horizon

In our second experiment, we evaluate for how long GoTube and existing methods can construct a reachtube before exploding due to over-approxmiation errors. To do so, we extend the benchmark setup of (Gruenbacher et al., 2020) by increasing the time horizon for which the tube should be constructed and set GoTube to a 95% confidence level, that is, probability of being conservative.

Table 3: Results of the extended benchmark of (Gruenbacher et al., 2020) by longer time horizons. The numbers show the volume of the constructed tube, "Blowup" indicate that the method produced `Inf` or `NaN` values due to a blowup. Lower is better, best method shown in bold.

Benchmark	CartPole-v1+CTRNN		CartPole-v1+LTC	
Time horizon	1s	10s	0.35s	10s
LRT	Blowup	Blowup	Blowup	Blowup
CAPD	Blowup	Blowup	Blowup	Blowup
Flow*	Blowup	Blowup	Blowup	Blowup
LRT-NG	3.9e-27	Blowup	4.5e-33	Blowup
GoTube (ours)	**8.8e-34**	**1.1e-19**	**4.9e-37**	**8.7e-21**

The results in Table 3 demonstrate that GoTube produces significantly longer reachtubes than all considered state-of-the-art approaches, without suffering from severe over-approximation errors.

Table 4: Volume of the reachtube constructed by GoTube for our three newly proposed benchmarks. The volume stays small for increasing radius of the set of initial states $r = 0.01$.

Benchmark	r	$5r$	$10r$
LDS + CT-RNN	7.0e-11	5.9e-4	0.702
Inverted Pendulum + CT-RNN	1.4e-12	1.5e-05	0.0178
Oscillatory CT-RNN	4.7e-37	7.8e-26	4.8e-21

Particularly, Figure 1 visualizes the difference to the existing methods and over-approximation margins for two example dimensions of the CartPole-v1 environment and its CT-RNN controller.

5.3 GoTube can use significantly larger perturbation radius for its initial ball

In our last experiment we introduce a new set of benchmark models, entirely based on continuous-time recurrent neural networks. The first model is an unstable linear dynamical system of the form $\dot{x} = Ax + Bu$ that is stabilized by a CT-RNN policy via actions u. The second model corresponds to the inverted pendulum environment, which is similar to the CartPole environment but differs by that the control actions are applied via a torque vector on the pendulum directly instead of moving a cart. The CT-RNN policies for these two environments were trained using deep RL. Our third new benchmark model concerns the analysis of the learned dynamics of a CT-RNN trained on supervised data. In particular, we want to asses using reachability frameworks if the learned network expressed oscillatory behavior. The CT-RNN state vector consists of 16 dimensions, which is twice as much as existing CT-RNN reachability benchmarks (Gruenbacher et al., 2020). We vary the radius of the set of initial states $r \in \{0.01, 0.05, 0.1\}$ to evaluate whether GoTube can handle larger initial sets.

The results for a time horizon of 10s in the first two examples, and of 2s in the last example, are shown in Table 4. They demonstrate that GoTube can scale to different initial conditions, that is, perturbation magnitudes, and set a new benchmark for future methods to compare with.

6 Discussions, Scope and Conclusions

We proposed a new stochastic verification algorithm (GoTube) that scales up to providing robustness guarantees (also safety guarantees if a set of states to be avoided is given) for complex time-continuous systems. GoTube is stable and sets the state-of-the-art for its ability to scale up to time-horizons well-beyond what has been possible before. The algorithm moreover allows larger perturbation radius for the initial ball for which other models fail. Lastly, GoTube scales up to the verification of advanced continuous-depth neural models where state-of-the-art deterministic approaches fail.

Stochastic Lagrangian Reachability (SLR) vs. GoTube? SLR (Grunbacher et al., 2021) is a theoretical stochastic-reachability framework quantifying the robustness of continuous-depth models, in particular of neural ODEs. Using our code base we implemented SLR, as no implementation was available yet, and observed that although it does not blow up in space, it blows up in time such that we were not able to construct reachtubes for our high-dimensional benchmarks.

What about Gaussian Processes as a tool for stochastic verification? Gaussian Processes (GP) are powerful stochastic models which can be used for stochastic reachability analysis (Bortolussi and Sanguinetti, 2014) and uncertainty estimation for stochastic dynamical systems (Gal, 2016). The major shortcoming of GPs is that they simply cannot scale to the complex continuous-time systems that we tested here. Moreover, Gaussian Processes have a large number of hyperparameters to set which can be challenging to tune across different benchmarks.

Limitations of GoTube. GoTube does not necessarily perform better in terms of average volume of the bounding balls for smaller tasks and short time horizons, as shown in Table 2. GoTube is not yet suitable for the verification of stochastic dynamical systems for instance Neural Stochastic Differential Equations (Neural SDEs) (Li et al., 2020b; Xu et al., 2021). Although GoTube is considerably more computationally efficient than existing methods, the dimensionality of the system-under-test as well as the type of numerical ODE solver exponentially affect their performance. We can improve on this limitation by using Hypersolvers (Poli et al., 2020), closed-form continuous depth models (Hasani et al., 2021a), and compressed representations of neural ODEs (Liebenwein et al., 2021).

Future of GoTube. GoTube opens many avenues for future research. The most straightforward next step is to search for better intermediate steps in Algorithm 1. For instance better ways to compute the Lipschitz constant and improving the sampling process. GoTube is now applicable for complex deterministic ODE systems; it would be an important line of work to find ways to mary reachability analysis with machine learning approaches to verify neural SDEs as well. Last but not least, we believe that there is a close relationship between stochastic reachability analysis and uncertainty estimation techniques used for deep learning models (Abdar et al., 2021). Approaches such as Evidential Regression (Amini et al., 2019) provide stochastic bounds over the uncertainty of large-scale machine learning models, very similar to the objective of our GoTube algorithm. Uncertainty-aware verification could be worth to explore based on what we learned with GoTube.

Acknowledgments and Disclosure of Funding

SG is funded by the Austrian Science Fund (FWF) project number W1255-N23. ML and TH are supported in part by FWF under grant Z211-N23 (Wittgenstein Award). SS is supported by NSF awards DCL-2040599, CCF-1918225, and CPS-1446832. RH and DR are partially supported by Boeing. RG is partially supported by Horizon-2020 ECSEL Project grant No. 783163 (iDev40).

References

Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 2021.

Rajeev Alur, Radu Grosu, and Michael McDougall. Efficient reachability analysis of hierarchical reactive machines. In *International Conference on Computer Aided Verification*, pages 280–295. Springer, 2000.

Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. *arXiv preprint arXiv:1910.02600*, 2019.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pages 274–283. PMLR, 2018.

Zahra Babaiee, Ramin Hasani, Mathias Lechner, Daniela Rus, and Radu Grosu. On-off center-surround receptive fields for accurate and robust image classification. In *International Conference on Machine Learning*, pages 478–489. PMLR, 2021.

Stanley Bak, Hoang-Dung Tran, Kerianne Hobbs, and Taylor T Johnson. Improved geometric path enumeration for verifying relu neural networks. In *International Conference on Computer Aided Verification*, pages 66–96. Springer, 2020.

Aritra Bhowmick, Meenakshi D'Souza, and G Srinivasa Raghavan. Lipbab: Computing exact lipschitz constant of relu networks. *arXiv preprint arXiv:2105.05495*, 2021.

Luca Bortolussi and Guido Sanguinetti. A statistical approach for computing reachability of non-linear and stochastic dynamical systems. In Gethin Norman and William Sanders, editors, *Quantitative Evaluation of Systems*, pages 41–56, Cham, 2014. Springer International Publishing.

Axel Brunnbauer, Luigi Berducci, Andreas Brandstätter, Mathias Lechner, Ramin Hasani, Daniela Rus, and Radu Grosu. Model-based versus model-free deep reinforcement learning for autonomous racing cars. *arXiv preprint arXiv:2103.04909*, 2021.

Rudy Bunel, Alessandro De Palma, Alban Desmaison, Krishnamurthy Dvijotham, Pushmeet Kohli, Philip Torr, and M Pawan Kumar. Lagrangian decomposition for neural network verification. In *Conference on Uncertainty in Artificial Intelligence*, pages 370–379. PMLR, 2020a.

Rudy Bunel, P Mudigonda, Ilker Turkaslan, P Torr, Jingyue Lu, and Pushmeet Kohli. Branch and bound for piecewise linear neural network verification. *Journal of Machine Learning Research*, 21 (2020), 2020b.

Rudy R Bunel, Ilker Turkaslan, Philip Torr, Pushmeet Kohli, and Pawan K Mudigonda. A unified view of piecewise linear neural network verification. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/be53d253d6bc3258a8160556dda3e9b2-Paper.pdf.

Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 6571–6583. Curran Associates, Inc., 2018.

Xin Chen, Erika Ábrahám, and Sriram Sankaranarayanan. Flow*: an analyzer for non-linear hybrid systems. In *CAV*, pages 258–263, 2013a.

Xin Chen, Erika Ábrahám, and Sriram Sankaranarayanan. Flow*: an analyzer for non-linear hybrid systems. In *CAV*, pages 258–263, 2013b.

J. Cyranka, M. A. Islam, G. Byrne, P. Jones, S. A. Smolka, and R. Grosu. Lagrangian reachabililty. In Rupak Majumdar and Viktor Kunčak, editors, *CAV'17, the 29th International Conference on Computer-Aided Verification*, pages 379–400, Heidelberg, Germany, July 2017. Springer.

Alessandro De Palma, Rudy Bunel, Alban Desmaison, Krishnamurthy Dvijotham, Pushmeet Kohli, Philip HS Torr, and M Pawan Kumar. Improved branch and bound for neural network verification via lagrangian decomposition. *arXiv preprint arXiv:2104.06718*, 2021.

Joseph DelPreto, Andres F Salazar-Gomez, Stephanie Gil, Ramin M Hasani, Frank H Guenther, and Daniela Rus. Plug-and-play supervisory control using muscle and brain signals for real-time gesture and error detection. In *Robotics: Science and Systems*, 2018.

Alex Devonport, Mahmoud Khaled, Murat Arcak, and Majid Zamani. Pirk: Scalable interval reachability analysis for high-dimensional nonlinear systems. In Shuvendu K. Lahiri and Chao Wang, editors, *Computer Aided Verification*, pages 556–568, Cham, 2020. Springer International Publishing.

Alexandre Donzé. Breach, a toolbox for verification and parameter synthesis of hybrid systems. In *CAV'10, the 22nd International Conference on Computer Aided Verification*, pages 167–170, Edinburgh, UK, July 2010. Springer.

Alexandre Donzé and Oded Maler. Systematic simulation using sensitivity analysis. In *International Workshop on Hybrid Systems: Computation and Control*, pages 174–189. Springer, 2007.

Parasara Sridhar Duggirala, Sayan Mitra, Mahesh Viswanathan, and Matthew Potok. C2e2: A verification tool for stateflow models. In Christel Baier and Cesare Tinelli, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 68–82, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

Ruediger Ehlers. Formal verification of piece-wise linear feed-forward neural networks. In *International Symposium on Automated Technology for Verification and Analysis*, pages 269–286. Springer, 2017.

Joshua A. Enszer and Mark A. Stadtherr. Verified solution and propagation of uncertainty in physiological models. *Reliab. Comput.*, 15(3):168–178, 2011. URL http://interval.louisiana.edu/reliable-computing-journal/volume-15/no-3/reliable-computing-15-pp-168-178.pdf.

Chuchu Fan, James Kapinski, Xiaoqing Jin, and Sayan Mitra. Simulation-driven reachability using matrix measures. *ACM Trans. Embed. Comput. Syst.*, 17(1), December 2017.

Mahyar Fazlyab, Alexander Robey, Hamed Hassani, Manfred Morari, and George Pappas. Efficient and accurate estimation of lipschitz constants for deep neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/95e1533eb1b20a97777749fb94fdb944-Paper.pdf.

Martin Fränzle, Ernst Moritz Hahn, Holger Hermanns, Nicolás Wolovick, and Lijun Zhang. Measurability and safety verification for stochastic hybrid systems. In *Proceedings of the 14th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2011, Chicago, IL, USA, April 12-14, 2011*, pages 43–52, 2011.

Martin Fränzle, Tino Teige, and Andreas Eggers. Engineering constraint solvers for automatic analysis of probabilistic hybrid automata. *The Journal of Logic and Algebraic Programming*, 79 (7):436 – 466, 2010. The 20th Nordic Workshop on Programming Theory (NWPT 2008).

Yarin Gal. Uncertainty in deep learning. *University of Cambridge*, 1(3):4, 2016.

S. Gao, S. Kong, and E. M. Clarke. Satisfiability modulo odes. In *2013 Formal Methods in Computer-Aided Design*, pages 105–112, 2013.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

Sophie Gruenbacher, Jacek Cyranka, Mathias Lechner, Md Ariful Islam, Scott A. Smolka, and Radu Grosu. Lagrangian reachtubes: The next generation. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 1556–1563, Dec 2020.

Sophie Grunbacher, Ramin Hasani, Mathias Lechner, Jacek Cyranka, Scott A. Smolka, and Radu Grosu. On the verification of neural odes with stochastic guarantees. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):11525–11535, May 2021.

Amit Gurung, Rajarshi Ray, Ezio Bartocci, Sergiy Bogomolov, and Radu Grosu. Parallel reachability analysis of hybrid systems in xspeed. *International Journal on Software Tools for Technology Transfer*, 21(4):401–423, 2019.

Eldon Hansen and G William Walster. *Global optimization using interval analysis: revised and expanded*, volume 264. CRC Press, 2003.

N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

Ramin Hasani. *Interpretable recurrent neural networks in continuous-time control environments*. PhD thesis, Wien, 2020.

Ramin Hasani, Alexander Amini, Mathias Lechner, Felix Naser, Radu Grosu, and Daniela Rus. Response characterization for auditing cell dynamics in long short-term memory networks. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019a.

Ramin Hasani, Guodong Wang, and Radu Grosu. A machine learning suite for machine components' health-monitoring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9472–9477, 2019b.

Ramin Hasani, Mathias Lechner, Alexander Amini, Daniela Rus, and Radu Grosu. The natural lottery ticket winner: Reinforcement learning with ordinary neural circuits. In *Proceedings of the 2020 International Conference on Machine Learning*. JMLR. org, 2020.

Ramin Hasani, Mathias Lechner, Alexander Amini, Lucas Liebenwein, Max Tschaikowski, Gerald Teschl, and Daniela Rus. Closed-form continuous-depth models. *arXiv preprint arXiv:2106.13898*, 2021a.

Ramin Hasani, Mathias Lechner, Alexander Amini, Daniela Rus, and Radu Grosu. Liquid time-constant networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9):7657–7666, May 2021b.

Ramin M Hasani, Dieter Haerle, and Radu Grosu. Efficient modeling of complex analog integrated circuits using neural networks. In *2016 12th Conference on Ph. D. Research in Microelectronics and Electronics (PRIME)*, pages 1–4. IEEE, 2016.

Ramin M Hasani, Magdalena Fuchs, Victoria Beneder, and Radu Grosu. Non-associative learning representation in the nervous system of the nematode caenorhabditis elegans. *arXiv preprint arXiv:1703.06264*, 2017a.

Ramin M Hasani, Dieter Haerle, Christian F Baumgartner, Alessio R Lomuscio, and Radu Grosu. Compositional neural-network modeling of complex analog circuits. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2235–2242. IEEE, 2017b.

Patrick Henriksen and Alessio Lomuscio. Efficient neural network verification via adaptive refinement and adversarial search. In *ECAI 2020*, pages 2513–2520. IOS Press, 2020.

Thomas A Henzinger and Vlad Rusu. Reachability verification for hybrid automata. In *International Workshop on Hybrid Systems: Computation and Control*, pages 190–204. Springer, 1998.

Thomas A Henzinger, Mathias Lechner, and Đorđe Žikelić. Scalable verification of quantized neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3787–3795, 2021.

Chao Huang, Xin Chen, Wang Lin, Zhengfeng Yang, and Xuandong Li. Probabilistic safety verification of stochastic hybrid systems using barrier certificates. *ACM Trans. Embed. Comput. Syst.*, 16 (5s), September 2017. ISSN 1539-9087.

C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation*, 15(1):1–28, 2007.

Fabian Immler. Verified reachability analysis of continuous systems. In Christel Baier and Cesare Tinelli, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 37–51, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

Md Ariful Islam, Qinsi Wang, Ramin M Hasani, Ondrej Balún, Edmund M Clarke, Radu Grosu, and Scott A Smolka. Probabilistic reachability analysis of the tap withdrawal circuit in caenorhabditis elegans. In *2016 IEEE International High Level Design Validation and Test Workshop (HLDVT)*, pages 170–177. IEEE, 2016.

Tomasz Kapela, Marian Mrozek, Daniel Wilczak, and Piotr Zgliczynski. Capd:: Dynsys: a flexible c++ toolbox for rigorous numerical analysis of dynamical systems. *Pre-Print - ww2.ii.uj.edu.pl*, 2020.

Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017.

Dmitri E Kvasov and Ya D Sergeyev. Lipschitz global optimization methods in control problems. *Automation and Remote Control*, 74(9):1435–1448, 2013.

Mathias Lechner and Ramin Hasani. Learning long-term dependencies in irregularly-sampled time series. *arXiv preprint arXiv:2006.04418*, 2020.

Mathias Lechner, Ramin Hasani, Manuel Zimmer, Thomas A Henzinger, and Radu Grosu. Designing worm-inspired neural networks for interpretable robotic control. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 87–94. IEEE, 2019.

Mathias Lechner, Ramin Hasani, Alexander Amini, Thomas A Henzinger, Daniela Rus, and Radu Grosu. Neural circuit policies enabling auditable autonomy. *Nature Machine Intelligence*, 2(10): 642–652, 2020a.

Mathias Lechner, Ramin Hasani, Daniela Rus, and Radu Grosu. Gershgorin loss stabilizes the recurrent neural network compartment of an end-to-end robot learning scheme. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5446–5452. IEEE, 2020b.

Mathias Lechner, Ramin Hasani, Radu Grosu, Daniela Rus, and Thomas A Henzinger. Adversarial training is not ready for robot learning. *arXiv preprint arXiv:2103.08187*, 2021.

Dongxu Li, Stanley Bak, and Sergiy Bogomolov. Reachability analysis of nonlinear systems using hybridization and dynamics scaling. In Nathalie Bertrand and Nils Jansen, editors, *Formal Modeling and Analysis of Timed Systems*, pages 265–282, Cham, 2020a. Springer International Publishing.

Xuechen Li, Ting-Kam Leonard Wong, Ricky TQ Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics*, pages 3870–3882. PMLR, 2020b.

Lucas Liebenwein, Ramin Hasani, Alexander Amini, and Daniela Rus. Sparse flows: Pruning continuous-depth models. *arXiv preprint arXiv:2106.12718*, 2021.

J Lu and P Mudigonda. Nueral network branching for nueral network verification. In *Proceedings of the International Conference on Learning Representations (ICLR 2020)*. Open Review, 2020.

Cedric Malherbe and Nicolas Vayatis. Global optimization of lipschitz functions. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 2314–2323. JMLR.org, 2017.

Pierre-Jean Meyer, Alex Devonport, and Murat Arcak. Tira: Toolbox for interval reachability analysis. In *Association for Computing Machinery*, HSCC '19, page 224–229, New York, NY, USA, 2019.

Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*, pages 3578–3586. PMLR, 2018.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 807–814, 2010.

Arnold Neumaier. Complete search in continuous global optimization and constraint satisfaction. *Acta Numerica*, 13:271–369, 2004.

S.A. Piyavskii. An algorithm for finding the absolute extremum of a function. *USSR Computational Mathematics and Mathematical Physics*, 12(4):57 – 67, 1972.

Michael Poli, Stefano Massaroli, Atsushi Yamashita, Hajime Asama, Jinkyoo Park, et al. Hypersolvers: Toward fast continuous-depth models. *Advances in Neural Information Processing Systems*, 33, 2020.

Lev Semenovich Pontryagin. *Mathematical theory of optimal processes*. Routledge, 2018.

Wenjie Ruan, Xiaowei Huang, and Marta Kwiatkowska. Reachability analysis of deep neural networks with provable guarantees. *arXiv preprint arXiv:1805.02242*, 2018.

Hadi Salman, Greg Yang, Huan Zhang, Cho-Jui Hsieh, and Pengchuan Zhang. A convex relaxation barrier to tight robustness verification of neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/246a3c5544feb054f3ea718f61adfa16-Paper.pdf.

Fedor Shmarov and Paolo Zuliani. Probreach: verified probabilistic delta-reachability for stochastic hybrid systems. In Antoine Girard and Sriram Sankaranarayanan, editors, *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control, HSCC'15, Seattle, WA, USA, April 14-16, 2015*, pages 134–139. ACM, 2015a.

Fedor Shmarov and Paolo Zuliani. Probreach: A tool for guaranteed reachability analysis of stochastic hybrid systems. In Sergiy Bogomolov and Ashish Tiwari, editors, *1st International Workshop on Symbolic and Numerical Methods for Reachability Analysis, SNR@CAV 2015, San Francisco, CA, USA, July 19, 2015*, volume 37 of *EPiC Series in Computing*, pages 40–48. EasyChair, 2015b. URL https://easychair.org/publications/paper/z1f.

Bruno O. Shubert. A sequential method seeking the global maximum of a function. *SIAM Journal on Numerical Analysis*, 9(3):379–388, 1972. ISSN 00361429. URL http://www.jstor.org/stable/2156138.

Gagandeep Singh, Jonathan Maurer, Christoph Müller, Matthew Mirman, Timon Gehr, Adrian Hoffmann, Petar Tsankov, Dana Drachsler Cohen, Markus Püschel, and Martin Vechev. Eth robustness analyzer for neural networks (eran). *URL https://github. com/eth-sri/eran*, 2020.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Christian Tjandraatmadja, Ross Anderson, Joey Huchette, Will Ma, KRUNAL KISHOR PATEL, and Juan Pablo Vielma. The convex relaxation barrier, revisited: Tightened single-neuron relaxations for neural network verification. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21675–21686. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/f6c2a0c4b566bc99d596e58638e342b0-Paper.pdf.

Vincent Tjeng, Kai Y Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. In *International Conference on Learning Representations*, 2018.

Jonathan Uesato, Brendan O'donoghue, Pushmeet Kohli, and Aaron Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*, pages 5025–5034. PMLR, 2018.

Abraham P Vinod and Meeko MK Oishi. Stochastic reachability of a target tube: Theory and computation. *Automatica*, 125:109458, 2021.

Charles Vorbach, Ramin Hasani, Alexander Amini, Mathias Lechner, and Daniela Rus. Causal navigation by continuous-time neural networks. *arXiv preprint arXiv:2106.08314*, 2021.

Guodong Wang, Ramin M Hasani, Yungang Zhu, and Radu Grosu. A novel bayesian network-based fault prognostic method for semiconductor manufacturing process. In *2017 IEEE International Conference on Industrial Technology (ICIT)*, pages 1450–1454. IEEE, 2017.

Guodong Wang, Anna Ledwoch, Ramin M Hasani, Radu Grosu, and Alexandra Brintrup. A generative neural network model for the quality prediction of work in progress products. *Applied Soft Computing*, 85:105683, 2019.

Qinsi Wang, Paolo Zuliani, Soonho Kong, Sicun Gao, and Edmund M. Clarke. Sreach: A probabilistic bounded delta-reachability analyzer for stochastic hybrid systems. In Olivier Roux and Jérémie Bourdon, editors, *Computational Methods in Systems Biology*, pages 15–27, Cham, 2015. Springer International Publishing.

Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. Efficient formal safety analysis of neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/2ecd2bd94734e5dd392d8678bc64cdab-Paper.pdf.

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5286–5295. PMLR, 2018.

Winnie Xu, Ricky TQ Chen, Xuechen Li, and David Duvenaud. Infinitely deep bayesian neural networks with stochastic differential equations. *arXiv preprint arXiv:2102.06559*, 2021.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/d04863f100d59b3eb688a11f95b0ae60-Paper.pdf.

Anatoly Zhigljavsky and Antanas Zilinskas. *Stochastic Global Optimization*, volume 9 of *Springer Optimization and Its Applications*. Springer US, 2008.

S1 Proofs

Theorem 3 (Radius of Lipschitz Caps) *Given the continuous-depth model f of Eq. (1) in the main paper ($\partial_t x = f(x)$ with $x(t_0) \in B(x_0, \delta_0)$). Let $\gamma \in (0,1)$, $\mu > 1$, target time t_j, the set of all sampled points $\mathcal{V}$, the number of sampled points $N = |\mathcal{V}|$, the sample maximum $\bar{m}_{j,\mathcal{V}} = \max_{x \in \mathcal{V}} d_j(x)$, the IVP solutions $\chi(t_j, x)$, and the corresponding stretching factors $\lambda_x = \|\partial_x \chi(t_j, x)\|$ for all $x \in \mathcal{V}$. For $x \in \mathcal{V}$, let $\nu_x = |\lambda_x - \lambda_X| / \|x - X\|$ be a new random variable, where $X \in \mathcal{B}_0^S$ is the random variable which is thrown by random sampling on the surface of the initial ball. Let the upper bound $\Delta\lambda_{x,\mathcal{V}}$ of the confidence interval of $\mathbb{E}\nu_x$ be defined as follows:*

$$\Delta\lambda_{x,\mathcal{V}}(\gamma) = \bar{\nu}_x + t^*_{\gamma/2}(N-2)\frac{s(\nu_x)}{\sqrt{N-1}}, \tag{S1}$$

with $\bar{\nu}_x$ and $s(\nu_x)$ being the sample mean and sample standard deviation of ν_x, and t^* being the Student's t-distribution. Let r_x be defined as:

$$r_x = (2 \cdot \Delta\lambda_{x,\mathcal{V}})^{-1} \left(-\lambda_x + \sqrt{\lambda_x^2 + 4 \cdot \Delta\lambda_{x,\mathcal{V}} \cdot (\mu \cdot \bar{m}_{j,\mathcal{V}} - d_j(x))} \right), \tag{S2}$$

then it holds that:

$$\Pr\left(d_j(y) \leq \mu \cdot \bar{m}_{j,\mathcal{V}}\right) \geq 1 - \gamma \quad \forall y \in B(x, r_x)^S, \tag{S3}$$

and thus that $B(x, r_x)^S$ is a γ, t_j-Lipschitz cap.

Proof. Eq. (3) defines $\Delta\lambda_{x,\mathcal{V}}$ as the upper bound of the confidence interval for the mean $\mathbb{E}\nu_x$ with unknown standard deviation, confidence coefficient $1-\gamma$ and sample size $N-1$ (we compare the stretching factor λ_x with the ones of the other $N-1$ samples), thus

$$\Pr(\Delta\lambda_{x,\mathcal{V}} \geq \mathbb{E}\nu_x) \geq 1 - \gamma, \quad \text{with} \quad \mathbb{E}\nu_x = \mathbb{E}\left[\frac{|\lambda_x - \lambda_X|}{\|x - X\|}\right] \tag{S4}$$

It holds for $x, y \in \mathcal{V}$ that:

$$\begin{aligned}
\lambda_y &= \lambda_x + \frac{\lambda_y - \lambda_x}{\|x-y\|} \cdot \|x - y\| \\
&\leq \lambda_x + \frac{|\lambda_x - \lambda_y|}{\|x-y\|} \cdot \|x - y\|, \quad \text{thus using Eq. (S4)} \\
\Pr\left(\lambda_y \leq \lambda_x + \Delta\lambda_{x,\mathcal{V}} \cdot \|x - y\|\right) &\geq 1 - \gamma
\end{aligned} \tag{S5}$$

Using the mean value inequality for vector-valued functions it holds that:

$$|d_j(x) - d_j(y)| = \big|\|\chi(t_j, x) - \chi(t_j, x_0)\| - \|\chi(t_j, y) - \chi(t_j, x_0)\|\big| \quad \{\text{triangle inequality}\} \tag{S6}$$

$$\leq \|\chi(t_j, x) - \chi(t_j, y)\| \quad \{\text{mean value theorem}\} \tag{S7}$$

$$\Rightarrow \exists z \in [x, y] \colon |d_j(x) - d_j(y)| \leq \|\partial_x \chi(t_j, z)\| \|x - y\| = \lambda_z \cdot \|x - y\| \tag{S8}$$

Combining this with Eq. (S5) and thus using $\lambda_x + \Delta\lambda_{x,\mathcal{V}} \cdot \|x - y\|$ as a probabilistic upper bound for λ_z, we obtain the following results for all y with $\|x - y\| \leq r_x$:

$$\Pr\left(|d_j(x) - d_j(y)| \leq (\lambda_x + \Delta\lambda_{x,\mathcal{V}} \cdot \|x - y\|) \cdot \|x - y\|\right) \geq 1 - \gamma \tag{S9}$$

$$\Pr\left(|d_j(x) - d_j(y)| \leq (\lambda_x + \Delta\lambda_{x,\mathcal{V}} \cdot r_x) \cdot r_x\right) \geq 1 - \gamma \tag{S10}$$

As r_x defined like in Eq. (4) is the solution of the quadratic equation $\mu \cdot \bar{m}_{j,\mathcal{V}} - d_j(x) = \lambda_x r_x + \Delta\lambda_{x,\mathcal{V}} r_x^2$, it holds that:

$$\Pr\left(|d_j(x) - d_j(y)| \leq \mu \cdot \bar{m}_{j,\mathcal{V}} - d_j(x))\right) \geq 1 - \gamma \quad \forall y \in B(x, r_x)^S \tag{S11}$$

As in the proof of (Grunbacher et al., 2021)[Theorem 1] we now distinguish between two cases for y: (a) $d_j(y) \leq d_j(x)$ and (b) $d_j(y) \geq d_j(x)$. In case (a) it is trivial: $d_j(y) \leq d_j(x) \leq \mu \cdot \bar{m}_{j,\mathcal{V}}$. Having case (b), Eq. (S11) is equivalent to

$$\Pr\left(d_j(y) - d_j(x) \leq \mu \cdot \bar{m}_{j,\mathcal{V}} - d_j(x))\right) \geq 1 - \gamma \quad \Longleftrightarrow \tag{S12}$$

$$\Pr\left(d_j(y) \leq \mu \cdot \bar{m}_{j,\mathcal{V}})\right) \geq 1 - \gamma, \tag{S13}$$

thus Eq. (5) holds and $B(x, r_x)^S$ is a Lipschitz cap.

Theorem 4 (Convergence Guarantee using Lipschitz caps) *Given the over-approximation factor $\mu > 1$, the set of all sampled points $\mathcal{V}$ and the sample maximum $\bar{m}_{j,\mathcal{V}} = \max_{x \in \mathcal{V}} d_j(x)$. Let $m_j^\star = \max_{x \in \mathcal{B}_0} d_j(x)$ be the initial-ball maximum (the global maximum). Then:*

$$\forall \gamma \in (0,1), \exists N \in \mathbb{N} \text{ s.t. } \Pr(\mu \cdot \bar{m}_{j,\mathcal{V}} \geq m_j^\star) \geq 1 - \gamma, \tag{S14}$$

where $N = |\mathcal{V}|$ is the number of sampled points.

Proof. Let $x_j^\star$ be a point such that $d_j(x_j^\star) = m_j^\star$. Given $\gamma \in (0,1)$ and cap radii r_x as defined in Eq. (4), we know from the proof of (Grunbacher et al., 2021)[Theorem 2] that

$$\Pr(\exists y \in \mathcal{V} \colon B(y, r_y)^S \ni x_j^\star) = 1 - \prod_{x \in \mathcal{V}} \left(1 - p_{r_x}\right), \quad \text{with} \tag{S15}$$

$$p_{r_x} = \Pr(B(x, r_x)^S \ni x_j^\star) = \frac{\text{Area}(B(x, r_x)^S)}{\text{Area}\left(\mathcal{B}_0\right)} \tag{S16}$$

As in the proof of (Grunbacher et al., 2021)[Theorem 2] we derive a lower bound of r_x by using the first sample $x_{j,1}$ and replacing the values in Eq. (4) as follows:

$$\mu \cdot \bar{m}_{j,\mathcal{V}} - d_j(x) \geq \mu \cdot \bar{m}_{j,\mathcal{V}} - \bar{m}_{j,\mathcal{V}} = (\mu - 1) \cdot \bar{m}_{j,\mathcal{V}} \geq (\mu - 1) \cdot d_j(x_{j,1}), \tag{S17}$$

thus a lower bound of all Lipschitz cap radii is given by

$$r_{bound} = (2 \cdot \Delta\lambda_{x,\mathcal{V}})^{-1} \left(-\lambda_x + \sqrt{\lambda_x^2 + 4 \cdot \Delta\lambda_{x,\mathcal{V}} \cdot (\mu - 1) \cdot d_j(x_{j,1})} \right) \leq r_x \quad \forall x \in \mathcal{V} \tag{S18}$$

$$\Rightarrow \Pr(\exists y \in \mathcal{V} \colon B(y, r_y)^S \ni x_j^\star) \geq 1 - (1 - p_{r_{bound}})^N \tag{S19}$$

As in the limit of $N \to \infty$ the probability of Eq. (S19) is 1, it follows that $\forall \gamma \in (0,1)\ \exists N \in \mathbb{N} \colon \Pr(\exists x \in \mathcal{V} \colon B(x, r_x)^S \ni x_j^\star) \geq \sqrt{1 - \gamma}$.

Using a set of sampled points $\mathcal{V}$ with cardinality N and using $\hat{\gamma} = 1 - \sqrt{1 - \gamma}$ as the error rate for the upper bound $\Delta\lambda_x$ of the confidence interval in Eq. (3). Using the result of Theorem 1, the resulting probability is:

$$\Pr\left(d_j(y) \leq \mu \cdot \bar{m}_{j,\mathcal{V}}\right) \geq 1 - \hat{\gamma} = \sqrt{1 - \gamma} \quad \forall y \in B(x, r_x)^S \tag{S20}$$

If there is an $x \in \mathcal{V}$ such that $B(x, r_x)^S \ni x_j^\star$, then Eq. (S20) obviously holds also for $x_j^\star$, thus:

$$\Pr(d_j(x^\star) \leq \mu \cdot \bar{m}_{j,\mathcal{V}} | \exists x \in \mathcal{V} \colon B(x, r_x)^S \ni x^\star) \geq \sqrt{1 - \gamma} \tag{S21}$$

For any sets A, B it holds that $\Pr(A) \geq \Pr(A \cap B) = \Pr(A|B) \cdot \Pr(B)$, and using:

$$A = (\mu \cdot \bar{m}_{j,\mathcal{V}} \geq m_j^\star) \tag{S22}$$

$$B = (\exists x \in \mathcal{V} \colon B(x, r_x)^S \ni x_j^\star) \tag{S23}$$

it follows that $\Pr(\mu \cdot \bar{m}_{j,\mathcal{V}} \geq m_j^\star) \geq \Pr(A|B) \cdot \Pr(B) = 1 - \gamma$ and therefore Eq. (6) holds.