TU WIEN Informatics

Robustness Analysis of Continuous-Depth Neural Networks

DISSERTATION

zur Erlangung des akademischen Grades

Doktorin der Technischen Wissenschaften

eingereicht von

Dipl.-Ing. Sophie A. Neubauer
Matrikelnummer 00926081

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dr.rer.nat. Radu Grosu
Zweitbetreuung: Prof. Dr. Georg Gottlob

Diese Dissertation haben begutachtet:

Daniela Rus	Sriram Sankaranarayanan

Wien, 30. April 2023

Sophie A. Neubauer

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

TU WIEN Informatics

Robustness Analysis of Continuous-Depth Neural Networks

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktorin der Technischen Wissenschaften

by

Dipl.-Ing. Sophie A. Neubauer
Registration Number 00926081

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dr.rer.nat. Radu Grosu
Second advisor: Prof. Dr. Georg Gottlob

The dissertation has been reviewed by:

Daniela Rus	Sriram Sankaranarayanan

Vienna, 30th April, 2023

Sophie A. Neubauer

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Dipl.-Ing. Sophie A. Neubauer

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 30. April 2023

Sophie A. Neubauer

Acknowledgements

First and foremost, I want to thank my supervisor, Radu Grosu, for being a constant source of inspiration. His guidance and mentorship have been invaluable, providing me with both freedom and direction in my research. His contagious enthusiasm for research questions, as well as his ability to discuss technical details and share ideas, have greatly enriched my work. I extend my appreciation to my thesis committee, Georg Gottlob, Daniela Rus and Sriram Sankaranarayanan, for their valuable feedback, suggestions, and critical evaluation of my research. Their expertise and constructive criticism helped me improve the quality of my work significantly. I would like to express my appreciation to the academic and support staff at DK LogiCS, especially Anna Prianichnikova and Beatrix Buhl, for their assistance, guidance, and cooperation throughout my academic journey. I feel honored to have visited Daniela Rus at MIT Computer Science and Artificial Intelligence Laboratory (CSAIL). My research stay at her group expanded my horizons and boosted my scientific career. Thanks for welcoming me into that community. I would like to whole-heartedly thank my colleagues at TU Wien CPS group and collaborators, especially Alexander Amini, Zahra Babaiee, Guillaume Berger, Jacek Cyranka, Ramin Hasani, Thomas Henzinger, Md. Ariful Islam, Marvin Kleinlehner, Mathias Lechner, Julian Lemmel, Anna Lukina, Ivan Majic, Daniela Rus, Johannes Scholz, Scott A. Smolka and Max Tschaikowski. Thanks for engaging in fruitful discussions, providing insights, and sharing ideas that have enriched my research. Their contributions have been invaluable in shaping the quality of my work.

I am incredibly thankful to my husband for inspiring me to follow my dreams and aim for high goals in science. His unwavering support, interest in my research, and thought-provoking questions have been instrumental in shaping my perspectives and overcoming challenges. I am grateful for his encouragement and help to align my focus on science. I extend my heartfelt thanks to my parents for always believing in me and supporting me. Their unwavering love and encouragement have been a driving force behind my success, and I am grateful for their constant support. I am also thankful to my brother for engaging in discussions about life goals and aspirations, especially during times when clarity was lacking. I would also like to thank my friends who have been a source of encouragement and support throughout my thesis journey.

Funding sources. Austrian Science Fund (FWF) project no. W1255-N23, the Austrian Research Promotion Agency (FFG) project no. FO99988751 and DatenVorsprung.

Kurzfassung

Diese Dissertation konzentriert sich auf Theorien und Algorithmen zur deterministischen und statistischen Erreichbarkeitsanalyse von Cyber-physischen Systemen, welche durch neuronale Netze mit kontinuierlicher Tiefe (CDNN) geregelt werden. Hierbei betrachten wir geschlossene Systeme, deren Zustandsdynamik durch ein System von Differentialgleichungen (ODEs) gegeben ist. Die Motivation dieser Arbeit ist es, das große Potenzial von CDNN bei Anwendungen, welche für traditionelle Regler zu komplex sind, ausschöpfen zu können. Die Fragestellung lautet daher: Wie können Sicherheitsgarantien, vorhersagbares Verhalten und damit Vertrauenswürdigkeit von CDNN-Regler bereitgestellt werden, sodass der Einsatz auch für sicherheitskritische Anwendungen möglich wird?

In dieser Arbeit verbessern und erweitern wir den Stand der Technik mit zwei komplementären Ansätzen: Der erste besteht aus dem Algorithmus LRT-NG und einer Reihe von analytischen, symbolischen Techniken zur Approximation von nichtlinearen ODEs. Hierbei wird zum ersten Mal die optimale Metrik der Kugel, welche die erreichbaren Zustände umschließt, bewiesenermaßen analytisch so berechnet, dass diese Metrik das Volumen der Kugel minimiert. Im zweiten Teil wird zunächst eine reine Theorie dargestellt und bewiesen, dass neuronale ODEs, eine Teilklasse der CDNNs, durch die Lösung einer Reihe globaler Optimierungsprobleme verifiziert werden können. Anschließend wird ein neuer statistischer Verifizierungsalgorithmus, GoTube, vorgestellt, der die Robustheit eines beliebigen zeitkontinuierlichen Prozesses, der als CDNN-Modell formuliert ist, formal quantifiziert, indem er statistische obere Schranken lokaler Lipschitz-Konstanten berechnet. LRT-NG erhielt den Outstanding Student-Paper Award des IEEE CPS-DES TC, und GoTube wurde mit dem Scientia Preis ausgezeichnet.

Wir vergleichen LRT-NG mit modernsten konservativen Algorithmen wie LRT, CAPD und Flow* und zeigen seine Überlegenheit anhand Tests an umfassenden ODE-Benchmarks, einschließlich zweier neuronaler ODEs, wobei LRT-NG als einziger CDNNs verifizieren kann. GoTube, unser statistischer Algorithmus, akkumuliert - im Vergleich zum Stand der Technik - keine Fehler zwischen den Zeitschritten und vermeidet den berüchtigten Wrapping-Effekt, der symbolischen Techniken innewohnt. Wir zeigen anhand einer großen Anzahl von Experimenten, dass GoTube die modernsten Verifizierungsalgorithmen in Bezug auf die mögliche Größe der Anfangskugel, den Zeithorizont und die Skalierbarkeit deutlich übertrifft. Unsere Programme, LRT-NG und GoTube, sind auf GitHub frei verfügbar: `https://github.com/DatenVorsprung`.

Abstract

The main focus of this thesis is the development of a theory and associated algorithms and tools, for the deterministic and statistical reachability analysis, of cyber-physical systems controlled by continuous-depth neural networks (CDNNs). We assume that the dynamics of the closed-system's states is given by a set of ordinary differential equations (ODEs), and the output is a function of the solution of the ODEs at a given time.

The primary motivation for this work is the huge potential of CDNNs in the design and implementation of safety-critical applications which are required to solve difficult tasks. So the main question of the thesis is: How to provide safety-guarantees, predictable behaviour, strong assurances, and thus trustworthiness, such that the use of CDNN controllers becomes a feasible strategy for safety-critical applications, too?

In this thesis, we improve and extend state-of-the-art with two complementary approaches. The first, introduces a set of conservative, symbolic techniques and an algorithm, LRT-NG, for the reachability analysis of nonlinear ODEs. This uses for the first time an analytically computed metric for the ball enclosing the propagated reachable states, which is proven to minimize the ball's volume. The second, first discusses a purely theoretical framework and shows that Neural-ODEs, an emerging class of CDNNs, can be verified by solving a set of global-optimization problems. It then introduces a new statistical verification algorithm, GoTube, that formally quantifies the behavioural robustness of any time-continuous process formulated as a CDNN model, by computing statistical upper bounds of local Lipschitz constants. LRT-NG received the Outstanding Student-Paper Award from the IEEE CPS-DES TC, and GoTube won the Scientia Prize.

We experimentally demonstrate that LRT-NG, our conservative algorithm, is the only symbolic tool capable of handling CDNNs, compared to the state-of-the-art tools such as LRT, CAPD and Flow*. Moreover, our experiments on a comprehensive set of ODE benchmarks, including two Neural ODEs, demonstrates LRT-NG's superior performance. Compared to advanced reachability analysis tools for time-continuous neural networks, our statistical theory, and algorithm GoTube, 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. Our tools, LRT-NG and GoTube, are freely available on GitHub: https://github.com/DatenVorsprung.

Contents

List of Publications and Tools

This cumulative thesis is based on the following three **publications**.

CDC-20 **S. Gruenbacher**, J. Cyranka, M. Lechner, M.A. Islam, S.A. Smolka and R. Grosu. "Lagrangian Reachtubes: The Next Generation," Proceedings of the 59th IEEE Conference on Decision and Control (CDC), Jeju, Korea (South), 2020, pp. 1556-1563, doi: 10.1109/CDC42340.2020.9304042. This paper won the **IEEE TC Outstanding Student Paper Prize.**

AAAI-21 **S. Gruenbacher**, R. Hasani, M. Lechner, J. Cyranka, S.A. Smolka, and R. Grosu. "On the Verification of Neural ODEs with Stochastic Guarantees", Proceedings of the 35th AAAI Virtual Conference on Artificial Intelligence, 35(13), February, 2021, pages 11525-11535.

AAAI-22 **S.A. Gruenbacher**, M. Lechner, R. Hasani, D. Rus, T.A. Henzinger, S.A. Smolka, and R. Grosu. "GoTube: Scalable Stochastic Verification of Continuous-Depth Models", Proceedings of the 36th AAAI Virtual Conference on Artificial Intelligence, February, 2022, pages 6755-6764. This paper won the **Scientia Prize** presented by Joerg Schmiedmayer.

My contributions to the three before-mentioned publications are the following: I developed the ideas and theoretical background during several scientific discussions with the co-authors. I formulated the Theorems and conducted the proofs. The main part of the manuscripts and most figures and tables were generated by me. I set up the experimental design and conducted the experiments together with Mathias Lechner. The code for the tools LRT-NG and GoTube were written by me and were developed in the course of the before-mentioned publications. Short description of the **tools**:

LRT-NG A toolset that computes a conservative reachtube of continuous-depth neural networks as well as any other nonlinear dynamical systems. The code is written in C++. After installing the external libraries IBEX, Eigen and Boost, the code can be easily compiled via cmake and used on different models through an easy interface, without the need of recompiling it. The code can be found here: `https://github.com/DatenVorsprung/LRTNG`

GoTube This tool constructs statistical reachtubes of continuous-time systems. GoTube is made deliberately for the verification of continuous-depth neural networks. The code is written in python. After installing the requirements and defining the model, there are a few arguments that need to be setup, e.g. the error probability *gamma* or the maximum multiplicative tolerance of over-approximation *mu*. The code can be found here: `https://github.com/DatenVorsprung/GoTube`

Moreover, the following publications and co-supervised master thesis were published during my PhD research. Some are in the research field of this thesis and others loosely connected to the content. Throughout this thesis all publications listed in this chapter will not be cited explicitly, but rather quoted in verbatim.

ARCH-19 **S. Gruenbacher**, J. Cyranka, M.A. Islam, M. Tschaikowski, S.A. Smolka, and R. Grosu. "Under the Hood of a Stand-Alone Lagrangian Reachability Tool". In G. Frehse and M. Althoff, editors, ARCH19. 6th International Workshop on Applied Verification of Continuous and Hybrid Systems, part of CPS-IoT Week 2019, Montreal, QC, Canada, April 15, 2019, volume 61 of EPiC Series in Computing, pages 211–219. EasyChair, 2019.

VISU-21 M. Sbardellati. "Exploratory Visual System for Predictive Machine Learning of Event-Organisation Data". Advisor: M. Waldner, Co-Advisor: **S. Gruenbacher**, E. Groeller. Institute for Visual Computing and Human-Centered Technology. TU Wien, 2021.

HENZ-22 **S.A. Neubauer (née Gruenbacher)**, R. Grosu. "Robustness Analysis of Continuous-Depth Models with Lagrangian Techniques", Accepted for publication at Henzinger-60 (Thomas Henzinger Festschrift - Conference celebrating his 60th birthday), 2022.

AI4TS-22 J. Lemmel*, Z. Babaiee*, M. Kleinlehner, I. Majic, P. Neubauer, J. Scholz, R. Grosu, **S.A. Neubauer (née Gruenbacher)**. "Deep-Learning vs Regression: Prediction of Tourism Flow with Limited Data", Accepted for publication at the IJCAI'22 Workshop AI for Time Series Analysis (AI4TS-22), 2022.

Part I

Introduction

CHAPTER 1

Motivation and Problem Statement

Since the advent of neural ordinary differential equations (Neural ODEs) [CRBD18], modern cyber-physical systems (CPS) increasingly use deep-learning systems powered by continuous-depth neural networks (CDNN). In these networks, the dynamics of the hidden states is defined by a set of nonlinear ordinary differential equations (ODE) and the output is a function of the solution of the ODEs at a given time. CDNNs thus generalise Neural ODEs and are used within the cyber part of a CPS, responsible for state-estimation, planning, and (adaptive) optimal control, of the physical part of the CPS. CDNNs are especially suited for the task of controlling safety-critical CPS, as they are: 1) more robust against both random Gaussian perturbations and adversarial attacks than conventional convolutional neural networks [YDTF20], 2) there is a better characterization of Neural ODEs [RCD19, DDT19, DBMP19, JB19], and 3) a better understanding of their stability [YWL+20], and controllability [QGMK19, HKT20, KMFL20]. As the use of CDNNs in real-world applications increases [FJNO20, LHA+20, EAQM20, LH20, HLA+20], so does the importance of ensuring their safety through the use of verification techniques, such as their reachability analysis (see Fig. 1.1).

Formally, a CDNN is an *infinite depth* neural network, which means that there are not several hidden layers which processes the input step by step. Rather, the transformation of the hidden states are described by an ODE, such that they can be evaluated at any time/depth using an ODE solver. As often the ODE is specified by a neural network, CDNNs are also generally referred to as Neural ODEs, even though this was the name of a specific CDNN presented in [CRBD18]. Let us define a CDNN [MPP+20]:

$$\partial_t h = f_{\theta(t)}(t, h, s) \quad \text{with } h(t_0) = \hat{g}(s(t_0)) \text{ and } c(t) = \hat{f}(h(t)), \tag{1.1}$$

with h being the hidden states (or generally speaking the neurons), $s(t)$ the input, $\hat{g}$ and $\hat{f}$ input and output functions correspondingly, $c(t)$ the output, $\theta(t)$ the parameters of the

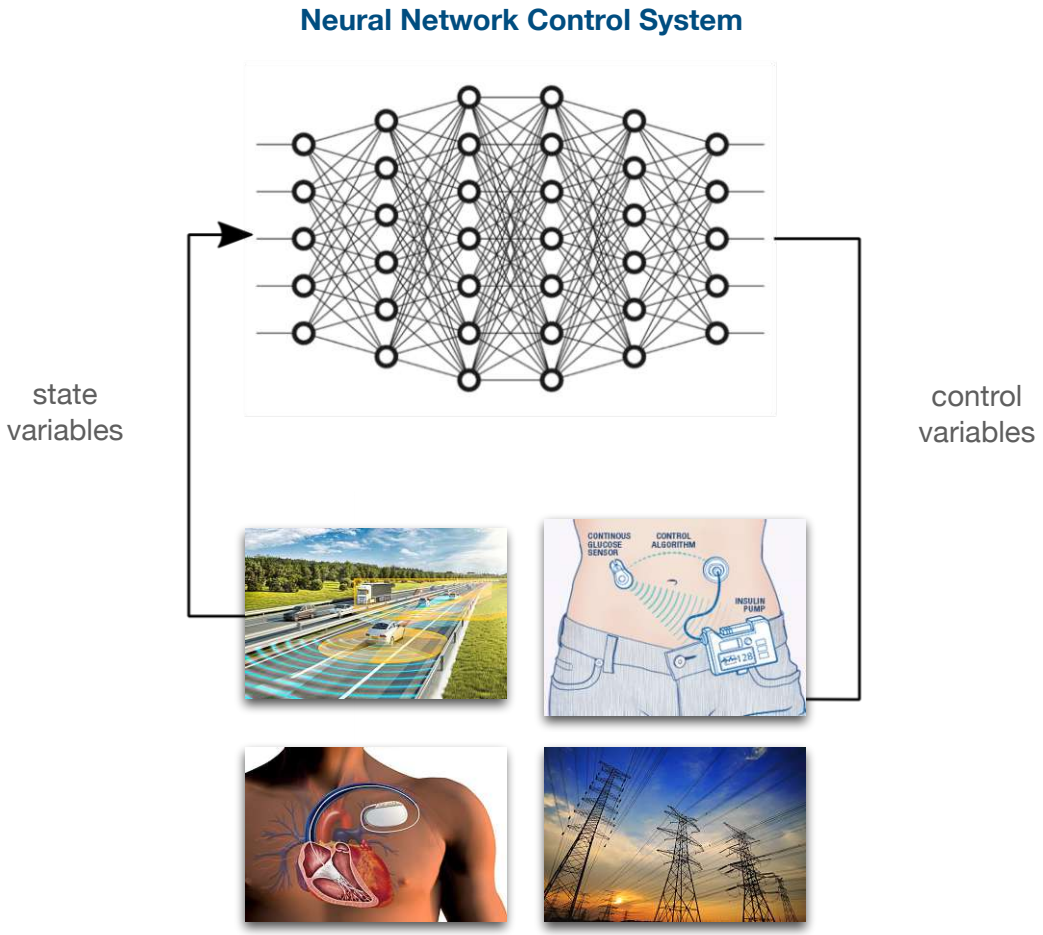


Figure 1.1: A Cyber-Physical System with a neural network controller is safe, if the system will not reach any unsafe state for all inputs and initial states.

CDNN. The derivative of the unknown hidden states h is described by a parameterized vector-valued function $\delta_t h = f_{\theta(t)}(t, h, s)$ which is often a neural network. This definition is a general formulation as CDNN is an umbrella term for different neural networks with ODEs, e.g. such as those introduced in [CRBD18] or the augmented version [DDT19].

In contrast to regular neural networks, where the input is passed through several hidden layers until the output layer, we integrate the ODE of the hidden states in Eq.(1.1) until a specific time T and evaluate the output $c(T) = \hat{f}(h(T))$. When using an integration timestep of $\Delta t = 1$ and Euler method as the solver, a CDNN can be interpreted as similar to a ResNet: every integration step is equivalent to $h + f(h)$, resembling one hidden layer in a ResNet. This is why these networks are considered *continuous-depth*. Another way to interpret CDNNs is as a continuous-time RNN: if we compare a CDNN with an RNN, it has continuous-time hidden state and can be evaluated at any desired time point T.

CDNNs have shown to be effective in handling irregularly sampled input data. In addition, they are particularly useful in control applications, as they can be used with varying control input timings after being trained, whereas regular neural networks would require distinct neural networks for different control input times. This reflects the ability of parameter sharing in CDNNs, allowing for more flexible and adaptive control in different situations, such as variable control input rates during specific maneuvers or turns.

Since all these networks are characterized by nonlinear ODEs, it is impossible, that is, undecidable, to exactly predict their behaviour, as they do not have a closed-form solution. This is very problematic because safety is an important concern in many of such systems, as for example, smart mobility, industry 4.0, or smart health-care. Robustness analysis of CDNNs, can be seen as a special case of reachability analysis of nonlinear ODEs, as it measures the ability to resist change in the input values. Fortunately, it is possible to approximate this behaviour but there is a big problem: traditional verification approaches for hybrid systems are not scalable enough to tackle the complex system dynamics that arises due to the use of a machine-learned model, such as a neural network. Deterministic verification approaches ensure conservative bounds [CÁS13, GDS+18, MGV18, BDPD+20, KMWZ20], but often sacrifice speed and accuracy [Ehl17]. Another big drawback is that, especially due to the wrapping effect caused by interval arithmetic, they suffer from scalability in space and in time; see CAPD, Flow*, Lagrangian Reachability (LRT), and LRT-NG in Fig. 1.2(a). Statistical methods, on the other hand, only ensure a weaker notion of conservativeness in the form of confidence intervals (statistical bounds). This, however, allows them to achieve much more accurate and faster verification algorithms that scale up to much larger dynamical systems [SZ15b, BS14]. However, to the best of our knowledge, there did not exist any statistical method for the robustness analysis of CDNNs.

The main problem is how to over-approximate the system dynamics, and thus the behaviour of the system, in as tight a way as possible, so that one can rely upon and use the huge potential of these CDNNs even in safety-critical systems, when it comes to difficult tasks. In principle the computation of the aforementioned over-approximations of the reachable states (also called a reachtube) is straightforward: one uses the Taylor expansion (in time) of the nonlinear ODEs, by replacing the set of initial states with a box (an interval in every dimension) and applying interval arithmetics to get the results (in interval arithmetics every function and relation is conservatively extended from points to boxes). In practice, however, this approach is too coarse: the result would be a far too wide reachtube (and thus over-conservative), giving us false positives when looking for intersections of the reachtube with unsafe regions. To avoid false positives, it is crucial to have as-tight-as-possible reachtubes. Otherwise they would e.g. predict that a car driven by an CDNN would cause a crash even if the CDNN is behaving perfectly and never causes a crash. Such reachtubes are thus not useful in evaluating CDNNs' safety.

This doctoral thesis is based on three peer-reviewed conference papers (see Fig. 1.3)

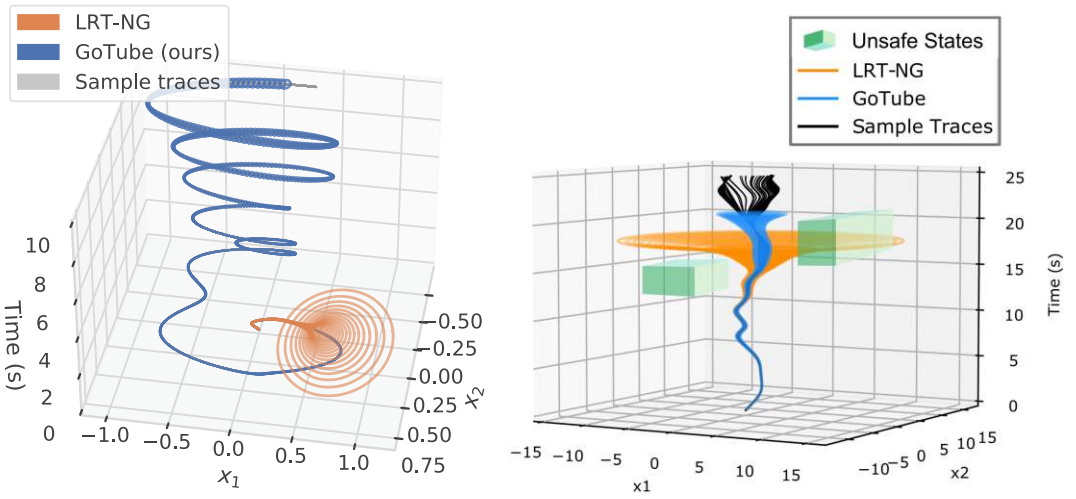


Figure 1.2: Reachtubes of LRT-NG [GCL+20] and GoTube [GLH+21]: (a) for a CT-RNN controlling Cart-Pole-v1 environment. LRT [CIB+17], CAPD [KMWZ20], and Flow* [CÁS13] failed on this benchmark. (b) For Dubins Car dynamical system. False positives are avoided by achieving tight reachtubes with GoTube compared to LRT-NG. The box representing unsafe states on the top right of the picture intersects with LRT-NG and yields a false positive, as the true sample traces are not inside that box.

which resulted in two tools: LRT-NG[1] and GoTube[2]. With the first conference paper, we improved the LRT theory dramatically and built our own conservative, set-based reachability tool and technique LRT-NG [GCL+20]. We used as comparison metrics the average volume of the constructed tubes as well as the maximum time horizon the tools were able to handle. Our results show that LRT-NG is very competitive with both Flow* and CAPD. Moreover, it is the only conservative tool able to handle the CDNNs. Flow* and CAPD either completely fail to handle these dynamical systems or fail after a very short time horizon (to the best of our knowledge at the time when this work was done). The second conference paper [GHL+21] presented a new non-conservative, statistical theory SLR, where reachability analysis is formulated as a global optimization problem, which uses interval arithmetics to compute local Lipschitz constants, a forward-mode gradient-descent algorithm for local search, and uniform sampling for global search. In our third conference paper [GLH+21], we presented our scalable statistical tool GoTube, where we introduce a Theorem to compute statistical bounds for the Lipschitz constant. In addition, GoTube is tensor-based and thus allows efficient (even multiple) GPU computation. We performed a diverse set of experiments and are able to show that GoTube substantially outperforms all state-of-the-art verification tools in terms of the size of the initial ball, time-horizon, task completion, and scalability.

While LRT-NG guarantees safety, SLR and GoTube only guarantee with a certain

[1] https://github.com/DatenVorsprung/LRTNG
[2] https://github.com/DatenVorsprung/GoTube

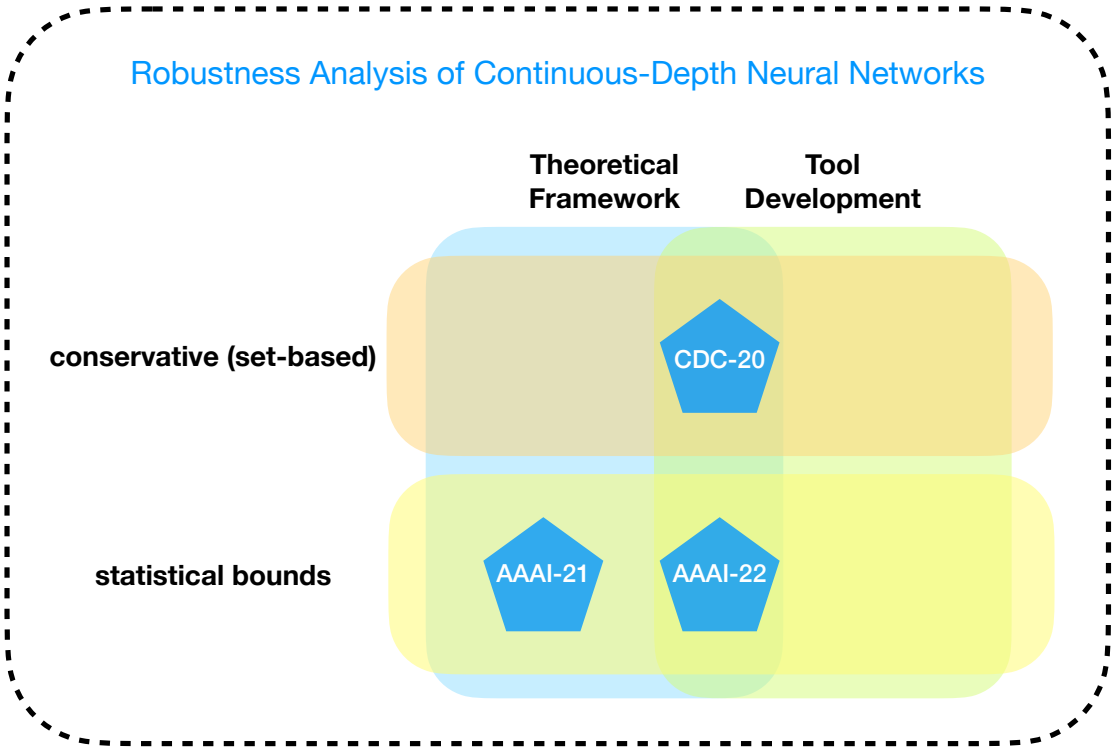


Figure 1.3: Areas of contribution of peer-reviewed conference papers on which this doctoral thesis is based.

confidence the safety of closed-loop CPS with CDNNs. The stochasticity of the latter two is only introduced through the algorithm, we are not looking at stochastic dynamical systems. The advantage of the statistical approach is to be able to predict the behaviour of a neural network control system in a much tighter and more scalable fashion, see Fig. 1.2(b). The price for it is a weaker assurance, for example that with a given probability (e.g. 99%) the reachable states of the CPS are inside the reachtube.

CHAPTER 2

Research Goals and Methodology

As described in Chapter 1, CPS with CDNN controllers were not yet ready for deployment in safety-critical systems. When using existing deterministic verification approaches, only an extremely short time-horizon could be verified. Statistical methods, on the other hand, were still not ready to be used in these complex nonlinear dynamical systems.

The research goal – was to provide safety-guarantees, a predictable behaviour and strong assurance even for high-dimensional neural network control systems, such that they can be used for complicated tasks in safety-critical environments. Therefore, the main motivation of my doctoral research project was to scale up reachability analysis of nonlinear ODEs to high-dimensional systems such as CDNNs.

The research questions – on which I focused in this thesis to get to the aforementioned goal, are the following:

1. *How to provide a stand-alone implementation of LRT that scales up to large systems of nonlinear ODEs?*

   When I started my PhD, the state-of-the-art technique for reachability analysis of nonlinear ODEs at TU Wien was LRT [CIB+17]. This was not able to scale up to high dimensional systems, not even to small neural network control systems. As LRT uses the over-approximation tool CAPD [KMWZ20, Zgl02] to handle the interval arithmetic calculations, our first goal was to remove that dependency.

   Methodology. This required us to replace CAPD routines with verified integration schemes [MB06, NJC99, Alt13], an approach also taken by other tools, including CAPD, CORA [AGK18] and Flow* [CÁS13]. Additionally, we took advantage of an improved Lohner's QR method [Loh92, NJC99] to account for the infamous wrapping effect, which is intrinsically connected to interval arithmetic [SB13].

2. *How to significantly improve LRT such that it can be applied even to Neural ODEs?*

After removing CAPD dependencies, we were able to focus on taming the infamous wrapping effect that occurs in every conservative, set-based over-approximation and computation as these prevent the desired scaling.

Methodology. We simulated and ploted the traces starting from different initial points as well as the reachtubes computed with LRT. After seeing where the biggest wrapping effect occurred, we developed new theorems combating that shortcoming. We ran experiments on a comprehensive set of benchmarks, including two Neural ODEs, to demonstrate LRT-NG's superior performance, when compared to state-of-the-art reachability tools such as LRT, CAPD, and Flow*.

3. *How to provide a theory of stochastic guarantees for much tighter and scalable reachtubes?*

   Improving LRT such that it could be applied to simple Neural ODEs, was not enough however, so it was necessary to develop the necessary mathematical foundations and theory for constructing tight reachtubes, and provide stochastic guarantees in the form of confidence intervals for the reachtubes' bounds.

   Methodology. We investigated statistical methods to find a good starting point for stochastic guarantees for reachtubes' bounds. By adapting these theories for our specific needs, we gave formal theorems and proofs. Finally, we introduced a theoretical framework for the verification of Neural ODEs by restating the reachability problem as a set of global-optimization problems.

4. *How to provide a statistical tool for robustness analysis that provides safety bounds up an arbitrary time horizon?*

   After developing the theoretical background for stochastic guarantees, we aimed at finding technical solutions for implementing and adapting the theory, such that it could be in fact applied for statistical verification. We wanted to introduce a practical statistical verification algorithm for continuous-depth neural networks.

   Methodology. We first implemented the theory of stochastic guarantees from Research-question 3 in form of the SLR algorithm. After observing and identifying the computational and implementation constraints resulting from the application of SLR, we proposed new technical solutions for addressing these fundamental issues, in the form of formal theorems and proofs. Finally, we introduced GoTube, a practical statistical verification algorithm for continuous-depth models.

Before talking about the contributions of this thesis, it is important to give an overview of its main research topics. Therefore, I will first clarify the state of the art in Chapter 3, then present the underlying background techniques in Chapter 4, before summarizing the scientific results and contributions of each paper in Chapter 5.

CHAPTER 3

State of the Art

A CPS consists of an environment, modeled as nonlinear ODEs (more generally as a Markov decision process), and a controller, modeled e.g., as an neural network controller (NNC), as seen in Fig. 1.1. No matter which specific NNC is chosen, one part of the CPS is defined by the environment which is a system of nonlinear ODEs in unknown $x \in \mathbb{R}^n$, where the field $f : \mathbb{R}^n \mapsto \mathbb{R}^n$ is assumed to be a sufficiently smooth (at least twice differentiable), time-invariant function:

$$\partial_t x = f(x), \quad x_0 = x(t_0). \tag{3.1}$$

Since time dependence can be incorporated by adding the auxiliary variable $\partial_t x_n = 1$, our discussion naturally extends to time-varying systems of the form $\partial_t x = f(t, x)$.

3.1 Reachability Analysis of Nonlinear ODEs

Linear ODEs posses a general closed-form solution, describing the behaviour of the solution-traces over time, for every initial state. Nonlinear ODEs however, have no closed-form solution anymore. One is able to calculate the solution for different initial states, but one does not know what happens between these already calculated traces.

The main goal of the reachability analysis of nonlinear ODEs, is to over-approximate the reachable states of the ODEs, starting from a set of initial states, symbolically represented for example as an interval, a ball, or an ellipsoid, in such a way that one can guarantee that all solution traces of the nonlinear ODE are inside the over-approximation. We call such an over-approximation a *reachtube*. Let us now define this mathematically:

Definition 1 (initial value problem (IVP)) *We have a time-invariant ordinary differential equation $\partial_t x = f(x), f : \mathbb{R}^n \to \mathbb{R}^n$, a set of initial values defined by a ball $\mathcal{B}_0 = B(x_0, \delta_0)$ with center $x_0 \in \mathbb{R}^n$ and radius $\delta_0 \in \mathbb{R}$, the initial condition $x(t_0) \in \mathcal{B}_0$*

and a sequence of k timesteps $\{t_j : j \in [0,\dots,k] \wedge (t_0 < t_1 < \cdots < t_k)\}$. *For every* t_j, *we want to know the solution* $x(t_j)$ *of*

$$\partial_t x = f(x), \quad x(t_0) \in \mathcal{B}_0 = B(x_0, \delta_0). \tag{3.2}$$

Let $\chi(t_j, x_0) = x(t_j)$ be the solution of Eq. (3.2) at time t_j, for $x(t_0) = x_0$. In reachability analysis, the goal is to find for every time step t_j an over-approximation $\mathcal{B}_j \supseteq \{\chi(t_j, x) : x \in \mathcal{B}_0\}$, such that the set of these over-approximations build up a reachtube, containing the reachable states.

Definition 2 (Reachtube) *Given a set of initial values* $\mathcal{B}_0 \in \mathbb{R}^{n\times n}$, *a nonlinear ODE as in Eq. (3.2), the pointwise solution function* $\chi(t_j, \cdot) : \mathbb{R}^n \to \mathbb{R}^n$ *and over-approximations* $\mathcal{B}_j \supseteq \{\chi(t_j, x) : x \in \mathcal{B}_0\}$. *The reachtube for a sequence of k timesteps* $\{t_j : j \in [0,\dots,k] \wedge t_0 < t_1 < \cdots < t_k)\}$ *is defined as*

$$\mathcal{R} = \{\mathcal{B}_0, \mathcal{B}_1, \dots, \mathcal{B}_k\}. \tag{3.3}$$

As we use balls and ellipsoids, we call the reachset $\mathcal{B}_j$ *bounding balls*. For every ellipsoid there is a metric in which the ellipsoid equals a ball. Let $M_j \in \mathbb{R}^{n\times n}$ be a positive definite matrix ($M_j \succ 0$). Then there exists a decomposition $A_j \in \mathbb{R}^{n\times n}$ with $A_j^\top A_j = M_j$. Every ellipsoid can be defined as $B_{M_j}(x_j, \delta_j) = \{x : \|x - x_j\|_{M_j} = \delta_j\}$ with center x_j, weighted radius δ_j and norm $\|x\|_{M_j} = \sqrt{x^\top M_j x} = \|A_j x\|_2$. If M_j is the identity matrix, then $\mathcal{B}_j$ is a ball in the Euclidean metric, so we will omit the subscript and use $B(x_j, \delta_j)$.

When using reachability analysis to check for intersections of the reachtube with regions of bad states, it is crucial to compute an as-tight-as-possible reachtube.

Robustness Analysis. Given the definition of reachability analysis, it is straightforward to see that robustness analysis is a special case of reachability analysis. Intuitively, the first computes how an initial perturbation δ_0 evolves over time, and gives guarantees about the maximum distance between the perturbed and unperturbed solutions for δ_0.

3.2 Reachability Analysis of CPS with Continuous-Depth Neural Networks

In the robustness analysis of a CPS with a CDNN controller as in Eq. (1.1), the nonlinear dynamics of the physical part of the CPS (referred to as the environment) is combined with the ODEs of the NNC In this thesis, we propose a combined ODE system for the CDNN controller and the environment as follows:

Definition 3 (CDNN Control System) *Let* $s \in S \subseteq \mathbb{R}^{n_s}$ *be the state variables of the environment,* $h \in H \subseteq \mathbb{R}^{n_h}$ *be the hidden states of a CDNN,* $\delta_t h = f_\theta(s, h)$ *be a*

Technique	Determ.	Parallel	Wrapping Effect	Arbitrary Time-horizon
LRT [CIB+17] with Infinitesimal strain theory	yes	no	yes	no
CAPD [KMWZ20] implements Lohner algorithm	yes	no	yes	no
Flow* [CÁS13] with Taylor models	yes	no	yes	no
δ-reachability [GKC13] with approximate satisfiability	yes	no	yes	no
C2E2 [DMVP15] with discrepancy functions	yes	no	yes	no
LDFM [FKJM17] by simulation, matrix measures	yes	yes	no	no
TIRA [MDA19] with second-order sensitivity	yes	yes	no	no
Isabelle/HOL [Imm15] with proof-assistant	yes	no	yes	no
Breach [Don10, DM07] by simulation	yes	yes	no	no
PIRK [DKAZ20] with contraction bounds	yes	yes	no	no
HR [LBB20] with hybridization	yes	no	yes	no
ProbReach [SZ15a] with δ-reachability,	no	no	yes	no
VSPODE [ES11] using p-boxes	no	no	yes	no
Gaussian process (GP) [BS14]	no	no	no	no
LRT-NG [GCL+20] (part of this thesis)	yes	no	yes	no
SLR [GHL+21] (part of this thesis)	no	yes	no	no
GoTube [GLH+21] (part of this thesis)	no	yes	no	**yes**

Table 3.1: Related work on the reachability analysis of continuous-time systems. Determ.= Deterministic. "No" indicates a stochastic method.

parameterized vector-valued function describing the derivatives of the hidden states h, $c(t) = \hat{f}(h(t)) \in C \subseteq \mathbb{R}^{n_c}$ represent the control input variables as a function of the hidden states, $\delta_t s = g_c(s,c) = g_c(s,\hat{f}(h)) = g(s,h)$ be a vector field which defines the environment as a function of the state and control variables and $h(t_0) = \hat{g}(s(t_0))$ be the input function from the environment to the CDNN. Then an NNC system is represented as a nonlinear ODE that combines the environment with the CDNN:

$$x = \begin{pmatrix} s \\ h \end{pmatrix}, \quad \delta_t x = f(x) = \begin{pmatrix} f_\theta(x) \\ g(x) \end{pmatrix}, \quad x_0 = x(t_0) = \begin{pmatrix} s(t_0) \\ \hat{g}(s(t_0)) \end{pmatrix} \in \mathcal{B}_0, \tag{3.4}$$

where f is assumed to be Lipschitz-continuous and forward-complete.

Hence, a CDNN control system can be seen as a special case of a high-dimensional system of nonlinear ODEs as given by Eq. (3.1).

Solving this equation solely by Taylor expansion is not feasible due to the fact that the initial value x_0 is not a single real number but a set of initial states. Therefore, we need to consider alternative methods to avoid blow-up in space of the over-approximations when running the algorithm for a longer time horizon.

We start by providing a summary of methods developed for the reachability analysis of nonlinear ODEs in Table 3.1. A fundamental shortcoming of the majority of the methods described in Table 3.1 is their lack of scalability while providing conservative bounds.

Interval-Arithmetic-Based Approaches. The set of initial states can be represented as a multi-dimensional interval box. When using intervals instead of reals and applying

Taylor expansion, the arguably best approach to combat the wrapping effect is Lohner's method, which wraps the linear image of a box with a box aligned with its largest side [Loh92]. The state-of-the-art tool CAPD [KMWZ20, Zgl02, WZ12] employs this method when integrating the nonlinear ODEs and additionally integrates the associated variational equations to improve the tightness of the reachtube by also exploiting the sensitivity of the ODEs with respect to their initial states.

Taylor-Models-Based Approaches. Instead of working with intervals directly to describe the set of reachable states (reachset) in the space dimension, it is also possible to use Taylor models, which represent intervals as a pair, consisting of a symbolic part (a Taylor expansion expression), and a remainder part (an interval). The symbolic part supports the parametric definition of the initial states (e.g. $(1 + a, 1 - a)$, for a ranging in a particular interval), and consequently, the higher order terms can encode non-convex sets. This is the so-called expansion in space [NJN07]. Nevertheless it is still necessary to work with intervals to conservatively express the remainder part. This approach was first used in COSY-Infinity [MB03], and then in Flow* [CÁS13]. As a straightforward application of Taylor models would be also too coarse, similar results as with Lohner's method in interval arithmetic are achieved by using shrink-wrapping and preconditioning.

Bloating-based Techniques. They avoid the explicit propagation of the reachset by conservatively bloating each state of an execution starting in an initial state, to a ball in some appropriate metric [MA15, FKJM17]. By bloating we mean increasing the diameter of the current reachset overestimate (e.g. a ball), with the goal of conservatively bounding all reachable states for a given time. Discrepancy-function techniques (tool C2E2) compute the bloating by over-approximating the ODE solution with an exponential function, whose time constant is the interval approximation of the ODE's Jacobian in a metric computed through semidefinite programming [FM15, FKJM16, FKJM17]. Lagrangian techniques (tool LRT) compute the bloating numerically, by integrating the variational ODEs (capturing system's sensitivity) with interval arithmetic, and computing the Jacobian's metric either analytically [CIS+18, GCI+19] or by using semidefinite programming [CIB+17]. The key observation of LRT is that interval arithmetic should be used with considerable care, and only when necessary, due to its inevitable blow up.

Statistical Methods. Bortolussi et al [BS14] developed a method to construct reachtubes which is not exact and not conservative, but provides statistical guarantees. Their method is simulation-based, such that they only consider a finite number of sample traces. Using non-parametric Bayesian methods they were able to statistically control the error and to produce a statistical over-approximation. With their approach they construct reachtubes and prove that the true reachable set is inside that reachtubes with a given confidence. In [BCP+19] the focus is on neural predictive monitoring, where they also compute statistically sound estimates of uncertainty. As the conservative methods make use of the sensitivity analysis, Oakley et al [OO04] presented a scalable probabilistic sensitivity analysis with a Bayesian approach. In this case instead of intervals describing

the initial set, they use a distribution describing the uncertainty in the input and then they analyse the uncertainty in the produced output.

3.3 Reachability Analysis of CPS with Feedforward Neural Networks

Some state-of-the-art methods are limited to feedforward neural networks, or rather specific activation functions, which I also want to mention here for the sake of completeness.

ReLu-Based Approaches. Dutta et al [DCS19] developed an approach to construct reachtubes for CPS controlled by NNCs using ReLU activation functions, represented by piecewise linear functions. They use Flow* to over-approximate the dynamics of the environment. Instead of directly over-approximating the output of the neural network, they produce a "local" Taylor model for that part, which can be integrated into Flow*, thus constructing a reachtube for the whole system using that tool. More precisely, they replace the neural network feedback law for a small subset of inputs, by a polynomial mapping using regression. In addition they compute an error interval which conservatively covers the difference between the polynomial function and the real neural network. Similar to Taylor models, this yields a polynomial function plus an error interval. The computation of the error interval can be solved as a mixed integer nonlinear optimization problem.

Sigmoid-Based Approaches. Ivanov et al [IWA+18, ICW+20, ICW+21] focused on the reachability analysis of CPS using NNCs with sigmoids or tanh as activation functions. They exploited the property of a sigmoid that it is the solution to a quadratic differential equation. In their tool Verisig, they converted the dynamical system, consisting of the controller and the environment, into an equivalent hybrid system. This conversion was achieved by replacing the neurons in each layer with ODEs, and by converting each layer into a mode of the hybrid system. Finally, they combined the nonlinear dynamical system of the environment with the hybrid system of the controller, and used Flow* to construct the reachtubes of the closed-loop CPS.

Bernstein-Polynomials-Based Approaches. ReachNN [HFL+19] and the corresponding tool ReachNN* [FHC+20] are not limited to the activation function of the neural network, as long as the activation function is Lipschitz continuous. However, ReachNN* only considers feedforward neural networks. The authors abstract the NNC as Bernstein Polynomials for a small subset of inputs and then over-approximate the error bound between the real output and the one of the polynomials, thus creating Taylor models. The error can be either computed conservatively using the Lipschitz constant of the neural network, or with a sampling-based estimation error.

Star-Sets-Based Approaches. NNV [TBXJ20] verifies convolutional neural networks with a set-based method by first introducing a new type of set called ImageStar, and then

by showing that they can compute tighter over-approximations than other approaches using zonotopes or polytopes, by taking advantage of ImageStar.

Statistical Methods. Weng et al [WCN+18] did not focus on CPS using NNC but on probabilistic robustness of neural networks. They assume that the input noises are either zero-mean Gaussian or independent bounded random noises. Their tool is able to certify the probability that the classifiers top prediction cannot be altered. As expected, they can improve robustness certification by up to 75% compared to the worst-case approaches, with 99.99% confidence. Ruan et al [RWS+19] propose an optimization-based approach to compute robustness of convolutional neural networks, for the Hamming distance of images. As a sort of inverse reachability analysis, they compute the maximum radius of the initial ball, within which there are no adversarial examples for a trained network.

CHAPTER 4

Background

4.1 Interval Arithmetic and Lohner Method

There are different ways to define conservative regions by set representations: intervals, balls, ellipsoids, polytopes and more. In the papers [GCI+19, GCL+20] we relied on interval arithmetic, so we want to shortly review the benefits and problems with that method. The set of intervals on the real numbers is defined as ([NJC99]):

$$\mathbb{IR} = \{[a] = [\underline{a}, \overline{a}] : \underline{a}, \overline{a} \in \mathbb{R}, \underline{a} \leq \overline{a}\}, \tag{4.1}$$

whereas an *interval vector* $[x] \in \mathbb{IR}^n$ is a vector with interval components and an *interval matrix* $[A] \in \mathbb{IR}^{n \times m}$ is a matrix with interval components. The biggest problem in interval arithmetic is the wrapping effect, which happens if we apply concatenated functions on intervals (see Fig. 4.1). In the LRT algorithms [GCI+19, GCL+20] an improved version of Lohner's QR method [Loh92, NJC99] is used to directly address the wrapping effect

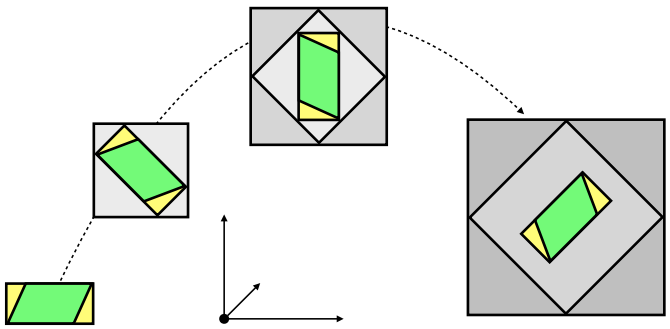


Figure 4.1: *Wrapping effect.* Symbolic illustration for wrapping of a parallelogram (green) when applying a consecutive rotation of 45° to it with interval boxes (grey) and with interval boxes in adapted coordinate systems using Lohner's QR method (yellow).

caused by interval arithmetic. Intuitively, the rotational part Q of a function evaluation is extracted, which is subsequently used as a new coordinate system. For a more detailed discussion of the steps mentioned above, please refer to [GCI+19].

4.2 Lipschitz Constant and the Variational Equation

The Lipschitz constant defines a relation between the domain and the range of a function, more precisely it bounds the distance in the range by a multiple of the distance in the domain. Intuitively, if we know the Lipschitz constant of a dynamical system, we are able to bound the maximum distance between trajectories at a given time, having the distance of their initial starting points.

Definition 4 (Lipschitz Constant) *Let $f\colon A \to \mathbb{R}^m$ $(A \subseteq \mathbb{R}^n)$ be a function, $M_A, M_B \succ 0$ metrics on the domain and range, respectively, $S \subseteq A$ a subset of the domain. Then:*

$$\lambda_S = \sup_{x,y \in S, x \neq y} \frac{\|f(x) - f(y)\|_{M_B}}{\|x - y\|_{M_A}} \tag{4.2}$$

is called the Lipschitz constant of f on set S.

An upper bound of the Lipschitz constant can be computed using the mean value theorem from calculus with the statement for vector valued functions:

Theorem 1 (Mean value theorem (generalized Rolle's theorem)) *Let $M_1, M_2 \succ 0$ be respectively metrics on the domain and the range with $M_1 = A_1^\top A_1, M_2 = A_2^\top A_2$ and norm $\|x\|_{M_{1,2}} = \|A_2 x A_1^{-1}\|_2$. Considering the change of metric [CIB+17, Lemma 2] and the well-known mean value theorems, we are able to make the following statement:*

Let $\chi\colon A \to \mathbb{R}^n$ $(A \subseteq \mathbb{R}^n)$ be a vector-valued function, $S \subseteq A$ a subset of the domain and the norm of the Jacobian matrix of χ be bounded by some constant $\Lambda_{1,2} \geq \|\partial_x \chi(x + h \cdot (y - x))\|_{M_{1,2}}$ for all $h \in [0, 1]$ and all $x, y \in S$. Then it holds:

$$\|\chi(x) - \chi(y)\|_{M_2} \leq \Lambda_{1,2} \cdot \|x - y\|_{M_1} \quad \forall x, y \in S, \tag{4.3}$$

and thus $\Lambda_{1,2}$ is an upper bound of the Lipschitz constant λ_S of Definition 4.

The mean value theorem can be used to find an upper bound of the local Lipschitz constant. We will need such an upper bound for both deterministic and statistical guarantees. In both cases we need the Jacobian matrix of the solution function $\chi(t_j, \cdot)$ of Eq. (3.2), so the question is how to compute the Jacobian matrix for the solution of a differential equation, for which we do not even have a closed form solution?

To this end, we introduce $F_x : \mathbb{R} \to \mathbb{R}^{n \times n}$ with $F_x(t) = \partial_x \chi(t, x)$ called the *deformation gradient* in [Sla02, Abe98], and *sensitivity* in [Don10, DM07]. $F_x(t)$ describes how sensitive the solution to the IVP at time t is to an infinitesimal small perturbation in the initial value x. To compute $F_x(t)$ we make use of the variational equation, which intuitively describes how an initial perturbation in the initial value evolves over time.

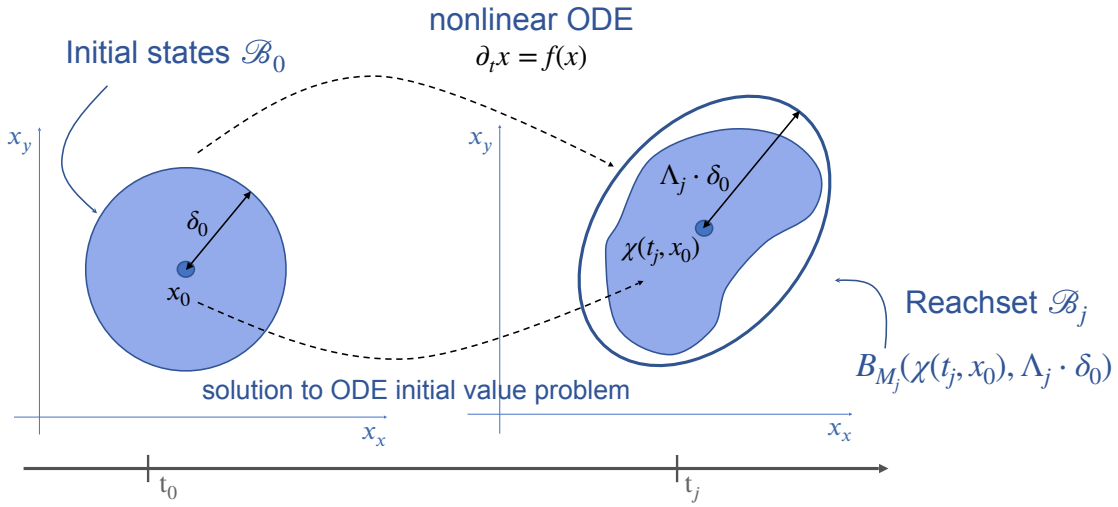


Figure 4.2: A graphical overview of the underlying algorithm of LRT. The figure shows the over-approximation of one timestep.

Definition 5 (Variational equation) *Let f be the system equations of the initial value problem defined in Eq. (3.2) and $\chi(t, x_0)$ be the solution at time t for $x(t_0) = x_0$, then the following equation is called the variational equation:*

$$\partial_t F(t) = (\partial_x f)(\chi(t, x))F(t), \quad F(t_0) = I, \tag{4.4}$$

with $I \in \mathbb{R}^{n \times n}$ is the identity matrix.

In [GCL+20] it was shown that F_x is a solution of the *variational equations* associated to the system equations in Eq. (3.2). Thus, we compute $F_x(t)$ for different initial points x by solving the IVP of Eq. (4.4) for these points.

4.3 Lagrangian Reachability (LRT)

The most straightforward way to compute a conservative reachtube as defined in Def. 2, would be to use an interval enclosure $[\mathcal{X}_0] \supseteq \mathcal{B}_0$ of the initial values and just use interval-arithmetic evaluations of an integration method, for example, the Runge-Kutta method, to propagate them from timestep to timestep.

However, due to the infamous wrapping effect (as shown in Fig. 4.1), this approach would lead very soon to a blow-up in space. As we already mentioned in the related work sections in Chapter 3, there are various ways on how to avoid that blow-up in space and create as tight as possible reachtubes.

LRT is a bloating based technique. As shown in Fig. 4.2, starting with an initial ball $\mathcal{B}_0 = B(x_0, \delta_0)$, at a sequence of k timesteps $\{t_j : j \in [0, \ldots, k] \wedge (t_0 < t_1 < \cdots < t_k)\}$, it propagates the center of the ball $x(t_j) = \chi(t_j, x_0)$, and computes the new radius δ_j,

by multiplying δ_0 with a stretching factor Λ_j. In addition, LRT claims to compute an optimal metric M_j of the ball at time t_j, to make sure that the reachsets are ellipsoids of an optimal shape. So the question is, how to compute a stretching factor Λ_j and a metric M_j such that $\mathcal{B}_j = B_{M_j}(\chi(t_j, x_0), \Lambda_j \cdot \delta_0)$ is a tight over-approximation of the reachable set of states at time t_j.

With $\chi(t_j, x_0)$ being the solution of Eq. (3.1) at time t_j, for $x(t_0) = x_0$ and using Thm. 1 it holds that:

$$\max_{x \in \mathcal{B}_0} \|\chi(t_j, x) - \chi(t_j, x_0)\|_{M_j} \leq \max_{x \in \mathcal{B}_0} \|F_x(t_j)\|_{M_j} \max_{x \in \mathcal{B}_0} \|x - x_0\|_{M_0}. \tag{4.5}$$

Interval arithmetic is used to over-approximate $\max_{x \in \mathcal{B}_0} \|F_x(t_j)\|_{M_j}$ by propagating all possible deformation gradients as an interval $[\mathcal{F}_j] \supseteq \{F_x(t_j) : x \in \mathcal{B}_0\}$ with an interval arithmetic version of the variational equation Eq. (4.4):

$$\partial_t[\mathcal{F}] = (\partial_x f)([\mathcal{X}_j])[\mathcal{F}], \quad \mathcal{F}_0 = [I] \tag{4.6}$$

where $[\mathcal{X}_j]$ is an as-tight-as-possible interval over-approximation of $\mathcal{B}_j$. Thus the challenge is to bound the norm of the interval deformation gradients:

$$\|[\mathcal{F}_j]\|_{M_j} \leq \Lambda_j \Rightarrow \delta_j = \Lambda_j \delta_0. \tag{4.7}$$

Λ_j - the upper bound of the Lipschitz constant of $\chi(t_j, \cdot)$ - is called the *stretching factor (SF)* associated to the interval gradient tensor, as it shows by how much the initial ball $\mathcal{B}_0$ has to be stretched, such that it encloses the set of all reachable states. Having the interval gradient $[\mathcal{F}_j]$ at time t_j, Eq. (4.7) is solved using algorithms from [HDT10, Rum01, Roh98], and choosing the tightest result available. The correctness of LRT is rooted in [CIB+17, Theorem 1].

The metric M_j is chosen by solving the following optimization problem:

$$M_j = \arg\min_{M \succ 0} \|[\mathcal{F}_j]\|_M. \tag{4.8}$$

In [CIS+18], they developed a simple explicit analytical formula for finding M_j in Eq. (4.8). But in fact, as shown in our LRT-NG paper [GCL+20], this was a correct result for the wrong optimization problem.

As the tightness of the bounding balls $\mathcal{B}_j$ depends on the previous values, for example $\mathcal{B}_{j-1}, [\mathcal{X}_j]$ or $[\mathcal{F}_j]$, the wrapping deficiencies accumulate in time, as shown in Fig. 4.1.

CHAPTER 5

Summary of Scientific Results

5.1 Lagrangian Reachtubes: The Next Generation

This section presents the theoretical advances of LRT-NG in [GCL+20] which concentrates on minimizing the volume of the bounding balls and their enclosure, and thus on creating tighter and longer reachtubes. In particular, we first state the optimization problem to be solved in order to get the optimal metric, and thus the bounding ball with minimal volume. We first describe an analytic solution of an optimal metric minimizing the volume of the ellipsoid and prove that it solves the optimization. Finally, we focus on the new reachset box $[\mathcal{X}_j]$ computation, the interval over-approximation of the ellipsoid-ball-intersection.

As shown in Algorithm 1, LRT-NG iterates over the sequence of k timesteps, until it reaches the given time horizon T. After propagating the center point, it computes the interval deformation gradient by integrating Eq. (4.6) in Line 4. After computing the optimal metric M_j, it bounds the maximum singular value of $[\mathcal{F}_j]$ in both the Euclidean norm and M_j norm, such that LRT-NG constructs an ellipsoid $\mathcal{B}_j$ and an Euclidean ball $\mathcal{B}_j^{circle}$. This allows us to define an as-tight-as-possible interval box $[\mathcal{X}_j]$, as the intersection of the ellipsoid and the ball. This intersection-based approach considerably reduces the wrapping effect of the next integration of the interval variational equation.

5.1.1 Computation of the Metric

To obtain an as-tight-as-possible over-approximation, we minimize the volume of the n-dimensional ball $\mathcal{B}_j = B_{M_j}(x_j, \delta_j)$. Hence, the optimization problem is given by:

$$\arg\min_{M_j \succ 0} \mathrm{Vol}(B_{M_j}(x_j, \delta_j)), \tag{5.1}$$

where $\delta_j = \Lambda_j(M_j) \cdot \delta_0$. Let us further define $\hat{F}_{j-1,j} = \partial_x \chi_{t_{j-1}}^{t_j}(x)|_{x=x_{j-1}}$ as the deformation gradient from time t_{j-1} to t_j at the center of the ball. Using the chain rule it holds that

Algorithm 1 LRT-NG

Require: initial ball $\mathcal{B}_0 = B(x_0, \delta_0)$, initial metric M_0, initial metric decomposition A_0 ($M_0 = A_0^\top A_0$), time horizon T, sequence of timesteps t_j ($t_0 < \ldots < t_k = T$), system dynamics f

set $[\mathcal{F}] \leftarrow \{I\}, [\mathcal{X}] \leftarrow$ over-approximation of $\mathcal{B}_0$
for $(j = 1; j \leq k; j = j + 1)$ **do**
  $x_j \leftarrow solveIVP(f, x_{j-1}, [t_{j-1}, t_j])$
  $[\mathcal{F}] \leftarrow F_{[\mathcal{X}_0]}(t_j) = rungeKuttaVariational((\partial_x f)([\mathcal{X}]), [\mathcal{F}], [t_{j-1}, t_j]))$
  $M_j \leftarrow computeOptimalMetric(F_{x_j}(t_j), A_0)$
  for all $M \in \{M_j, I\}$ **do**
    compute $\Lambda \geq \|[\mathcal{F}]\|_M$ (stretching factor)
  end for
  $\mathcal{B}_j \leftarrow B_{M_j}(x_j, \delta_{M_j})$
  $\mathcal{B}_j^{circle} \leftarrow B(x_j, \delta_I)$
  $[\mathcal{X}] \leftarrow intersectionBox(\mathcal{B}_j, \mathcal{B}_j^{circle})$
end for
return $(\mathcal{B}_1, \ldots, \mathcal{B}_k), (\mathcal{B}_1^{circle}, \ldots, \mathcal{B}_k^{circle})$

$F_j = \prod_{m=1}^{j} \hat{F}_{m-1,m}$, where F_j is defined as the deformation gradient at x_0. The following theorem defines a metric $\hat{M}_j$ and shows that this metric minimizes the ellipsoid volume, and is therefore optimal.

Theorem 2 (Thm. 1 in [GCL+20]) *Let the gradient-of-the-flow matrices $\hat{F}_{j-1,j} \in \mathbb{R}^{n \times n}$ and F_j be full rank, and the coordinate-system matrix of the last time-step $A_{j-1} \in \mathbb{R}^{n \times n}$ be full-rank and $A_{j-1} \succ 0$. Define metric $\hat{M}_j(F_j) = \hat{A}_j(F_j)^\top \hat{A}_j(F_j)$, where:*

$$\hat{A}_j(F_j) = A_{j-1} \hat{F}_{j-1,j}^{-1} = A_0 F_j^{-1} \tag{5.2}$$

When F_j is known, we simply abbreviate $\hat{A}_j(F_j)$ with $\hat{A}_j$, and $\hat{M}_j(F_j)$ with $\hat{M}_j$. Let $\Lambda_{0,j}(M_j)$ be given by (with M_0 fixed):

$$\Lambda_{0,j}(M_j) = \sqrt{\lambda_{\max}\left((A_0^\top)^{-1} F_j^\top M_j F_j A_0^{-1}\right)}.$$

Then, it holds that $\mathrm{Vol}\left(B_{\hat{M}_j}(\chi(t_j, x_0), \Lambda_{0,j}(\hat{M}_j)\, \delta_0)\right)$ *is equal to:*

$$\min_{M_j \succ 0} \mathrm{Vol}\left(B_{M_j}(\chi(t_j, x_0), \Lambda_{0,j}(M_j)\, \delta_0)\right).$$

In other words, the symmetric matrix $\hat{M}_j \succ 0$ minimizes the volume of the ellipsoid $B_{M_j}(\chi(t_j, x_0), \Lambda_{0,j}(M_j)\, \delta_0)$ as a function of M_j.

Thus, Thm. 2 gives us an analytic solution for the optimal metric, releasing us from either solving an optimization problem with semi-definite programming in every time-step

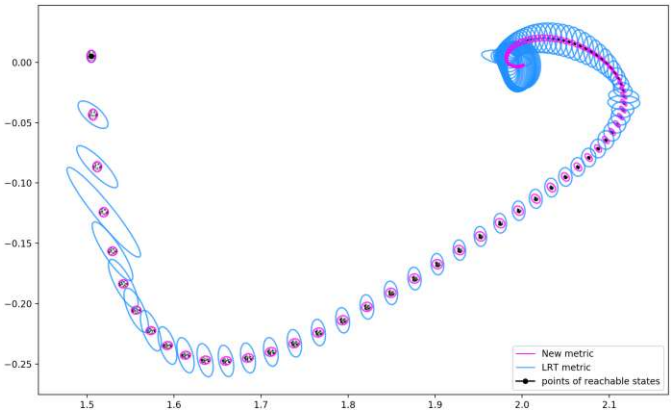


Figure 5.1: Reachtube for the Robotarm model, obtained with the LRT (in blue) and LRT-NG metric (in purple), respectively. The small, black dots represent points of trajectories starting in the set initial states. The time is bounded to the interval $t \in [0, 5]$, and it evolves starting at the top left corner of the figure, and going to the right.

like in [FKJM17, CIB+17], or risking false-positive consequences of using a suboptimal metric as in LRT [CIS+18, GCI+19].

A comparison of the reachsets obtained with LRT metric $\tilde{M}_j$ [CIS+18, Definition 1] and the one obtained with LRT-NG metric $\hat{M}_j$ is illustrated in Fig. 5.1. It shows that the LRT metric is by far not an optimal choice, and it also shows how well our new analytically computed metric $\hat{M}_j$ follows the shape of the set of reachable states.

5.1.2 Intersection of the Bounding Balls

Another novelty in LRT-NG, is that the next reachset is the intersection of an ellipsoid computed in the optimal metric and an Euclidean ball. This considerably reduces the volume and therefore enables LRT-NG to work also for CDNNs.

An effective way of getting a much tighter conservative bound $[\mathcal{X}_j]$ is taking the intersection of the ellipsoid in the optimal metric $\hat{M}_j$, and the ball in Euclidean metric. As small errors accumulate in interval arithmetic, taking the intersection leads to a considerable improvement especially as the time horizon increases. This new approach is conservative as shown in Lemma 1 of [GCL+20], which allows us to dramatically reduce the volume of the reachtube and combat the wrapping effect in a way that has not been considered before by bloating-based techniques [CIB+17, CIS+18, GCI+19, FKJM17, FKJM16, FM15].

5.1.3 Theoretical Contributions

The theory of this paper answers our Research-Questions 1 and 2 from Chapter 2. We present a stand-alone tool LRT-NG, whose underlying theory and algorithm significantly improved LRT and can be even applied to Neural ODEs. In summary, our theoretical contributions resulting in much tighter reachtubes are as follows:

Table 5.1: Performance comparison with Flow*, CAPD and LRT. We use following labels for models M: B(2)- Brusselator, V(2)- Van der Pol oscillator, R(4)- Robotarm, D(3)- Dubins Car, M(2)- Mitchell Schaeffer cardiac-cell, C(4)- controlled cartpole, Q(17)-Quadcopter, C-N(12)- cartpole wt. Neural ODEs, C-L(12)- cartpole wt. LTC RNN (number in parenthese denotes dimension). T: time horizon, dt: time step, r: initial radius in each dimension, AV: average volume of reachtubes in T, Fail: Volume blow-up before T; (1), (2) and (4) denotes the integration order. We mark in bold the best performers for low orders (1,2nd) and the higher order (4th).

M	dt	T	r	AV			
				LRT-NG	Flow*	CAPD	LRT
B(2)	0.01	9	0.01	**1.5e-4 (1)** 1.4e-4 (2, 4)	5.1e-3 (2) **9.8e-5 (4)**	4.3e-4 (2) 3.6e-4 (4)	6.7e-4 (1) 6.1e-4 (2, 4)
V(2)	0.01	40	0.01	**4.2e-4 (1)** 4.1e-4 (2, 4)	5.6e-3 (2) **3.5e-4 (4)**	1.5e-3 (2, 4)	4.1e-3 (1) 3.7e-3 (2, 4)
R(4)	0.01	40	0.005	8.1e-11 (1) **8e-11 (2)** **7.9e-11 (4)**	1.1e-9 (2) 8.7e-10 (4)	1.1e-9 (2, 4)	Fail
D(3)	0.00125	15	0.01	0.1323 (1) 0.1312 (2, 4)	6.6037 (2) **4.5e-2 (4)**	**0.1181 (2, 4)**	390 (1) 385 (2, 4)
M(2)	0.01	10	10^{-4}	**3.8e-9 (1, 2)** **3.7e-9 (4)**	3.9e-8 (2) 1.5e-8 (4)	4.9e-8 (2) 4.4e-8 (4)	3.2e-8 (1, 2, 4)
C(4)	0.001	10	10^{-4}	**8.4e-17 (1)** **7.2e-17 (2,4)**	1.1e-11 (2) 7e-13(4)	2.6e-13 (2) 2.6e-13 (4)	Fail
Q(17)	10^{-4}	2	0.005	**3.21e-54 (1)** **9.31e-56 (2)**	9.7e-25 (2)	1.7e-31 (2)	Fail
C-N(12)	10^{-5}	1	10^{-4}	**3.9e-27 (1, 2)**	Fail	Fail	Fail
C-L(12)	10^{-6}	0.35	10^{-4}	**4.49e-33 (1)**	Fail	Fail	Fail

Metric computation. We introduce a new analytic method for computing the next-ball (next bloating) metric, and prove that by using this metric we minimize the volume of the resulting next-ellipsoid (Cartesian bloating).

Reachset computation. We considerably reduce the wrapping effect in the computation of the next reachset by intersecting the ball resulting in the Cartesian metric with the ellipsoid computed in the optimal metric.

Center propagation. We show how to conservatively propagate the bloating-center states, without incurring the infamous wrapping effect due to the interval-arithmetic

propagation of boxes (multi-dimensional intervals).

5.1.4 Experimental Evaluation

To assess the performance of LRT-NG in terms of accuracy and speed, we applied it to a comprehensive set of available nonlinear ODE benchmarks, and to two Neural ODEs we developed ourselves. The benchmarks are as follows: Brusselator, Van der Pol Oscillator, Robotarm, Dubins Car, Mitchell-Schaeffer Cardiac-cell, linearly controlled Cartpole, Quadcopter, Cartpole controlled with a Neural ODEs [LZLD18], and a Cartpole controlled with an LTC [LHZ+19]. For comparison, we applied LRT and the latest versions of Flow* and CAPD to these benchmarks, too, and show the results in Table 5.1. LRT-NG is the only tool able to compute reachtubes for the Neural ODEs benchmarks (to the best of our knowledge at the time when this work was done).

5.2 On the Verification of Neural ODEs with Stochastic Guarantees

In order to avoid the bounding-balls blow up, as it happens in the conservative methods, we developed a statistical version of LRT. This technique provides convergence guarantees for computing the upper bound of the confidence interval, for the maximum perturbation at time t_j with confidence level $1-\gamma$ and tube tightness μ.

We first review Stochastic Lagrangian Reachability (SLR), a purely theoretical statistical version of LRT framework [GHL+21], and then GoTube, a practical statistical verification algorithm for continuous-depth models [GLH+21], where we achieved technical solutions for fundamental problems occurring when applying SLR.

We describe *reachability as an optimization problem* and solve that problem for every timestep such that the size of the bounding ball $\mathcal{B}_j$ at time t_j does not depend on the previous values $\mathcal{B}_{j-1}, [\mathcal{X}_j]$ or $[\mathcal{F}_j]$ like in LRT-NG. To compute a bounding reachtube, we have to compute at every time step t_j, the maximum perturbation δ_j in metric M_j for $x \in \mathcal{B}_0$, which is defined as the solution of the optimization problem:

$$\delta_j \geq \max_{x \in \mathcal{B}_0} \|\chi(t_j, x) - \chi(t_j, x_0)\|_{M_j} = \max_{x \in \mathcal{B}_0} d(\chi(t_j, x)) = m^\star \tag{5.3}$$

where $d_j(x) = d(\chi(t_j, x))$ denotes the *distance* at time t_j from the center $\chi(t_j, x_0)$, if the initial center x_0 and metric M_j is known from the context.

As we require Lipschitz-continuity and forward-completeness of the CDNN in Eq. (3.4), 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. (5.3) 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.

In order to be able to optimize this problem, we describe the points on the surface with (n-dimensional) polar coordinates such that every point $x \in \mathcal{B}_0^S$ is represented by a

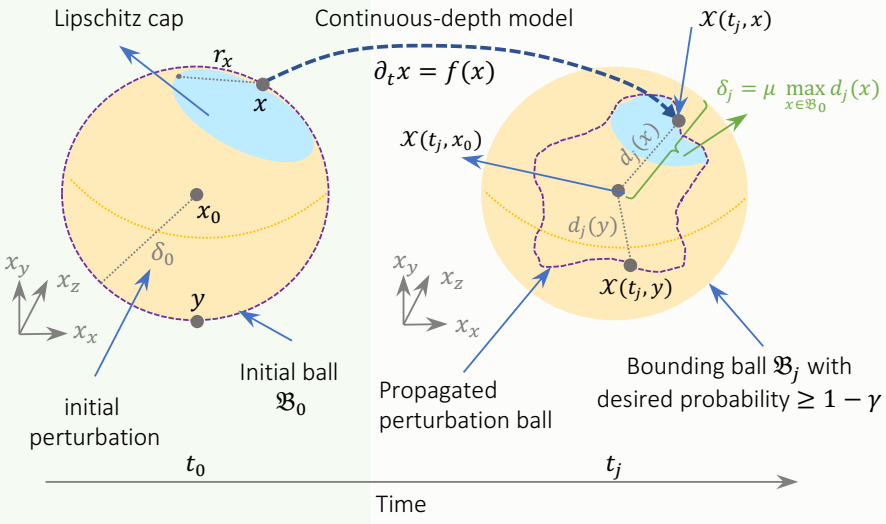


Figure 5.2: Statistical Guarantees in a nutshell. The center x_0 of ball $\mathcal{B}_0 = B(x_0, \delta_0)$, with the initial perturbation δ_0, and samples x drawn uniformly from $\mathcal{B}_0$'s surface, are numerically integrated in time to $\chi(t_j, x_0)$ and $\chi(t_j, x)$, respectively. The Lipschitz 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 $\mathcal{B}_j$ depending on the chosen tightness factor μ. The ratio between the caps' surfaces and $\mathcal{B}_0$'s surface are correlated to the desired confidence $1 - \gamma$.

tuple (δ_0, φ), with angles $\varphi = (\varphi_1, \ldots, \varphi_{n-1})$, radius δ_0 and center x_0, having a conversion function $x((\delta_0, \varphi), x_0)$ from polar to Cartesian coordinates. Whenever the center x_0 and the radius δ_0 of the initial ball $\mathcal{B}_0$ are known from the context, we will use the following notation: $x(\varphi)$ for the conversion from polar to Cartesian coordinates, and just x if we do not want to mention the polar coordinates explicitly.

5.2.1 Forward-Mode Use of Adjoint-Sensitivity Method.

For both algorithms, we need the deformation gradient F_x for several sample points. The integral of Eq. (4.4) has the same form as the ODE used for reverse-mode automatic differentiation of Neural ODEs, when optimized by the adjoint sensitivity method [CRBD18] with one exception: our F_x defines the differential of the solution by the initial value, and their equivalent function a defines the differential of the loss function by the initial value. Their approach computes gradients by solving a second, augmented ODE backwards in time. In our case, solving the variational Eq. (4.4) until target time t_j, already gives us the required gradient F_x, but requires knowledge of $\chi(t, x)$ for all $t \in [t_0, t_j]$. This is why in our forward-mode adjoint sensitivity method, we propagate for all samples x using Eq. (3.2) and $F_x(t)$ using Eq. (4.4) forwards in time together until t_j, starting from its augmented (combined) initial state (x, I) and using its augmented dynamical system $(f(x), (\partial_x f)(\chi(t, x))F(t))$.

Algorithm 2 Stochastic Lagrangian Reachability (SLR)

Require: initial ball $\mathcal{B}_0 = B(x_0, \delta_0)$, time horizon T, sequence of timesteps t_j $(t_0 \leq t_1 \leq \cdots \leq t_k = T)$, tolerance $\mu > 1$, confidence level $\gamma \in (0,1)$, distance function d_j, gradient of loss $\nabla_\varphi L$

for $(j = 1; j \leq k; j = j+1)$ **do**
  $\mathcal{V}, \mathcal{U} \leftarrow \{\}$ (list of visited and random points)
  $x_j \leftarrow solveIVP(f, x_{j-1}, [t_{j-1}, t_j])$
  $[\mathcal{F}] \leftarrow F_{[\mathcal{X}_0]}(t_j) = rungeKuttaVariational((\partial_x f)([\mathcal{X}]), [\mathcal{F}], [t_{j-1}, t_j]))$
  compute $\Lambda \geq \|[\mathcal{F}]\|$ (interval arithmetic Lipschitz constant)
  $\bar{p} \leftarrow 0$, $\mathcal{S} \leftarrow \{\}$
  while $\bar{p} < 1 - \gamma$ **do**
    sample $x \in \mathcal{B}_0$ and add sample to $\mathcal{V}$ and $\mathcal{U}$
    $\chi(t_j, x), F_x(t_j) \leftarrow forwardModeAdjointSensitivity(x, I, [t_0, t_j])$
    if $x \notin \mathcal{S}$ **then** $findLocalMinimum(x, \nabla_\varphi L, F_x(t_j))$ and add x to $\mathcal{V}$
    $\bar{m} \leftarrow \max_{x \in \mathcal{V}} d_j(x)$
    $r_x \leftarrow computeSafetyRegionRadius(d_j(x), \bar{m}, \Lambda) \quad \forall x \in \mathcal{V}$
    $\mathcal{S} \leftarrow \bigcup_{x \in \mathcal{V}} B(x, r_x)$
    $\bar{p} \leftarrow \Pr(\mu \cdot \bar{m} \leq m^\star)$
  end while
  $\mathcal{B}_j \leftarrow B(x_j, \mu \cdot \bar{m})$
end for
return $(\mathcal{B}_1, \ldots, \mathcal{B}_k)$

5.2.2 Theoretical Statistical Verification Framework

In SLR, we solve each optimization problem globally, via uniform sampling, and locally, through gradient descent, whereas gradient descent is avoided in spherical-caps around the start/end states of previous searches. The cap radius is derived from its local Lipschitz constant, computed via interval arithmetic.

5.2.3 Gradient Computation

The SLR algorithm uses gradient descent locally, when solving the global optimization problem of Eq. (5.3). In [GHL+21] the *loss function* $L(\varphi) = -d_j \circ x(\varphi)$ is introduced in polar coordinates at time t_j to be able to do gradient descent on the surface, in order to find the optimum. L also depends on the initial radius δ_0 and initial center x_0; as these are fixed inputs, we do not consider them in the notation. Gradient descent is started from uniformly sampled points not contained in already constructed safety regions.

In [GHL+21], we introduced a new framework to compute the loss's gradient which is needed to find the local minimum in a unified fashion, and improved the optimization runtime by 50%, compared to the optimization scheme used in [CRBD18]: we save half of the time because we do not have to go backwards to compute the loss.

5.2.4 Safety-Region Computation

In contrast to the perspective in [GHL+21], we will discuss the problem of finding a global maximum of $d_j(x)$ for points $x \in \mathcal{B}_0$ instead of the equivalent problem of finding a global minimum of $L(\varphi)$ for points $\varphi \in \mathbb{R}^{n-1}$. With our global search strategy, we are covering the feasible region $\mathcal{B}_0^S$ with already visited points $\mathcal{V}$. Consequently, we have access to the current maximum in $\mathcal{V}$:

$$\bar{m}_{j,\mathcal{V}} = \max_{x \in \mathcal{V}} d_j(x) \tag{5.4}$$

with $\bar{m} \leq m^\star$, where $m^\star$ is the global maximum of Eq. (5.3). We will now identify safety regions for a continuous-depth model flow and describe how the use of these regions is incorporated in the SLR algorithm.

Definition 6 (Safety Region) *Let $x_i \in \mathcal{V} \subseteq \mathcal{B}_0$ be an already-visited point. A safety-radius $r_{x_i} = r(x_i)$ defines a* safe spherical-cap *$B(x_i, r_{x_i})^S = B(x_i, r_{x_i}) \cap \mathcal{B}_0^S$, if it holds that $d_j(y) \leq \mu \cdot \bar{m}$ for all $y \in B(x_i, r_x)^S$ and $\bar{m} \leq m^\star$.*

In Thm. 3 below, we use Thm. 1 to bound the local Lipschitz constant (Def. 4) and to define the radius r_x of the safety region $B(x, r_x)^S$ around an already-visited point $x \in \mathcal{V}$.

Theorem 3 (Radius of Safety Region ([GHL+21], Thm. 1)) *At target time t_j, let $\bar{m}$ be the current global maximum, as in Eq. (5.4). Let $x \in \mathcal{V}$ be an already-visited point with value $d_j(x)$, and let r_x and $B(x, r_x)^S$ be defined as follows:*

$$r_x = \lambda_{\Sigma_x}^{-1}(\mu \cdot \bar{m} - d_j(x)) \tag{5.5}$$

with $\mu > 1$, $\lambda_{\Sigma_x} = \max_{y \in \Sigma_x} \|F_y(t_j)\|_{M_{0,j}}$ and $\Sigma_x \supseteq B(x, r_x)^S$, then it holds that:

$$d_j(y) \leq \mu \cdot \bar{m} \quad \forall y \in B(x, r_x)^S \tag{5.6}$$

We can now use these safety regions around the samples to compute the probability needed in Line 14 of Algorithm 2:

$$\Pr(\mu \cdot \bar{m} \geq m^\star) \geq \Pr(\exists x \in \mathcal{U} : S_x \ni x^\star) = 1 - \prod_{x \in \mathcal{U}} (1 - \Pr(S_x)), \tag{5.7}$$

with $S_x = B(x, r_x)^S$ being the safety region around x. In [GHL+21], we provide a convergence guarantee as well as a convergence rate for the probability $\Pr(S_x) = \text{Area}(S_x)/\text{Area}(\mathcal{B}_0)$. Thm. 2 of [GHL+21] shows that in the limit of the number of samples, the constructed reachset converges with probability 1 to the smallest ellipsoid that encloses the true reachable set using tightness bound μ.

5.2.5 Theoretical Contributions

With SLR, we presented a theoretical framework for the statistical verification of Neural ODEs and provided an answer to Research-Question 3 in Chapter 2. We summarize our key research contributions as follows:

- We introduced a theoretical framework for the verification of Neural ODEs by restating the reachability problem as a set of global-optimization problems.
- We solved each optimization problem globally, via uniform sampling, and locally, through gradient descent (GD), avoiding costly Hessian computations.
- GD is avoided in spherical-caps around the start/end states of previous searches. The cap radius is computed from its local Lipschitz constant via interval arithmetic.
- We designed a forward-mode GD algorithm based on the so-called adjoint sensitivity method for (Neural) ODEs.
- We proved convergence properties of SLR, the safety guarantees it ensures, and analysed its time and space complexity.

5.3 GoTube: Scalable Statistical Verification of Continuous-Depth Models

As we implemented the SLR algorithm, we observed that even after resolving its first-occurring inefficient sampling and its vanishing gradient problems, SLR still blew up in time, even for low-dimensional benchmarks such as the Dubins Car. Our GoTube algorithm and its associated theory solve fundamental scalability problems of SLR (see Table 3.1), by replacing the interval arithmetic used to compute deterministic caps, with statistical Lipschitz caps. This enables us to verify continuous-depth models up to an arbitrary time-horizon, a capability beyond what was achievable before.

To be able to do that, we formulated theorems on: 1) How to choose the radius of a Lipschitz cap, using statistical bounds of local Lipschitz constants of the samples, together with the expected difference quotients of their Lipschitz constants. 2) How to provide convergence guarantees using these new statistical caps, as they are used by GoTube to compute the probability of $\mu \cdot \delta_j$ being an upper bound of the biggest perturbation. In addition, as in machine learning, we encapsulated a large number of samples within a tensor. This allowed us to dramatically increase the computation speed by employing the latest advances in machine learning technology.

We start by describing the novelties in GoTube compared to SLR. This facilitates the comprehension of the different computation and theory steps. Although the input and output is similar to Algorithm 2, we had to significantly change the algorithm by creating new theorems, such that GoTube is scalable and also works on continuous-depth models.

GoTube starts by sampling a batch (tensor) $x^B \in \mathcal{B}_0^S$ and if needed, it adds new samples to that tensor in Line 14 and computes every step in a tensorized manner, for all samples

Algorithm 3 GoTube

Require: initial ball $\mathcal{B}_0 = B(x_0, \delta_0)$, time horizon T, sequence of timesteps t_j ($t_0 < \cdots < t_k = T$), error 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**
    $\mathcal{V} \leftarrow \mathcal{V} \cup \{x^B\}$
    $x_j \leftarrow \mathit{solveIVP}(f, x_{j-1}, [t_{j-1}, t_j])$
    $\bar{m}_{j,\mathcal{V}} \leftarrow \max_{x \in \mathcal{V}} d(t_j, x)$
    $x, F_x(t_j) \leftarrow \mathit{forwardModeAdjointSensitivity}(x, I, [t_0, t_j]) \quad \forall x \in \mathcal{V}$
    compute local Lipschitz constants $\lambda_x = \|F_x\|$ for $x \in \mathcal{V}$
    compute statistical quantile $\Delta\lambda_{\mathcal{V}}$
    compute cap radii $r_x(\lambda_x, \Delta\lambda_{\mathcal{V}})$ (Lipschitz Cap) for $x \in \mathcal{V}$
    $\bar{p} \leftarrow \mathit{computeProb}(\gamma, \{r_x : x \in \mathcal{V}\}, n, \delta_0)$
    sample batch $x^B \in \mathcal{B}_0$
  end while
  $\mathcal{B}_j \leftarrow B(x_j, \mu \cdot \bar{m}_{j,\mathcal{V}})$
end for
return $(\mathcal{B}_1, \ldots, \mathcal{B}_k)$

at the same time. 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, and their local Lipschitz constant according to the variational Eq. (4.4) using the forward-mode adjoint sensitivity method. Unlike SLR, F_x is not used within gradient descent to find local optima, but to compute local Lipschitz constants for all samples.

Based on this information, GoTube then computes a statistical upper bound for Lipschitz constants and the cap radii accordingly. The total surface of the caps is then employed to compute and update the achieved confidence. Once the desired confidence is achieved, GoTube exits the inner loop, and computes the bounding ball in terms of its center and radius, which is given by tightness factor μ times the maximum distance $\bar{m}_{j,\mathcal{V}}$. After exiting the outer loop, GoTube returns the reachtube.

Definition 7 (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, are a statistical version of the safety regions around samples, commonly used to cover a state space. Intuitively, the points within a cap do not have to be explored. The difference with Lipschitz caps is that we statistically bound the values inside that space, and develop a theory enabling us to calculate a probability of having found an upper bound of the true maximum $m_j^\star = d_j(x_j^\star) = \max_{\{x_1,\dots,x_m\}\subset\mathcal{B}_0} d_j(x)$ for any m-dimensional set of the optimization problem in Eq. (5.3).

Our objective is to avoid the usage of interval arithmetic, for computing the Lipschitz constant - as it is done in SLR - because this impedes scaling up to continuous depth models. Instead, we define statistical bounds on the Lipschitz constant, to set the radius r_x of the Lipschitz caps, such that $\mu \cdot \bar{m}_{j,\mathcal{V}}$ is a γ-statistical upper bound for all distances $d_j(y)$ at time t_j, from values inside the ball $B(x, r_x)^S$.

Theorem 4 (Radius of Statistical Lipschitz Caps, [GLH+21], Thm. 1) *Given a continuous-depth model f from Eq. (3.2), $\gamma \in (0,1)$, $\mu > 1$, 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 $x \in \mathcal{V}$. Let $\nu_x = |\lambda_x - \lambda_X|/\|x - X\|$, for $x \in \mathcal{V}$, 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, and $\Delta\lambda_\mathcal{V}$ be the upper bound of the confidence interval of the mean $\mathbb{E}\nu_x$ defined as follows:*

$$\Delta\lambda_\mathcal{V}(\gamma) = \overline{\nu_x} + t^*_{\gamma/2}(N-2)\frac{s(\nu_x)}{\sqrt{N-1}}, \tag{5.8}$$

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:

$$r_x = \frac{\left(-\lambda_x + \sqrt{\lambda_x^2 + 4 \cdot \Delta\lambda_\mathcal{V} \cdot (\mu \cdot \bar{m}_{j,\mathcal{V}} - d_j(x))}\right)}{2 \cdot \Delta\lambda_\mathcal{V}}, \tag{5.9}$$

then it holds that:

$$\Pr\left(d_j(y) \le \mu \cdot \bar{m}_{j,\mathcal{V}}\right) \ge 1 - \gamma \quad \forall y \in B(x, r_x)^S, \tag{5.10}$$

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

Using conditional probabilities, we were able to state that the convergence guarantee also holds for the GoTube algorithm, thus ensuring that the algorithm terminates in finite time, even when using statistical Lipschitz caps around the samples, instead of the deterministic local balls in [GLH+21][Thm. 2].

5.3.1 Theoretical Contributions

With GoTube, we provide the necessary theory extension in order to make a scalable tool for statistical robustness analysis out of the theoretical framework SLR. As GoTube

is stable and sets the state-of-the-art in terms of its ability to scale to time horizons well beyond what has been previously possible, we anwer the last research question of Chapter 2. In summary, the theoretical contributions of GoTube are the following:

- We introduced a novel and efficient theory for computing statistical bounds for the Lipschitz constant of systems of nonlinear ODEs, which helps us achieve tight reachtubes for continuous-depth neural network models.
- We proved convergence guarantees for the GoTube algorithm, thus ensuring that the algorithm always terminates in finite time even when using statistical Lipschitz caps around the samples instead of deterministic local balls.

5.3.2 Experimental Evaluation

With a variety of experiments, we evaluated GoTube's performance and compared it with other tools using the volume of the reachtubes or the time horizon as a metric. We showed experimentally, that GoTube can trade runtime for reachtube tightness.

Our first experimental evaluation was concerned with the over-approximation errors of the constructed bounding tubes. The results are shown in Table 5.3. For the first five benchmarks, which are classical dynamical systems, we use the small time horizons T and small initial radii δ_0, which the other tools could also handle. GoTube, with 99% confidence, achieves a competitive performance to the other tools, coming out on top in 3 out of 5 benchmarks, by using $\mu = 1.1$ as the tightness bound. Intuitively this means, we are confident that the over-approximation includes all executions with a confidence level $1 - \lambda$, but this over-approximation might not be as tight as desired. GoTube is able to achieve any desired tightness by reducing μ and increasing the runtime.

The second experimental evaluation was concerned with the time interval for which GoTube and existing methods can construct a reachtube before exploding due to over-approximation errors. To this end, we extended the benchmarks by increasing the time horizon for which the reachtube should be constructed, used tightness bound $\mu = 1.1$ and

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**

Table 5.2: Results of the extended benchmark by longer time horizons. The numbers show the volume of the constructed tube, "Blowup" indicates that the method produced `Inf` or `NaN` values due to a blowup. Lower is better; the best method is shown in bold.

set a 95% confidence level, that is, probability of being conservative. Results in Table 5.2 demonstrate that GoTube produces significantly longer reachtubes than all considered state-of-the-art approaches, without suffering from severe over-approximation errors.

Benchmark	LRT-NG	Flow*	CAPD	LRT	GoTube (90%)	GoTube (99%)
Brusselator	1.5e-4	9.8e-5	3.6e-4	6.1e-4	8.6e-5	**8.6e-5**
Van Der Pol	4.2e-4	**3.5e-4**	1.5e-3	3.5e-4	3.5e-4	**3.5e-4**
Robotarm	**7.9e-11**	8.7e-10	1.1e-9	Fail	2.5e-10	2.5e-10
Dubins Car	0.131	4.5e-2	0.1181	385	2.5e-2	**2.6e-2**
Cardiac Cell	**3.7e-9**	1.5e-8	4.4e-8	3.2e-8	4.2e-8	4.3e-8
CartPole-v1+LTC	4.49e-33	Fail	Fail	Fail	2.6e-37	**4.9e-37**
CartPole-v1+CTRNN	3.9e-27	Fail	Fail	Fail	9.9e-34	**1.2e-33**

Table 5.3: Comparison of GoTube (using tightness bound $\mu = 1.1$) to existing reachability methods. The first five 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 [HLA+21, VHA+21]. The numbers show the volume of the constructed tube. Lower is better; best number in bold.

CHAPTER 6

Discussions, Scope and Conclusions

6.1 Comparison of our Algorithms

A common aspect of LRT, LRT-NG, SLR, and GoTube, is that they all make use of the variational equations of Eq. (4.4), together with the mean value theorem (Thm. 1). They allow our algorithms to have tighter bounds, less wrapping effect, and to be more efficient than other tools, as shown in the experimental evaluation. It is nevertheless important to know that for each algorithm, we had to develop new tools and theoretical techniques, allowing to avoid the blow up in space as well as in time.

When computing conservative guarantees, as it was done in LRT-NG [GCL+20], we employed the propagated interval deformation gradient, by using the interval version of the variational equations of Eq. (4.4), and by multiplying the starting radius δ_0 with the resulting stretching factor $\|\mathcal{F}_j\|_{M_j}$ to over-approximate the set of reachable states at time t_j. As we needed to use $[\mathcal{X}_{j-1}]$ to compute $\|\mathcal{F}_j\|_{M_j}$, the theoretical contributions of optimal metric computation and balls intersection with ellipsoids, are responsible for being the only conservative tool that could also verify continuous-depth models, by avoiding the accumulation of small errors (the infamous wrapping effect).

For the theoretical stochastic version of Lagrangian Reachability (the SLR algorithm in [GHL+21]), we used the variational equations even in two different ways: 1) To propagate the deformation gradient for several samples, by using the forward-mode adjoint sensitivity method, when calculating the gradient of loss needed to find local minima. 2) To propagate the interval variational equations and use the results of Thm. 1, to compute an upper bound of the Lipschitz constants for the distance function $d_j(x)$. This upper bound was then used to compute the safety regions radii.

In our scalable statistical robustness analysis algorithm GoTube, we completely avoided the use of interval arithmetic, as the interval Lipschitz constant in SLR lead to a blow-up in time. In Algorithm 3, the variational equation is used to compute $F_x(t)$ via the forward mode adjoint sensitivity method for a tensorized batch of samples. We presented a new theory on how to compute statistical upper bounds of the local Lipschitz constants λ_x for the samples, which we used to compute the cap radiuses and thus the probability.

Instead of using Lipschitz constants as a bloating factor for the ball's radius as in LRT-NG, we used it in SLR and GoTube to define regions (caps) around already visited points on the surface, and to compute an upper bound for the values inside that caps: either deterministic safety regions (SLR), or statistical Lipschitz caps (GoTube). This knowledge allowed us to compute the probability of having an upper bound for the global maximum of Eq. (5.3). The bigger the Lipschitz constant, the smaller the safety-region radius and thus the confidence of the reachtube. So a huge difference between the conservative and the statistical method is that a too large upper bound of the Lipschitz constant results in a state explosion for LRT-NG but in a time explosion for SLR and GoTube.

Another difference is that LRT-NG always computes as-tight-as-possible reachtubes, given the dynamical system. In contrast, SLR and GoTube allow to trade between time and accuracy, by using the tightness bound parameter μ. Thus, after finishing our global search strategy for timestep t_j, we have the statistical guarantee that the functional value of every $x \in \mathcal{B}_0$ is less or equal to $\mu \cdot \bar{m}$. This implies that we should initiate the search with a relatively large $\mu = \mu_1$, obtaining for every x a relatively large value of r_{x,μ_1} and therefore obtain a faster coverage of the search space. Subsequently, we can investigate whether the reachset $\mathcal{B}_j$ with radius $\delta_j = \mu_1 \cdot \bar{m}$ intersects with a region of bad (unsafe) states. If this is not the case, we can proceed to the next timestep t_{j+1}. Otherwise, we reduce μ to $\mu_2 < \mu_1$. Accordingly, we can find a first radius for $\mathcal{B}_j$ faster and refine it as long as $\mathcal{B}_j$ intersects with the region of bad states.

6.2 Scientific Contributions

In this thesis, we presented technical solutions and new theorems, and introduced practical verification algorithms and tools, for the reachability analysis of CPS with CDNN controllers, not only in a conservative but also in a statistical way.

We significantly improved the state-of-the-art in conservative reachability analysis of nonlinear ODEs with our LRT-NG theory and tool, which demonstrated a superior performance compared to LRT, CAPD and Flow*. From a theoretical point of view, we introduced a novel theorem for the analytical metric computation in an optimal way, such that it minimizes the volume of the reachtube. In addition, we introduced an intersection of ellipsoids and balls in the reachset computation, which was seen as two opposite approaches in previous work. Together with the improved center propagation, we minimized the over-approximation in every timestep, and thus we were able to run the reachability analysis for a longer time horizon than previously possible. Especially for Neural ODEs the difference was the biggest: we were able to run the Cartpole

model with a Neural ODEs controller until 1 second whereas CAPD and Flow* were not able to construct a reachtube for a longer time horizon than 0.135 second. From a tool perspective view, we improved upon LRT by providing a standalone tool with an improved interface and a scalable Runge-Kutta time-integrator.

We widened the limited possibilities of conservative reachability analysis, where an over-approximation of the reachset is provided without any uncertainties, by introducing a statistical confidence interval for the reachset computation. We introduced the theoretical framework SLR for statistical verification of Neural ODEs, by restating the reachability problem as a set of global-optimization problems. To locally solve the optimization problem in an efficient way, we presented a novel forward-mode gradient-descent algorithm based on the adjoint method for Neural ODEs [CRBD18]. Finally, we proved the convergence properties of the theoretical framework SLR.

After designing a theoretical framework for statistical reachability analysis, we developed the statistical robustness analysis algorithm GoTube. To put the theoretical results into practice, there were several adaptations and also new theories needed. We presented a novel and efficient theory for computing statistical bounds for the Lipschitz constant and proved convergence properties of GoTube with stochastic Lipschitz Caps. Finally, we demonstrated that GoTube considerably outperforms state-of-the-art verification tools on the highly complex task of robustness analysis of CPS with CDNN controllers.

6.3 Future Work

This thesis presents several intriguing avenues for future research. One promising direction is to extend robustness analysis to black-box systems, such as those encountered in real-world applications like autonomous driving, where the underlying differential equations of the environment are unknown. Investigating how to compute statistical upper bounds of Lipschitz constants using only simulated outputs of the function and not the function's derivative would be a significant research question in this context.

Another interesting area of exploration is the combination of conservative (LRT-NG) and statistical (GoTube) robustness analysis. Utilizing a statistical tool to achieve a probabilistic guarantee (e.g., 99%), and then using a conservative version of LRT-NG during runtime to monitor for unsafe states and trigger alerts or switch to a safe fallback controller, could be a valuable approach. As it would be crucial to synchronously monitor for critical events, the open question here is how to speed up LRT-NG such that it is able to run in parallel with the controller. A possible research direction would be to precompute set-based reachtubes and just compute deviations from the closest trajectory or reachtube bound in runtime. Another approach would be to compute a reachtube backwards in time starting from unsafe states to know in which region it is necessary to switch to a safe fallback controller. In any case, the execution time of the LRT-NG algorithm would introduce a new safety hazard to the system.

Furthermore, exploring the incorporation of GoTube in the training cycle of a neural

network controller to proactively train a safe controller is another interesting research direction. Investigating the feasibility of incorporating reachtubes in computing the loss function or leveraging insights from reachtube robustness analysis to retrain the neural network controller to satisfy safety criteria could yield valuable insights.

In summary, there are several compelling avenues for future research, including extending robustness analysis to black-box systems, combining conservative and statistical robustness analysis, addressing the challenge of synchronous monitoring, and incorporating reachtubes in the training cycle of neural network controllers.

Acronyms

CDNN continuous-depth neural networks. 3–7, 9, 12, 13, 23, 25, 36, 37

CPS cyber-physical systems. 3, 7, 9, 11, 12, 15, 16, 36, 37

IVP initial value problem. 11, 18, 19

LRT Lagrangian Reachability. 5, 6, 9, 10, 14, 17, 19, 20, 23, 25, 35–37

Neural ODEs neural ordinary differential equations. 3, 9, 10, 23–26, 29, 36, 37

NNC neural network controller. 11–13, 15, 16

ODE ordinary differential equations. 3–5, 9–15, 25, 36

SLR Stochastic Lagrangian Reachability. 25, 27–31, 35–37

Bibliography

[Abe98] R. Abeyaratne. *Continuum Mechanics*. Lecture Notes on The Mechanics of Elastic Solids, 1998.

[AGK18] M. Althoff, D. Grebenyuk, and N. Kochdumper. Implementation of Taylor models in CORA 2018. In *Proc. of the 5th International Workshop on Applied Verification for Continuous and Hybrid Systems*, pages 145–173, 2018.

[Alt13] M. Althoff. Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets. In *HSCC*, pages 173–182, 2013.

[BCP+19] Luca Bortolussi, Francesca Cairoli, Nicola Paoletti, Scott A. Smolka, and Scott D. Stoller. Neural predictive monitoring. In Bernd Finkbeiner and Leonardo Mariani, editors, *Runtime Verification*, pages 129–147, Cham, 2019. Springer International Publishing.

[BDPD+20] Rudy Bunel, Alessandro De Palma, Alban Desmaison, Krishnamurthy Dvijotham, Pushmeet Kohli, Philip Torr, and M Pawan Kumar. Lagrangian decomposition for neural network verification. In *UAI*, pages 370–379. PMLR, 2020.

[BS14] 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.

[CÁS13] Xin Chen, Erika Ábrahám, and Sriram Sankaranarayanan. Flow*: an analyzer for non-linear hybrid systems. In *CAV*, pages 258–263, 2013.

[CIB+17] J. Cyranka, M. A. Islam, G. Byrne, P. Jones, S. A. Smolka, and R. Grosu. Lagrangian reachability. In Rupak Majumdar and Viktor Kunčak, editors, *CAV*, pages 379–400, Heidelberg, Germany, July 2017. Springer.

[CIS+18] J. Cyranka, M. A. Islam, S. A. Smolka, S. Gao, and R. Grosu. Tight Continuous-Time Reachtubes for Lagrangian Reachability. In *CDC*, pages 6854–6861. IEEE, 2018.

[CRBD18] 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, *NeurIPS 31*, pages 6571–6583. Curran Associates, Inc., 2018.

[DBMP19] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. In *NeurIPS*, pages 7511–7522, 2019.

[DCS19] Souradeep Dutta, Xin Chen, and Sriram Sankaranarayanan. Reachability analysis for neural feedback systems using regressive polynomial rule inference. In *Proceedings of the 22Nd ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '19, pages 157–168, New York, NY, USA, 2019. ACM.

[DDT19] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural odes. In *NeurIPS*, pages 3140–3150, 2019.

[DKAZ20] 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.

[DM07] Alexandre Donzé and Oded Maler. Systematic simulation using sensitivity analysis. In *HSCC*, pages 174–189, 2007.

[DMVP15] 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.

[Don10] Alexandre Donzé. Breach, a toolbox for verification and parameter synthesis of hybrid systems. In *CAV*, pages 167–170, Edinburgh, UK, July 2010. Springer.

[EAQM20] N Benjamin Erichson, Omri Azencot, Alejandro Queiruga, and Michael W Mahoney. Lipschitz recurrent neural networks. *arXiv preprint arXiv:2006.12070*, 2020.

[Ehl17] 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.

[ES11] Joshua A. Enszer and Mark A. Stadtherr. Verified solution and propagation of uncertainty in physiological models. *Reliab. Comput.*, 15(3):168–178, 2011.

[FHC+20] Jiameng Fan, Chao Huang, Xin Chen, Wenchao Li, and Qi Zhu. Reachnn*: A tool for reachability analysis of neural-network controlled systems. In *Automated Technology for Verification and Analysis: 18th International Symposium, ATVA 2020, Hanoi, Vietnam, October 19–23, 2020, Proceedings*, page 537–542, Berlin, Heidelberg, 2020. Springer-Verlag.

[FJNO20] Chris Finlay, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam Oberman. How to train your neural ode: the world of jacobian and kinetic regularization. In *ICML*, pages 3154–3164. PMLR, 2020.

[FKJM16] Chuchu Fan, James Kapinski, Xiaoqing Jin, and Sayan Mitra. Locally optimal reach set over-approximation for nonlinear systems. In *ICES*, EMSOFT '16, pages 6:1–6:10, New York, NY, USA, 2016. ACM.

[FKJM17] Chuchu Fan, James Kapinski, Xiaoqing Jin, and Sayan Mitra. Simulation-driven reachability using matrix measures. *ACM Trans. Embed. Comput. Syst.*, 17(1), December 2017.

[FM15] Chuchu Fan and Sayan Mitra. Bounded verification with on-the-fly discrepancy computation. In Bernd Finkbeiner, Geguang Pu, and Lijun Zhang, editors, *Automated Technology for Verification and Analysis*, pages 446–463, Cham, 2015. Springer International Publishing.

[GCI+19] S. Gruenbacher, J. Cyranka, M. A. Islam, M. Tschaikowski, S.A. Smolka, and R. Grosu. Under the Hood of a Stand-Alone Lagrangian Reachability Tool. *EPiC Series in Computing*, 61, 2019.

[GCL+20] Sophie Gruenbacher, Jacek Cyranka, Mathias Lechner, Md Ariful Islam, Scott A. Smolka, and Radu Grosu. Lagrangian reachtubes: The next generation. In *CDC*, pages 1556–1563, 2020.

[GDS+18] 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.

[GHL+21] Sophie Gruenbacher, Ramin Hasani, Mathias Lechner, Jacek Cyranka, Scott A. Smolka, and Radu Grosu. On the verification of neural odes with stochastic guarantees. *AAAI*, 35(13):11525–11535, May 2021.

[GKC13] S. Gao, S. Kong, and E. M. Clarke. Satisfiability modulo odes. In *2013 Formal Methods in Computer-Aided Design*, pages 105–112, 2013.

[GLH+21] Sophie Gruenbacher, Mathias Lechner, Ramin M. Hasani, Daniela Rus, Thomas A. Henzinger, Scott A. Smolka, and Radu Grosu. Gotube: Scalable stochastic verification of continuous-depth models. *CoRR*, abs/2107.08467, 2021.

[HDT10] Milan Hladik, David Daney, and Elias Tsigaridas. Bounds on real eigenvalues and singular values of interval matrices. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2116–2129, 2010.

[HFL+19] Chao Huang, Jiameng Fan, Wenchao Li, Xin Chen, and Qi Zhu. Reachnn: Reachability analysis of neural-network controlled systems, 2019.

[HKT20] Philipp Holl, Vladlen Koltun, and Nils Thuerey. Learning to control pdes with differentiable physics. *arXiv preprint arXiv:2001.07457*, 2020.

[HLA+20] Ramin Hasani, Mathias Lechner, Alexander Amini, Daniela Rus, and Radu Grosu. The natural lottery ticket winner: Reinforcement learning with ordinary neural circuits. In *ICML*. JMLR. org, 2020.

[HLA+21] Ramin Hasani, Mathias Lechner, Alexander Amini, Daniela Rus, and Radu Grosu. Liquid time-constant networks. *AAAI*, 35(9), 2021.

[ICW+20] Radoslav Ivanov, Taylor J. Carpenter, James Weimer, Rajeev Alur, George J. Pappas, and Insup Lee. Verifying the safety of autonomous systems with neural network controllers. *ACM Trans. Embed. Comput. Syst.*, 20(1), dec 2020.

[ICW+21] Radoslav Ivanov, Taylor Carpenter, James Weimer, Rajeev Alur, George Pappas, and Insup Lee. Verisig 2.0: Verification of neural network controllers using taylor model preconditioning. In Alexandra Silva and K. Rustan M. Leino, editors, *Computer Aided Verification*, pages 249–262, Cham, 2021. Springer International Publishing.

[Imm15] 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.

[IWA+18] Radoslav Ivanov, James Weimer, Rajeev Alur, George J. Pappas, and Insup Lee. Verisig: verifying safety properties of hybrid systems with neural network controllers. *CoRR*, abs/1811.01828, 2018.

[JB19] Junteng Jia and Austin R Benson. Neural jump stochastic differential equations. In *NeurIPS*, pages 9847–9858, 2019.

[KMFL20] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *arXiv preprint arXiv:2005.08926*, 2020.

[KMWZ20] 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.

[LBB20] 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, 2020. Springer International Publishing.

[LH20] Mathias Lechner and Ramin Hasani. Learning long-term dependencies in irregularly-sampled time series. *arXiv preprint arXiv:2006.04418*, 2020.

[LHA+20] Mathias Lechner, Ramin Hasani, Alexander Amini, Thomas A Henzinger, Daniela Rus, and Radu Grosu. Neural circuit policies enabling auditable autonomy. *Nature MI*, 2(10):642–652, 2020.

[LHZ+19] Mathias Lechner, Ramin Hasani, Manuel Zimmer, Thomas Henzinger, and Radu Grosu. Designing worm-inspired neural networks for interpretable robotic control. In *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)*, Montreal, Canada, May 2019.

[Loh92] R.J. Lohner. *Computation of guaranteed enclosures for the solutions of ordinary initial and boundary value problems*, chapter Computational Ordinary Differential Equations. Clarendon Press, Oxford, 1992.

[LZLD18] Yiping Lu, Aoxiao Zhong, Quanzheng Li, and Bin Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In Jennifer Dy and Andreas Krause, editors, *35th ICML, ICML 2018*, volume 7, pages 5181–5190. International Machine Learning Society (IMLS), 1 2018.

[MA15] J. Maidens and M. Arcak. Reachability analysis of nonlinear systems using matrix measures. *IEEE Transactions on Automatic Control*, 60(1):265–270, 2015.

[MB03] K. Makino and M. Berz. Taylor models and other validated functional inclusion methods. *Int. J. Pure Appl. Math*, 4(4):379–456, 2003.

[MB06] M. Martel and O. Bouissou. GRKLib: a Guaranteed Runge Kutta Library. In *2006 12th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics (SCAN)*, volume 00, page 8, 2006.

[MDA19] 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.

[MGV18] Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In *ICML*, pages 3578–3586. PMLR, 2018.

[MPP+20] Stefano Massaroli, Michael Poli, Jinkyoo Park, Atsushi Yamashita, and Hajime Asama. Dissecting neural odes. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3952–3963. Curran Associates, Inc., 2020.

[NJC99] N.S. Nedialkov, K.R. Jackson, and G.F. Corliss. Validated solutions of initial value problems for ordinary differential equations. *Applied Mathematics and Computation*, 105(1):21 – 68, 1999.

[NJN07] M. Neher, K.R. Jackson, and N.S. Nedialkov. On taylor model based integration of odes. *SIAM J. Numer. Anal*, 45:236–262, 2007.

[OO04] Jeremy E. Oakley and Anthony O'Hagan. Probabilistic sensitivity analysis of complex models: A bayesian approach. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 66(3):751–769, 2004.

[QGMK19] Alessio Quaglino, Marco Gallieri, Jonathan Masci, and Jan Koutník. Snode: Spectral discretization of neural odes for system identification. *arXiv preprint arXiv:1906.07038*, 2019.

[RCD19] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *NeurIPS*, pages 5320–5330, 2019.

[Roh98] J. Rohn. Bounds on eigenvalues of interval matrices. *ZAMMZ.Angew.Math.Mech.*, 78:1049–1050, 1998.

[Rum01] Siegfried M. Rump. Computational error bounds for multiple or nearly multiple eigenvalues. *Linear Algebra and its Applications*, 324(1):209 – 226, 2001.

[RWS+19] Wenjie Ruan, Min Wu, Youcheng Sun, Xiaowei Huang, Daniel Kroening, and Marta Kwiatkowska. Global robustness evaluation of deep neural networks with provable guarantees for the hamming distance. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5944–5952. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[SB13] Joseph K. Scott and Paul I. Barton. Bounds on the reachable sets of nonlinear control systems. *Automatica*, 49(1):93 – 100, 2013.

[Sla02] W.S. Slaughter. *The Linearized Theory of Elasticity*. Springer Science and Business Media, LLC, 2002.

[SZ15a] Fedor Shmarov and Paolo Zuliani. Probreach: A tool for guaranteed reachability analysis of stochastic hybrid systems. In Sergiy Bogomolov and Ashish Tiwari, editors, *SNR-CAV*, volume 37, pages 40–48, 2015.

[SZ15b] Fedor Shmarov and Paolo Zuliani. Probreach: verified probabilistic delta-reachability for stochastic hybrid systems. In *HSCC*, pages 134–139. ACM, 2015.

[TBXJ20] Hoang-Dung Tran, Stanley Bak, Weiming Xiang, and Taylor T. Johnson. Verification of deep convolutional neural networks using imagestars. In Shuvendu K. Lahiri and Chao Wang, editors, *Computer Aided Verification*, pages 18–42, Cham, 2020. Springer International Publishing.

[VHA+21] Charles Vorbach, Ramin Hasani, Alexander Amini, Mathias Lechner, and Daniela Rus. Causal navigation by continuous-time neural networks. *arXiv preprint arXiv:2106.08314*, 2021.

[WCN+18] Tsui-Wei Weng, Pin-Yu Chen, Lam M. Nguyen, Mark S. Squillante, Ivan V. Oseledets, and Luca Daniel. PROVEN: certifying robustness of neural networks with a probabilistic approach. *CoRR*, abs/1812.08329, 2018.

[WZ12] Daniel Wilczak and Piotr Zgliczyński. Cr-Lohner algorithm. *Schedae Informaticae*, 2011(Volume 20), 2012.

[YDTF20] Hanshu Yan, Jiawei Du, Vincent YF Tan, and Jiashi Feng. On robustness of neural ordinary differential equations. *ICLR*, 2020.

[YWL+20] Yibo Yang, Jianlong Wu, Hongyang Li, Xia Li, Tiancheng Shen, and Zhouchen Lin. Dynamical system inspired adaptive time stepping controller for residual network families. *AAAI*, 2020.

[Zgl02] Piotr Zgliczynski. C1 lohner algorithm. *Foundations of Computational Mathematics*, pages 429–465, 2002.

Part II

Publications

CHAPTER 7

Lagrangian Reachtubes: The Next Generation

Title	Lagrangian Reachtubes: The Next Generation
Authors	Gruenbacher, S., Cyranka, J., Lechner, M., Islam, M.A., Smolka, S.A., Grosu, R.
Published in	2020 59th IEEE Conference on Decision and Control (CDC)
Year	2020
Url	https://doi.org/10.1109/CDC42340.2020.9304042
Peer Reviewed	Yes
h-index	180
Status	Published
Award	IEEE TC Outstanding Student Paper Prize

CHAPTER 8

On the Verification of Neural ODEs with Stochastic Guarantees

Title	On the Verification of Neural ODEs with Stochastic Guarantees
Authors	Gruenbacher, S., Hasani, R., Lechner, M., Cyranka, J., Smolka, S.A., Grosu, R.
Published in	Proceedings of the AAAI Conference on Artificial Intelligence, 35(13), 11525-11535
Year	2021
Url	https://ojs.aaai.org/index.php/AAAI/article/view/17372
Peer Reviewed	Yes
h-index	180
Acceptance Rate	21%
Status	Published

CHAPTER 9

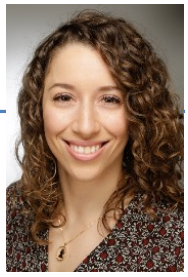
GoTube: Scalable Statistical Verification of Continuous-Depth Models

Title	GoTube: Scalable Statistical Verification of Continuous-Depth Models
Authors	Gruenbacher, S., Hasani, R., Lechner, M., Cyranka, J., Smolka, S.A., Grosu, R.
Published in	Proceedings of the AAAI Conference on Artificial Intelligence, 36(6), 6755-6764
Year	2022
Url	https://doi.org/10.1609/aaai.v36i6.20631
Peer Reviewed	Yes
h-index	180
Acceptance Rate	15%
Status	Published
Award	2021 Scientia Prize

Part III

Appendix

Sophie A. Neubauer



Personal Details

Name:	DI Sophie A. Neubauer (née Grünbacher)	Birthday:	[redacted]
Nationality:	Austria	Telephone:	[redacted]
E-mail:	sophie@datenvorsprung.at	Address:	[redacted]

Education

Since Nov. 2018 — **Research Assistant** at **TU Wien (Cyber-Physical Systems Group)**

- Admission to the doctoral program LogiCS (seeking exceptionally talented and motivated students)
- Research topic: Reachability Analysis of cyber-physical systems with neural network controllers
- [TC-DES and TC-HS Outstanding Student Paper Prize] September 2021: Received a prestigious IEEE award for the CDC-20 paper „Lagrangian Reachtubes: The Next Generation“.
- [Accepted papers at AAAI-21 and AAAI-22] The guide2research.com-ranking of computer science conferences puts AAAI at #5 out of over 1000 venues. The acceptance rate for the conference in 2021 was 21% and in 2022 only 15%.
- September 2019: Organized the IFIP WG 2.2 meeting between September 23-25 in Vienna (http://sites.google.com/view/ifip-wg22-vienna). The meeting had distinguished speakers from Australia, Austria, Belgium, Czech Republic, France, Germany, India, Italy, Norway, Poland, and the UK. There I also had the opportunity to give a talk on "Next-Generation Lagrangian Reachability".

2012-2014 — **Master Program in Statistics and Mathematics in Economics, TU Wien**

- Passed with honors (grade average of 1.16)
- Participated in the high potential program “TUtheTOP” of the Vienna University of Technology
- Master Thesis: “Optimal Bidding in the Sponsored Search”

2009-2012 — **Bachelor Program in Statistics and Mathematics in Economics, TU Wien**

- Passed with honors (grade average of 1.11)
- Bachelor Thesis: “Sustainability of fiscal policies”

2001-2009 — **Realgymnasium Kollegium Kalksburg, Vienna**

- Passed each grade and the Matura with honors
- High School Thesis: “Cryptography on chip cards”

Work Experience

Since 2020 — **Co-founder and CTO** at **DatenVorsprung GmbH**

- Providing data analysis, machine learning and visualization as a service
- Focusing on trustworthy AI, verifiable neural networks and fair control systems

Since 2020 — **Artistic Director** at **Wiener Mozart Orchester Konzertveranstaltungs GmbH**

- Artistic direction of the Vienna Mozart Orchestra (around 150 concerts per year, mostly in the Golden Hall of the Musikverein in Vienna)
- Responsible for arranging concert programs, hiring new singers, soloists or conductors

Since 2019 — **Advisor** at **Absolut Ticket GmbH**

- Focusing on innovations for the cultural sector and organizers
- [Co-Advisor of a Master Thesis in our company] *„Exploratory Visual System for Predictive Machine Learning of Event-Organization Data“, Max Sbardellati, 2021.* We worked with an ML model trained on event-organization data. The goal is to create an exploratory visual event-organization system that

enables event organizers to efficiently work with the model. The main user goals in this scenario are to maximize profits and to be able to prepare for the predicted number of visitors.
- Providing a free solution for contact tracing with secure data protection to about 1500 organizers, associations and restaurants in Austria.

2017-2019 **Founder, Owner and CEO** at **Absolut Ticket GmbH**

- Responsible for the technological strategy supervising a team of web developers
- New conception and improvement of the search engine advertising automation using the user behavior data
- Responsible for the strategic direction and the business development of the company being an employer of 30 employees

2014-2017 **CEO** at **Wiener Mozart Orchester Konzertveranstaltungs GmbH**

- Responsible for the "Ticket Office" business line (Vienna Ticket Office, German Ticket Office and Italian Ticket Office)
- Continuous improvement and development of the database and website together with a programming team
- Automation of repetitive activities to enable the employees to focus on the customer-oriented work
- Conception of a new strategy including motivation, promotion and training of the employees

2009-2014 **Technical manager** at **Wiener Mozart Orchester Konzertveranstaltungs GmbH**

- 20 hours per week
- Analysis and improvement of the search engine advertising and optimization strategy
- Project manager responsible for the relaunch of the website (2013)
- Responsible for the technical support of www.mozart.co.at and www.ViennaTicketOffice.com

2006-2007 **Pianist**

- Piano Concerts in the Golden Hall of the Musikverein in Vienna

Awards and Honors

Scientia Prize **AKV Scientia Prize presented by Joerg Schmiedmayer (Wittgenstein Award Winner)**

- May 2022: Received an award from the „Altkalksburger Vereinigung" for the AAAI-22 paper „GoTube: Scalable Statistical Verification of Continuous-Depth Models".

MIT CSAIL Admission to a 4-month research stay at MIT Computer Science & Artificial Intelligence Lab in 2022

IEEE Paper Prize **TC-DES and TC-HS Outstanding Student Paper Prize**

- September 2021: Received a prestigious IEEE award for the CDC-20 paper „Lagrangian Reachtubes: The Next Generation".

FFG grant **Call 2020: Data Driven Technologies - ICT of the Future**

- We (as DatenVorsprung) got a funding for our project „Data-driven Tourism for Sustainability"
- 3-years project starting in Dec. 2021 together with 5 project partners. DatenVorsprung's goal in this project is to make fair AI-based predictions and control of visitor flow throughout multiple POIs, and minimize ecological impact caused by overcrowding. The City of Salzburg will be one of the use cases, as Tourism Salzburg is also a project partner.

INiTS **Invited as a Keynote Speaker at „Women in Health IT Breakfast" (organized by INiTS).**

- Invited Keynote Talk about *„The strength of vision: a future with safe and trustworthy AI in safety-critical environments"*
- Summary of that talk: https://dv.tik.cc/INiTS

LogiCS Admission to the doctoral program LogiCS - seeking exceptionally talented and motivated students (https://logic-cs.at/phd/admission/)

Degree with honors	**TU Wien -** Master and Bachelor title with honors
TUtheTOP	Participated in the high potential program "TUtheTOP" of the Vienna University of Technology

Press and media

TU Wien News	**Artificial Intelligence: guaranteeing predictable outcomes** https://www.tuwien.at/tu-wien/aktuelles/news/news/kuenstliche-intelligenz-vorhersagbar-machen
Factory magazine	**How to provide safe AI for industry.** https://factorynet.at/a/der-vorsprung-sicherer-daten
FWF Scilog magazine	**Project of the week and first page article in FWF Scilog magazine: *„Leading edge with secure data*"** https://scilog.fwf.ac.at/en/natur-technik/13639/leading-edge-with-secure-data
FWF annual report	**Article on „Intelligent Contact Tracing"** https://dv.tik.cc/fwf-annual-report-2020
DerStandard	**Newspaper article on „Contact-Tracing with data privacy"** https://www.derstandard.at/story/2000120183125/contact-tracing-mit-datenschutz

First and Last Author Publications

AI4TS 2022 „Deep-Learning vs Regression: Prediction of Tourism Flow with Limited Data". Lemmel J., Babaiee Z., Kleinlehner M., Majic I., Neubauer P., Scholz J., Grosu R., and **Neubauer S.A.** Accepted for publication at the IJCAI'22 workshop AI for Time Series Analysis, 2022.

Henzinger-60 2022 **„**Robustness Analysis of Continuous-Depth Models with Lagrangian Techniques". **Neubauer S.A.**, and Grosu R. Accepted for publication at Henzinger-60 (Thomas Henzinger Festschrift - Conference celebrating his 60th birthday).

AAAI 2022 **„**GoTube: Scalable Stochastic Verification of Continuous-Depth Models". **Gruenbacher S.**, Hasani R., Lechner M., Henzinger T.A., Rus D., Smolka S.A., and Grosu R. Proceedings of the AAAI Conference on Artificial Intelligence, 36(6), 2022, pages 6755-6764, https://doi.org/10.1609/aaai.v36i6.20631

AAAI 2021 **„**On the Verification of Neural ODEs with Stochastic Guarantees". **Gruenbacher S.**, Hasani R., Lechner M., Cyranka J., Smolka S.A., and Grosu R. Proceedings of the AAAI Conference on Artificial Intelligence, 35(13), 2021, pages 11525-11535, https://ojs.aaai.org/index.php/AAAI/article/view/17372

CDC 2020 **„**Lagrangian Reachtubes: The Next Generation". **Gruenbacher S.**, Cyranka J., Lechner M., Islam M.A., Smolka S.A., and Grosu R. Proceedings of the 59th IEEE Conference on Decision and Control (CDC), 2020, pages 1556-1563, https://doi.org/10.1109/CDC42340.2020.9304042

ARCH 2019 **„**Under the Hood of a Stand-Alone Lagrangian Reachability Tool". **Gruenbacher S.**, Cyranka J., Islam M.A., Tschaikowski M., Smolka S.A., and Grosu R. ARCH19. 6th International Workshop on Applied Verification of Continuous and Hybrid Systems, vol 61, 2019, pages 211--219, https://doi.org/10.29007/ns8p

Skills

Languages	**German** (native), **Portuguese** (native), **English** (fluent), **French** (very good), **Spanish** (very good)
Program languages	**Python** (very good), **C++** (very good), **PHP** (very good), **Javascript** (very good), **MySQL** (very good)
Professional Trainings	Agile Scrum, Google AdWords

Others

Social Endeavour	Organizing and supporting social work in Brazil Volunteer tutor in mathematics, German and English for unaccompanied refugees since May 2015 Providing a free solution for contact tracing with secure data protection to about 1500 organizers, associations and restaurants in Austria.
Sports	Marathon running, volleyball, climbing, scuba diving (Advanced Open Water Diver), ballroom dancing (proficiency badge “Gold-Star”), apparatus gymnastics (silver proficiency badge), riding (proficiency badge “Reiternadel”), mountain biking, hiking, skiing, ski touring
Music	Piano (Regional competition in Niederösterreich “prima la musica” 2005 in the scoring category “Jugendlicher Begleiter” with honors, Piano Concerts in the Golden Hall of the Musikverein in Vienna)