

# Stability Analysis of Switched Linear Systems with Neural Lyapunov Functions

Virginie Debauche<sup>1</sup>, Alec Edwards<sup>2</sup>, Raphaël M. Jungers<sup>1</sup>, Alessandro Abate<sup>2</sup>,

<sup>1</sup>ICTEAM Institute, UCLouvain, B-1348 Louvain-la-Neuve, Belgium

<sup>2</sup>Department of Computer Science, University of Oxford, United Kingdom

virginie.debauche@uclouvain.be, alec.edwards@worc.ox.ac.uk, raphael.jungers@uclouvain.be, alessandro.abate@cs.ox.ac.uk

## Abstract

Neural-based, data-driven analysis and control of dynamical systems have been recently investigated and have shown great promise, e.g. for safety verification or stability analysis. Indeed, not only do neural networks allow for an entirely model-free, data-driven approach, but also for handling arbitrary complex functions via their power of representation (as opposed to, e.g. algebraic optimization techniques that are restricted to polynomial functions). Whilst classical Lyapunov techniques allow to provide a formal and robust guarantee of stability of a *switched* dynamical system, very little is yet known about correctness guarantees for *Neural* Lyapunov functions, nor about their performance (amount of data needed for a certain accuracy).

We thus formally introduce neural Lyapunov functions for the stability analysis of switched linear systems: we benchmark them on this paradigmatic problem, which is notoriously difficult (and in general Turing-undecidable), but which admits recently-developed technologies and theoretical results. Inspired by switched systems theory, we provide theoretical guarantees on the representative power of neural networks, leveraging recent results from the ML community. We additionally experimentally display how neural Lyapunov functions compete with state-of-the-art results and techniques, while admitting a wide range of improvement, both in theory and in practice. This study intends to improve our understanding of the opportunities and current limitations of neural-based data-driven analysis and control of complex dynamical systems.

## Introduction

The interaction between Lyapunov theory and learning has motivated emerging research in recent years. For instance, (Berkenkamp et al. 2017) or (Fazlyab, Morari, and Pappas 2021) employ Lyapunov approaches to certify safety conditions in AI-based systems. Other works (among which the present one), motivated by the great performance of neural networks in many computational technologies, tackle the other direction and use AI techniques in order to learn Lyapunov functions (Chang, Roohi, and Gao 2020; Abate et al. 2021; Dawson et al. 2021; Farsi et al. 2022; Zhang et al. 2023). They provide promising proofs of concept for automated, data-driven control solutions that are agnostic of

the particular properties of the considered control system. Indeed, a prime advantage of neural networks is that they can be deployed in an unknown setting and on complex systems, such as Cyber-Physical Systems, where classical control techniques would require verifying technical properties (such as Lipschitz continuity, linearity, or passivity) that are out of hand in practice.

A main paradigm along these lines is a situation where a neural network is constructed from observed trajectories, and its input-output relation is interpreted as a Lyapunov function for the system. Our work falls within this paradigm. However, a crucial limitation of neural techniques is that they rarely come with guarantees; that is, the obtained network is tailored to satisfy the properties of a Lyapunov function *on the observed trajectories*, but very little is known about its generalisation properties out of the sample set. In safety-critical application, one needs generalisation guarantees, where the observed performance of the Lyapunov function is guaranteed to hold for the whole universe, not only the sampled set. Even more, Lyapunov functions must have specific properties, which might be violated by the obtained neural network (think simply of positive definiteness).

In this work, we are primarily interested in understanding this (potential) gap between high performance on the sampled set, and behaviour on the true system. For this purpose, our strategy is to study this approach on a family of complex, but still well-understood systems, for which alternative computational techniques have been developed, so that one can compare the performance of neural networks, both in terms of computational performance and in terms of accuracy. In this paper we focus on switched systems.

Switched systems have attracted massive research efforts in Systems and Control, because they provide a relatively simple framework for representing many complex engineering applications. As a few examples, they have been used in bipedal robotics (Hurmuzlu, Génot, and Brogliato 2004), in image processing (Jungers 2009, Chapter 5), or multihop control networks (Alur et al. 2011). Closer to AI, switched systems have been used to model Q-learning algorithms (Lee and He 2020), and techniques developed for classification have been used in switched system identification (Lauer and Bloch 2008).

In particular, we focus on the stability analysis problem for switched systems, which is encapsulated by the notion of

*joint spectral radius* (in short, JSR). It has been proven that it is extremely hard to compute the JSR (Blondel and Tsitsiklis 2000b), and in full generality, the problem of computing this quantity is Turing-undecidable (Blondel and Tsitsiklis 2000a). However, different computational techniques have been developed, relying essentially on Lyapunov theory, with good practical performance. One of the main families of such techniques rely on the power of Semi-Definite Programming (SDP), and provides quadratic/ellipsoidal (Blondel, Nesterov, and Theys 2005) or Sum-Of-Squares (SOS) (Parrilo and Jadbabaie 2008) Lyapunov functions. Despite the hardness of the JSR computation, these references provide approximation guarantees; that is, one can obtain an estimate of the JSR, with a bound on the error made by the estimate. See (Jungers 2009) for a whole review on this topic.

As a first theoretical contribution, we provide similar theoretical approximation guarantees for neural network-based Lyapunov functions. For this, we leverage results from convex geometry, and combine them with recent results in Machine Learning (ML), about the representation power of neural networks. Results on the approximation power of neural networks are well established, but have not previously been tailored to Lyapunov functions constructed as neural networks.

Our second contribution is empirical. It is well known that ReLU neural networks compute piecewise linear functions. Such functions are another popular technique to compute the JSR, however with less efficient computational power: the computation of a piecewise linear norm cannot be achieved efficiently since it requires to solve a bilinear program (Athanasopoulos and Jungers 2019), which cannot be done in polynomial time. ReLU neural networks offer an alternative to compute this type of Lyapunov functions. We thus provide numerical experiments in order to benchmark this alternative way of obtaining piecewise linear Lyapunov functions. We first consider a low-dimensional switched system as proof of concept where we observe that we are competitive in terms of approximation precision. Then we increase the dimension to show that we compete with SDP-based techniques both in terms of computation time and approximation precision.

Towards the end of the paper, we provide an overview of technical problems that are encountered, and investigate techniques from ML and from control to alleviate these problems.

## Notation

Given an integer  $m \in \mathbb{N}$ , the notation  $\langle m \rangle$  refers to all the integers smaller than or equal to  $m$ , i.e.  $\{1, \dots, m\}$ . Given a real number  $x \in \mathbb{R}$ ,  $\lfloor x \rfloor$  denotes the floor of  $x$ , i.e. the largest integer which does not exceed  $x$ , while  $\lceil x \rceil$  denotes the ceiling of  $x$ , i.e. the smallest integer exceeding  $x$ .

## Background

### Switched Systems and Joint Spectral Radius

In classical linear systems theory, a typical question is whether the *discrete-time linear system*  $x_{k+1} := Ax_k$  is *stable*, that is, whether all the trajectories asymptotically tend to

zero. A necessary and sufficient condition for this is that the *spectral radius* of the square matrix  $A \in \mathbb{R}^{n \times n}$  with  $n \in \mathbb{N}$  defined by

$$\rho(A) := \lim_{k \rightarrow \infty} \|A^k\|^{1/k}$$

is less than one. The spectral radius expresses the maximal rate of growth of the system, and has been proved to be equal to the maximal modulus of the eigenvalues of the matrix  $A$ .

Discrete-time linear systems can be generalised to *switched linear systems* defined by

$$x_{k+1} = A_{\sigma(k)} x_k \quad (1)$$

where the dynamics evolves at each time step according to a finite set of square matrices  $\Sigma := \{A_1, \dots, A_M\} \subset \mathbb{R}^{n \times n}$ ,  $n \in \mathbb{N}$ , and the *switching signal*  $\sigma : \mathbb{N} \rightarrow \langle M \rangle$ . Here, the switching signal can be interpreted as an exogenous perturbation (think of an operator who may switch the operating mode of a system, or any other situation where the law of dynamics may switch from time to time e.g. due to external disturbance, or change of specification) and the system is thus defined by *several* matrices rather than a single one, as is the case for classical linear systems. A switched system is (*worst-case*) *stable* if for all  $x_0 \in \mathbb{R}^n$ , for all switching signals  $\sigma(\cdot)$ ,  $x_k \rightarrow 0$  when  $k \rightarrow \infty$ .

For switched systems, one can generalise the notion of *spectral radius* (Rota and Strang 1960) to the finite set of square matrices  $\Sigma$ .

**Definition 1** (Joint spectral radius). *Given a finite set of square matrices  $\Sigma := \{A_1, \dots, A_M\} \subset \mathbb{R}^{n \times n}$  with  $n \in \mathbb{N}$ , the joint spectral radius (JSR) of  $\Sigma$ , denoted by  $\rho(\Sigma)$ , is defined by*

$$\rho(\Sigma) := \limsup_{k \rightarrow \infty} \left\{ \|A\|^{1/k} : A \in \Sigma^k \right\} \quad (2)$$

where  $\Sigma^k := \{A_{\sigma_k} := A_{\sigma_k(1)} \cdots A_{\sigma_k(k)} \mid \sigma_k : \langle k \rangle \rightarrow \langle M \rangle\}$  encodes all the possible products of length  $k$  of the matrices in  $\Sigma$ .

Similar to the linear case above, the JSR actually expresses the maximal asymptotic behaviour of a switched linear system. The joint spectral radius of  $\Sigma$  can be related to the stability of (1) by the following Theorem.

**Theorem 1** (Corollary 1.1 in (Jungers 2009)). *Given a finite set of  $M$  square matrices  $\Sigma := \{A_1, \dots, A_M\}$ . The corresponding switched linear system is stable if and only if  $\rho(\Sigma) < 1$ .*

### Approximation of the JSR

Unfortunately, the JSR does not share the same nice algebraic properties as the spectral radius and is extremely hard to compute in practice. Indeed, approximating the JSR is NP-hard (Blondel and Tsitsiklis 1997), whereas deciding whether the JSR is smaller than one or not is Turing-undecidable (Blondel and Canterini 2008), and there does not exist any algebraic criterion to decide the stability (non-algebraicity) of switched systems (Kozyakin 1990). Despite these theoretical limitations, the approximation of the JSR has been tackled by many researchers. In particular, Lyapunov methods have been exploited thanks to the following characterization of the JSR.

**Theorem 2** (Proposition 1 in (Rota and Strang 1960)). *For any bounded set of matrices  $\Sigma$  such that  $\rho(\Sigma) \neq 0$ , the joint spectral radius of  $\Sigma$  can be defined as*

$$\rho(\Sigma) := \inf_{\|\cdot\|} \max_{A \in \Sigma} \|A\|. \quad (3)$$

This result in particular tells that any norm  $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$  provides an upper-approximation of the JSR. Therefore, one can in principle compute the JSR of a finite set of matrices  $\Sigma$  by looking at all norms and their corresponding approximation  $\rho(\Sigma, \|\cdot\|) := \max_{A \in \Sigma} \|A\|$ , and keep the smallest.

In practice, one usually looks through a subset of norms for which the computation can be more easily performed. This is for instance the case for the ellipsoidal approximation of the JSR, defined as

$$\rho_{\mathcal{Q}}(\Sigma) := \inf_{P \succ 0} \max_{A \in \Sigma} \|A\|_P \quad (4)$$

where  $\|A\|_P$  denotes the matrix norm induced by the ellipsoidal vector norm associated to  $P$ , i.e.  $\|x\|_P := \sqrt{x^\top P x}$ . Note that this denomination stems from the shape of the unit ball  $\mathcal{B}_{\|\cdot\|_P}$  of this norm. From a computational point of view, the ellipsoidal approximation  $\rho_{\mathcal{Q}}$  can be computed efficiently thanks to SDP, see (Blondel, Nesterov, and Theys 2005) for details. Moreover, the following theorem provides theoretical guarantees on the accuracy of the technique.

**Theorem 3** (Theorem 14 in (Blondel, Nesterov, and Theys 2005)). *Let  $\rho(\Sigma)$  be the joint spectral radius of a finite set of matrices  $\Sigma$  of dimension  $n \in \mathbb{N}$ . Then*

$$\frac{1}{\tau_{\mathcal{Q}}} \rho_{\mathcal{Q}}(\Sigma) \leq \rho(\Sigma) \leq \rho_{\mathcal{Q}}(\Sigma), \quad (5)$$

where  $\tau_{\mathcal{Q}} := \sqrt{n}$ .

A similar SDP-based approach using SOS polynomials instead of quadratic functions has been developed (Parrilo and Jadbabaie 2008, Theorem 3.4). In this case, the SOS approximation of degree  $2d$ ,  $d \in \mathbb{N}$  of the JSR denoted by  $\rho_{SOS,2d}(\Sigma)$  satisfies the following inequalities

$$\frac{1}{\tau_{SOS}(n, d)^{1/2d}} \rho_{SOS,2d}(\Sigma) \leq \rho(\Sigma) \leq \rho_{SOS,2d}(\Sigma), \quad (6)$$

where  $\tau_{SOS}(n, d) = \binom{n+d-1}{d}$ . See (Jungers 2009) for a comprehensive review on the JSR.

### ReLU-Activated Neural Nets, CPWL Functions

We consider neural networks with ReLU activation functions.

**Definition 2** (ReLU neural network). *A Rectified Linear Units (ReLU) feedforward neural network with  $k+1 \in \mathbb{N}$  hidden layers is defined by  $k$  affine transformations  $T^{(j)} : \mathbb{R}^{n_{j-1}} \rightarrow \mathbb{R}^{n_j}$ ,  $x \mapsto W^{(j)}x + b^{(j)}$  for  $j \in \langle k \rangle$ , and a linear transformation  $T^{(k+1)} : \mathbb{R}^{n_k} \rightarrow \mathbb{R}^{n_{k+1}}$ ,  $x \mapsto W^{(k+1)}x$ . The network represents the function  $V : \mathbb{R}^{n_0} \rightarrow \mathbb{R}^{n_{k+1}}$  given by*

$$V := T^{(k+1)} \circ \sigma \circ \dots \circ T^{(2)} \circ \sigma \circ T^{(1)}, \quad (7)$$

where  $\sigma$  is the component-wise rectifier function, i.e.  $\sigma(x) = (\max\{0, x_1\}, \dots, \max\{0, x_n\})$ . The matrices  $W^{(l)}$  and the vectors  $b^{(l)}$  are respectively the weights and the biases of the  $l$ -th layer while  $n_l$  is the width of the  $l$ -th layer. The maximum width of all the hidden layers is called the width of the neural network, and the depth is  $k+1$ .

Any function represented by a ReLU neural network is a continuous piecewise affine (CPWA) function whose sub-level sets are polytopes.

**Definition 3** (Continuous piecewise affine/linear function). *We say that a function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is a continuous piecewise affine (resp. linear) function (CPWA/L function) if there exists a finite set of polyhedra whose union is  $\mathbb{R}^n$ , and furthermore if  $f$  is affine (resp. linear) over each polyhedron. The number of pieces of  $f$  is the minimal number of polyhedra necessary to express  $f$  as above.*

This function is positively homogeneous if and only if all the biases are zero, see (Hertrich et al. 2021, Proposition 2.3).

### Expressivity Power of ReLU Neural Networks

It is well-known (Hornik 1991) that ReLU neural networks can approximate arbitrarily well any continuous function, even with a single hidden layer. More recently, researchers (Arora et al. 2018; Hertrich et al. 2021; Dereich and Kassing 2021) have provided quantified approximation guarantees in terms of complexity of the represented function, as a function of the depth and width of the network. In the theorem below, the complexity of the function is expressed by the number of its pieces.

**Theorem 4** (Theorem 4.4 in (Hertrich et al. 2021)). *Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  be a CPWL function with  $p$  affine pieces. Then  $f$  can be represented by a ReLU neural network with depth  $\lceil \log_2(n+1) \rceil + 1$  and width  $\mathcal{O}(p^{2n^2+3n+1})$ .*

Note that the depth of the network depends exclusively on the input size, and not on the number of pieces.

### Problem Statement

As a paradigm of safety constraint, we now study the ability of neural networks to provide guarantees for the stability of a given switched linear system. In view of Theorem 4 above, we are interested in approximating the JSR of a finite set of matrices using continuous piecewise linear (CPWL) functions, which we call a *CPWL approximation of the JSR*.

**Problem 1.** *Can one provide a guarantee on the accuracy of the best neural approximation of the JSR, that is, on the bound obtained from (3) when the norms  $\|\cdot\|$  are restricted to be of the form (7)? What is the relation between such a guarantee and the number of layers, of neurons, of the width of the network? How can one guarantee that the output  $V$  of the network is indeed a valid norm?*

### Theoretical Results

#### CPWL Approximation Guarantees

Inspired by the approximation results in (5) and (6) we exploit Theorem 2, additionally leveraging the following result

from Barvinok about the approximation power of a convex set by a polytope as a function of the dimension and the number of vertices in the approximation polytope.

**Theorem 5** (Theorem 1.1 in (Barvinok 2013)). *Let  $\tau > 1$  be a real number and  $n$  and  $k$  be positive integers and such that*

$$\left(\tau - \sqrt{\tau^2 - 1}\right)^k + \left(\tau + \sqrt{\tau^2 - 1}\right)^k \geq 6 D(n, k)^{1/2}, \quad (8)$$

where

$$D(n, k) := \sum_{m=0}^{\lfloor k/2 \rfloor} \binom{n+k-1-2m}{k-2m}. \quad (9)$$

Then, for any symmetric, compact and convex set  $K \subset \mathbb{R}^n$  with non-empty interior and containing the origin, there exists a symmetric polytope  $P \subset \mathbb{R}^n$  with at most  $8D(n, k)$  vertices such that

$$P \subseteq K \subseteq \tau P. \quad (10)$$

We can now derive approximation guarantees on the CPWL approximation of the JSR.

**Theorem 6.** *Let  $\rho(\Sigma)$  be the joint spectral radius of a finite set of matrices  $\Sigma$  of dimension  $n \in \mathbb{N}$ . For any  $\tau > 1$  for which there exists  $k_\tau \in \mathbb{N}$  such that relation (8) is satisfied, the following relation holds :*

$$\frac{1}{\tau} \rho_{\mathcal{P}}(\Sigma) \leq \rho(\Sigma) \leq \rho_{\mathcal{P}}(\Sigma), \quad (11)$$

where  $\rho_{\mathcal{P}}$  is the optimal solution of (3) where the norms are restricted to CPWL norms with at most  $8D(n, k)$  vertices.

*Proof.* Consider a finite set of  $m$  matrices  $\Sigma := \{A_1, \dots, A_m\} \subset \mathbb{R}^{n \times n}$  of dimension  $n \in \mathbb{N}$ , and  $\rho(\Sigma)$  its joint spectral radius. By Theorem 2 and by definition of an infimum, for any value  $\varepsilon > 0$ , there exists an  $\varepsilon$ -norm  $\|\cdot\|_\star$  such that  $\forall i = 1, \dots, m$

$$\forall x \in \mathbb{R}^n, \|A_i x\|_\star \leq (\rho(\Sigma) + \varepsilon) \|x\|_\star.$$

This norm defines a convex set  $K \subset \mathbb{R}^n$  (which contains the origin) that can be approximated by a polytope. By Theorem 5, for any positive integer  $d$  and any real number  $\tau > 1$  satisfying (8), there exists a symmetric polytope  $P \subset \mathbb{R}^n$  with at most  $8D(n, k)$  vertices such that

$$P \subseteq K \subseteq \tau P.$$

Therefore, the homogeneous function  $V(x)$  whose 1-level set is  $P$ , satisfies that for all  $x \in \mathbb{R}^n$ ,

$$V(x) \leq \|x\|_\star \leq \tau V(x),$$

and then for any  $i = 1, \dots, m$  and any  $x \in \mathbb{R}^n$ :

$$\begin{aligned} V(A_i x) &\leq \|A_i x\|_\star, \\ &\leq (\rho(\Sigma) + \varepsilon) \|x\|_\star, \\ &\leq (\rho(\Sigma) + \varepsilon) \tau V(x). \end{aligned}$$

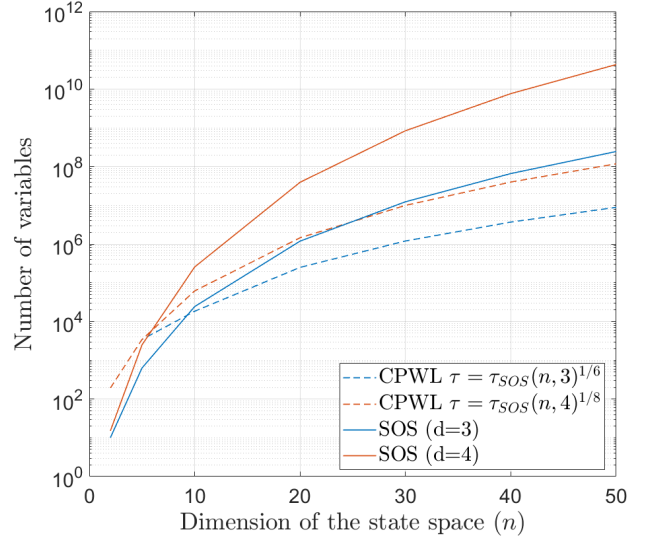


Figure 1: Evolution, as a function of the dimension ( $n$ ), of the number of variables required to achieve precision values  $\tau$  on the JSR approximation with the CPWL approach (dashed line) and with the SOS approach (continuous line). In blue,  $\tau = \tau_{SOS}(n, 3)$ , in red,  $\tau = \tau_{SOS}(n, 4)$ . These values are the minimal required accuracy to outperform the guarantee (6) for, respectively, degree-3 and -4 SOS. One can see that for large values of  $n$ , the CPWL approximation needs far fewer variables, for any given precision.

Then,  $V(A_i) \leq (\rho(\Sigma) + \varepsilon)\tau$  for all  $i = 1, \dots, m$  and for any  $\varepsilon > 0$ . Therefore, at the worst case, we have that

$$\frac{1}{\tau} \rho_{\mathcal{P}}(\Sigma) \leq \rho(\Sigma) \leq \rho_{\mathcal{P}}(\Sigma),$$

where the second inequality is direct since we consider a subset of norms.  $\square$

Using this theorem, we can compute an upper-bound on the number of variables required to achieve a given precision  $\tau$  as the number of vertices multiplied by the dimension of the state space. In Figure 1, we show the evolution with the dimension of this number of variables, for different values of  $\tau$ . For the sake of comparing with the performance of the SOS approximations of degree 3 and 4, we selected  $\tau = \tau_{SOS}(n, d)$  for  $d = 3$  and  $d = 4$ . These results show that for small dimensions, the CPWL approach requires slightly more variables than the SDP solution. However, when we consider higher dimensions, the trend is reversed. These observations motivate the CPWL approach rather than the polynomial approach since we need less variables in high dimension. However, computation methods rapidly suffer from the curse of dimensionality: the complexity is polynomial for SDPs while it is exponential for the CPWL norms, which necessitate solving a bilinear program as explained previously. For this reason, we now investigate generating CPWL approximations with neural networks.

## Bounds on the Structure of the Network

Theorem 6 provides a relation between the CPWL approximation precision and the number of vertices of its polytopic sublevel sets. Now, combining it with Theorem 4, we are able to derive bounds on the network structure (i.e. its depth and its width) to be able to achieve this precision.

To do so, we first need to bound the number of faces of a polytope, i.e. the number of pieces of the corresponding CPWL function, given the number of vertices.

**Theorem 7** ((McMullen 1970)). *Let  $P$  be a polytope in dimension  $n \in \mathbb{N}$  with  $k$  vertices. Then,  $P$  has at most*

$$\binom{k - \lfloor \frac{n+1}{2} \rfloor}{k-n} + \binom{k - \lfloor \frac{n+2}{2} \rfloor}{k-n}$$

faces.

We are finally able to state the following theorem, which provides original theoretical bounds on the structure of a ReLU neural network to provide a CPWL approximation of the JSR with a given precision  $\tau$ . To the best of our knowledge, this is the first universal approximation result tailored to neural Lyapunov functions.

**Theorem 8.** *Let  $\rho(\Sigma)$  be the joint spectral radius of a finite set of matrices  $\Sigma$  of dimension  $n$ . For any real  $\tau > 1$  for which there exists  $k_\tau \in \mathbb{N}$  satisfying relation (8), there exists a CPWL function represented by a ReLU neural network of depth*

$$\lceil \log_2(n+1) \rceil + 1$$

and width

$$\mathcal{O} \left( \left[ \binom{D_n^\tau - \lfloor \frac{n+1}{2} \rfloor}{D_n^\tau - n} + \binom{D_n^\tau - \lfloor \frac{n+2}{2} \rfloor}{D_n^\tau - n} \right]^{2n^2+3n+1} \right)$$

where  $D_n^\tau := 8D(n, k_\tau)$  as defined in (9), which approximates  $\rho(\Sigma)$  with a precision of  $\tau$ .

*Proof.* This theorem results from the successive application of Theorems 6, 7 and 4.  $\square$

## Experimental Evaluation

The above theoretical results provide a proof of concept that neural network can approximate the JSR up to an a priori fixed guarantee of accuracy. However, all the bounds above are on the worst case, and we now provide an empirical investigation of the practical efficiency of the neural approach, in comparison with the more classical SDP-based approach.

### Setup

Given a switched linear system (1) in dimension  $n$ , we consider a ReLU neural network without biases. This is without loss of generality, since we may restrict ourself to homogeneous functions  $V$ . We train the network with a sample set  $\mathcal{S}$  of data points uniformly distributed on the unit sphere  $\mathcal{S}^{n-1} := \{x \in \mathbb{R}^n : \|x\|_2 = 1\}$  (by linearity and homogeneity, it is sufficient to consider sample points on the

sphere). We structurally enforce during training the positivity of the represented function  $V(\cdot)$  by imposing nonnegativity for the output weights. We consider the loss function defined by

$$L(V, \mathcal{S}) := \max_{i=1, \dots, M} \max_{x \in \mathcal{S}} \frac{V(A_i x)}{V(x)}, \quad (12)$$

which provides a sample-based approximation of (3) where the norms are of the form  $V(\cdot)$ , as represented by the network. Therefore, given a ReLU neural network with  $k$  hidden layers and  $m$  neurons in each of them, the *neural approximation* of the JSR denoted by  $\rho_{NN(k,m)}(\Sigma)$  is defined as the lowest value of the loss function during the entire training campaign. Note that  $\rho_{NN(k,m)}(\Sigma)$  depends on the sampled set  $\mathcal{S}$  and for this reason, the quantity  $\max_{x \in \mathcal{S}} \frac{V(A_i x)}{V(x)}$  is not formally a norm and thus nothing guarantees that the network provides a formal upper bound on the JSR (it would be if  $\mathcal{S}$  was replaced by  $\mathbb{R}^n$ , in Equation (12) above).

## Numerical Results

As reference benchmarks, we consider two switched linear systems in dimension 2 and 8 respectively, which are challenging as they are known to lead to poor approximation with classical techniques (see (Debauche, Della Rossa, and Jungers 2023, Section 5) and (Ahmadi et al. 2014, Example 5.2)). The first one serves as proof of concept, while the second example shows that in higher dimensions, neural Lyapunov functions are competitive with SOS-based techniques both in terms of precision and (as suggested earlier in Figure 1) in computation time. Note that all experiments were run on an Intel i7 laptop with 4 cores and 8GB of RAM.

**Example 1** (Low dimension). *We consider the switched linear system  $\Sigma_2 := \{A_1, A_2\} \subset \mathbb{R}^{2 \times 2}$  defined by*

$$A_1 = \begin{bmatrix} 1.5519 & 0.4474 \\ 7.6412 & 7.4716 \end{bmatrix} \text{ and } A_2 = \begin{bmatrix} 0.4750 & 9.1755 \\ 1.8955 & 0.1850 \end{bmatrix}. \quad (13)$$

		Neural approximation		
		$\rho_{NN(k,m)}(\Sigma_2)$		
$k$	$m$	Best	Mean	Std.
1 layer	5 neurons	8.6977	9.0251	0.8800
	10 neurons	8.6910	8.6969	0.0056
2 layers	5 neurons	8.6983	8.9312	0.4645
	10 neurons	8.6944	8.7049	0.0077
3 layers	5 neurons	8.6967	9.1984	0.7293
	10 neurons	8.6946	8.7130	0.0175

Table 1: Best and mean/std. (over 20 seeds) approximation of the JSR of system (13) provided by a ReLU neural networks with different architectures.

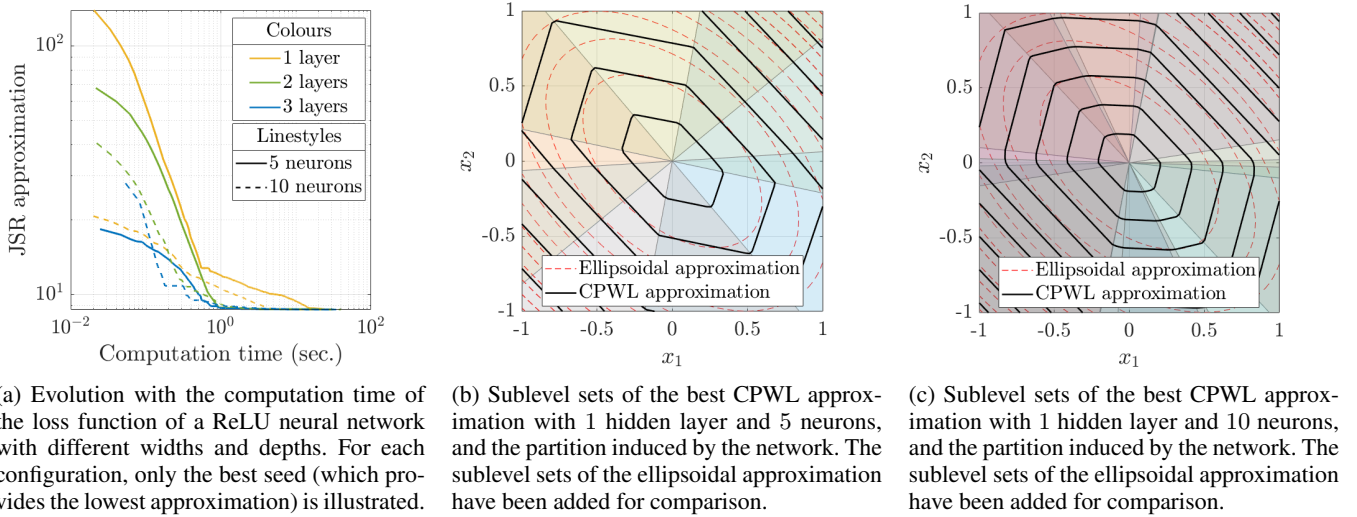


Figure 2: Approximation of the JSR of system (13) using a ReLU neural network with 500 sample points, alongside sublevel set of the best approximation from two networks. With 5 neurons (2b), the CPWL approximation appears geometrically similar to the ellipsoidal approximation. As more neurons are added (2c), the geometry of the sublevel sets diverges from the ellipsoidal approximation, corresponding to an improved approximation of the JSR (Table 1).

The joint spectral radius of  $\Sigma_2$  is 8.6881, the ellipsoidal approximation  $\rho_{\mathcal{Q}}(\Sigma_2)$  is 9.5868 and the SOS approximation of degree 4, i.e.  $\rho_{SOS,4}(\Sigma_2)$ , is 8.7203. We consider a ReLU neural network with different depths and widths that we train with 500 sample points for 20 different seeds. The best (i.e. the smallest) approximation, the mean and the standard deviation are summarized in Table 1.

Not surprisingly, the more neurons there are, the better the approximation. We also observe that with more neurons, there is significantly less variability with respect to the seed. In terms of approximation precision, the neural approach is more efficient than both the ellipsoidal and the SOS approaches, since the average approximation with 10 neurons is smaller than  $\rho_{\mathcal{Q}}(\Sigma_2)$  and  $\rho_{SOS,4}(\Sigma_2)$ . Figures 2b and 2c illustrate the partition and the sublevel sets of the Lyapunov function encoded by the best 1-layer network with 5 and 10 neurons respectively, and one can notice the similarity with the ellipsoidal sublevel sets. However, the computation of the ellipsoidal and the SOS approximation are very fast in such low dimension (they clock 0.2816 and 1.5156 seconds respectively), and the novel neural network approach cannot compete with them: for example, Figure 2a shows that in the best case, we need a few seconds, but in the worse cases, we need almost one minute.

Let us now consider a 8-dimensional switched system, such that the computation time of the ellipsoidal and SOS approximation are larger and the network starts to be competitive.

**Example 2 (High dimension).** We consider a switched system in dimension 8 with 8 different modes, defined by the matrices  $\Sigma_8 := \{A_i\} \subset \{0, 1\}^{8 \times 8}$  such that for  $i = 2, \dots, 8$ ,

$$A_i(k, l) := \begin{cases} -1 & \text{if } k = l = i, \\ 1 & \text{if } l = i \text{ and } k \neq i, \\ 0 & \text{otherwise,} \end{cases} \quad (14)$$

while the matrix  $A_1 = \mathbf{I}e_1^\top$ . One can prove that the joint spectral radius of this finite set of matrices is 1, i.e.  $\rho(\Sigma_8) = 1$ . Regarding the approximation of the JSR using classical techniques, the ellipsoidal approximation generates  $\rho_{\mathcal{Q}}(\Sigma_8) = 2.4286$  but the computation is relatively fast ( $\approx 2.5$  seconds), while the SOS approximation is better since  $\rho_{SOS,4}(\Sigma_8) = 1.0006$  but it requires much more computation time ( $\approx 258$  seconds).

For the neural approximation of the JSR, we use a ReLU neural network with a single hidden layer and different numbers of neurons: 10, 15 and 30. We train it using 500 sample points. Figure 3 shows the evolution of the mean and the min-max area of the JSR approximation with the com-

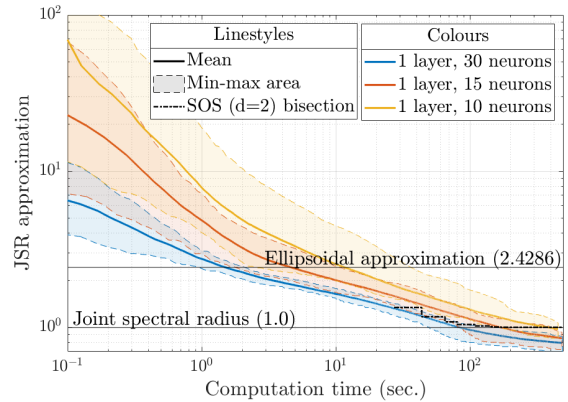
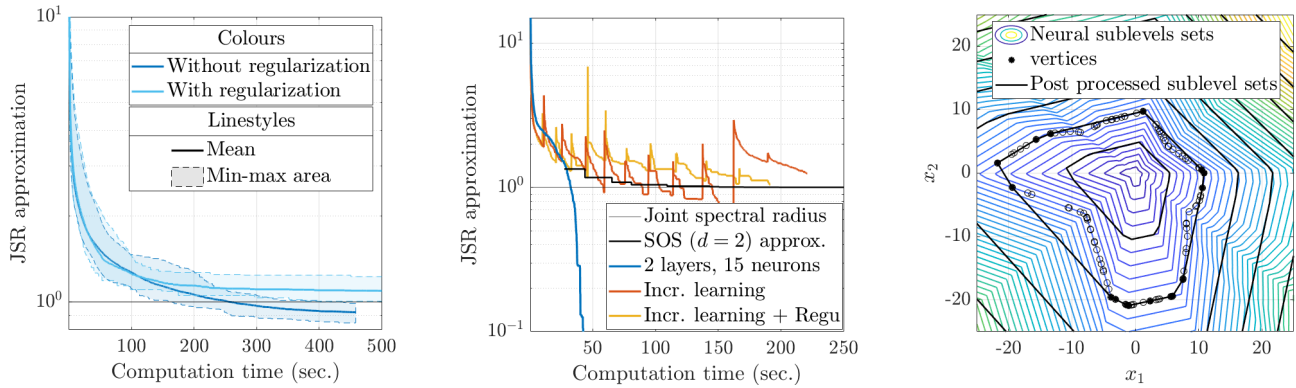


Figure 3: Evolution of the JSR approximation for the 8-dimensional system (14) provided by a ReLU neural network with different number of hidden layers and different numbers of neurons. The value of the JSR, the ellipsoidal approximation and the SOS degree-4 (in black) bisection have been added for comparison.



(a) Comparison of the evolution with the computation time of the JSR approximation provided by a ReLU network with 1 hidden layer and 15 neurons with and without L1-regularization.

(b) Comparison of the evolution with the computation time of the JSR approximation provided by a ReLU network with 1 hidden layer and 15 neurons with and without *incremental learning*.

(c) Comparison of the sublevel sets of the output of a 2-hidden-layer neural network with 10 neurons ( $\rho_{NN(2,10)}(\Sigma_2) = 8.5788$ ) trained with 100 sample points and the *post processed* norm  $N(\cdot)$  ( $\rho_N(\Sigma_2) = 9.4157$ ).

Figure 4: Techniques to avoid overfitting in higher dimensions.

putation time for each configuration. One can see that in dimension 8, the neural approach is almost as fast as the SOS ( $d = 2$ ) method. However, the network overfits the data and then provides an approximation which is smaller than the true JSR value, except with 10 neurons where half of the seeds provide an approximation larger than 1. One way to prevent this behaviour is to consider more sample points. However, the computation time increases as the sample set grows and we are no longer competitive with SDP-based techniques.

### Techniques for Improved Performance

As we have seen, the numerical experiments suffer from classical issues in ML such as overfitting, large number of sample points needed, etc. In this section, we consider some methods to mitigate these behaviours.

**Regularization** In Figure 3, we have seen that neural networks tend to overfit the data resulting in an incorrect estimate of the JSR. One classical method to prevent overfitting is regularization, i.e. for the problem at hand we penalize the network by adding a new term in the loss function to prevent it from learning overly-complex functions. Figure 4a shows promising results for system (14) using L1-regularization.

**Incremental Learning** The number of sample points is a key parameter since it determines a trade-off between computation time and the risk of overfitting. However, it seems that the network does not need many points at the beginning, but progressively requires more and more points during training. Figure 4b illustrates the evolution of the JSR approximation of system (14) provided by a 2-hidden-layer neural network with 15 neurons, where we now add successively new sample points to the training set. In this example, after combining the incremental learning with the regularization, the resulting network did not reach a wrong over-approximation value.

**Post Processing** Until now we have seen with Examples 1 and 2 that neural networks can approximate well the JSR,

but if badly tuned, they may under-estimate it. In safety-critical applications, we want a firm guarantee of stability: for this, we can leverage the information learnt by the network on the sample set  $\mathcal{S}$  into an actual norm  $\|\cdot\|_N$ , which provides a valid upper bound, at the price of a more conservative estimate of the JSR. One way to do that is to consider the absolutely homogeneous function whose unit ball is defined by

$$\overline{\mathcal{B}}_{\|\cdot\|_N} := \text{conv} \left\{ \frac{x_i}{V(x_i)} : x_i \in \mathcal{S} \right\},$$

such that  $V$  and the induced norm  $\|\cdot\|_N$  coincide on the sample set. See Figure 4c for an example with system (13).

### Conclusions

Motivated by recent developments in neural Lyapunov techniques, we have introduced for it a benchmark application and a theoretical framework, for the particular case of switched systems. These are a family of popular models for complex Cyber-Physical systems, that is algorithmically challenging, but rather well understood from a theoretical standpoint.

We have shown that one can determine theoretical bounds for the accuracy of neural Lyapunov functions, as a function of the parameters of the network. These guarantees are competitive with classical SDP-based Lyapunov approaches in terms of number of variables. From the empirical point of view, we have shown that in practice as well, the approach is competitive while our neural networks were trained on simple personal computers, which leaves an important room for improvement. We have emphasized the problem of overfitting, and proposed several avenues for mitigating it.

We see much possible further work, among which confirming the experiments at a more general level (different network architectures, different benchmark examples), potentially improving the learning algorithm (in particular, the loss function), and pushing further the approaches for mitigating overfitting.

## Acknowledgments

RJ is a FNRS honorary Research Associate. This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme under grant agreement No 864017 - L2C. Alec is supported by the EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems (EP/S024050/1).

## References

- Abate, A.; Ahmed, D.; Edwards, A.; Giacobbe, M.; and Perruffo, A. 2021. FOSSIL: A Software Tool for the Formal Synthesis of Lyapunov Functions and Barrier Certificates Using Neural Networks. In *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*, HSCC ’21. New York, NY, USA: Association for Computing Machinery.
- Ahmadi, A. A.; Jungers, R. M.; Parrilo, P. A.; and Roozbehani, M. 2014. Joint Spectral Radius and Path-Complete Graph Lyapunov Functions. *SIAM Journal on Control and Optimization*, 52(1): 687–717.
- Alur, R.; D’Innocenzo, A.; Johansson, K. H.; Pappas, G. J.; and Weiss, G. 2011. Compositional Modeling and Analysis of Multi-Hop Control Networks. *IEEE Transactions on Automatic Control*, 56(10): 2345–2357.
- Arora, R.; Basu, A.; Mianjy, P.; and Mukherjee, A. 2018. Understanding Deep Neural Networks with Rectified Linear Units. In *International Conference on Learning Representations*.
- Athanasopoulos, N.; and Jungers, R. M. 2019. Polyhedral Path-Complete Lyapunov Functions. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 3399–3404.
- Barvinok, A. 2013. Thrifty Approximations of Convex Bodies by Polytopes. *International Mathematics Research Notices*, 2014(16): 4341–4356.
- Berkenkamp, F.; Turchetta, M.; Schoellig, A. P.; and Krause, A. 2017. Safe Model-based Reinforcement Learning with Stability Guarantees. arXiv:1705.08551.
- Blondel, V.; and Canterini, V. 2008. Undecidable Problems for Probabilistic Automata of Fixed Dimension. *Theory of Computing Systems*, 36: 231–245.
- Blondel, V.; Nesterov, Y.; and Theys, J. 2005. On the accuracy of the ellipsoid norm approximation of the joint spectral radius. *Linear Algebra and its Applications*, 394: 91–107.
- Blondel, V.; and Tsitsiklis, J. 2000a. The Boundedness of All Products of a Pair of Matrices is Undecidable. *Systems & Control Letters*, 41: 135–140.
- Blondel, V. D.; and Tsitsiklis, J. N. 1997. When is a pair of matrices mortal? *Information Processing Letters*, 63(5): 283–286.
- Blondel, V. D.; and Tsitsiklis, J. N. 2000b. A survey of computational complexity results in systems and control. *Automatica*, 36(9): 1249–1274.
- Chang, Y.-C.; Roohi, N.; and Gao, S. 2020. Neural Lyapunov Control. In *arXiv:2005.00611 [Cs, Eess, Stat]*.
- Dawson, C.; Qin, Z.; Gao, S.; and Fan, C. 2021. Safe Non-linear Control Using Robust Neural Lyapunov-Barrier Functions. In *Conference on Robot Learning*.
- Debauche, V.; Della Rossa, M.; and Jungers, R. 2023. Characterization of the Ordering of Path-Complete Stability Certificates with Addition-Closed Templates. In *Proceedings of the 26th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC ’23. New York, NY, USA: Association for Computing Machinery.
- Dereich, S.; and Kassing, S. 2021. On minimal representations of shallow ReLU networks. *Neural networks : the official journal of the International Neural Network Society*, 148: 121–128.
- Farsi, M.; Li, Y.; Yuan, Y.; and Liu, J. 2022. A Piecewise Learning Framework for Control of Unknown Nonlinear Systems with Stability Guarantees. In *Learning for Dynamics and Control Conference, L4DC 2022, 23-24 June 2022, Stanford University, Stanford, CA, USA*, volume 168 of *Proceedings of Machine Learning Research*, 830–843. PMLR.
- Fazlyab, M.; Morari, M.; and Pappas, G. J. 2021. An introduction to neural network analysis via semidefinite programming. In *2021 60th IEEE Conference on Decision and Control (CDC)*, 6341–6350. IEEE.
- Hertrich, C.; Basu, A.; Di Summa, M.; and Skutella, M. 2021. Towards Lower Bounds on the Depth of ReLU Neural Networks. In *Advances in Neural Information Processing Systems*, volume 34, 3336–3348. Curran Associates, Inc.
- Hornik, K. 1991. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2): 251–257.
- Hurmuzlu, Y.; Génot, F.; and Brogliato, B. 2004. Modeling, Stability and Control of Biped Robots A General Framework. *Automatica*, 40: 1647–1664.
- Jungers, R. 2009. *The Joint Spectral Radius: Theory and Applications*, volume 385 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag.
- Kozyakin, V. 1990. Algebraic unsolvability of a problem on the absolute stability of desynchronized systems. *Automation and Remote Control*, 51: 754–759.
- Lauer, F.; and Bloch, G. 2008. Switched and PieceWise Nonlinear Hybrid System Identification.
- Lee, D.; and He, N. 2020. A Unified Switching System Perspective and Convergence Analysis of Q-Learning Algorithms. In *Advances in Neural Information Processing Systems*, volume 33, 15556–15567. Curran Associates, Inc.
- McMullen, P. 1970. The maximum numbers of faces of a convex polytope. *Mathematika*, 17(2): 179–184.
- Parrilo, P. A.; and Jadbabaie, A. 2008. Approximation of the joint spectral radius using sum of squares. *Linear Algebra and its Applications*, 428(10): 2385–2402. Special Issue on the Joint Spectral Radius: Theory, Methods and Applications.
- Rota, G.-C.; and Strang, W. G. 1960. A note on the joint spectral radius. In *Proceedings of the Netherlands Academy*, 379–381.

Zhang, S.; Xiu, Y.; Qu, G.; and Fan, C. 2023. Compositional Neural Certificates for Networked Dynamical Systems. In *Learning for Dynamics and Control Conference, LADC 2023, 15-16 June 2023, Philadelphia, PA, USA*, volume 211 of *Proceedings of Machine Learning Research*, 272–285. PMLR.