

Compositionally Verifiable Vector Neural Lyapunov Functions for Stability Analysis of Interconnected Nonlinear Systems

Jun Liu, Yiming Meng, Maxwell Fitzsimmons, and Ruikun Zhou

Abstract—While there has been increasing interest in using neural networks to compute Lyapunov functions, verifying that these functions satisfy the Lyapunov conditions and certifying stability regions remain challenging due to the curse of dimensionality. In this paper, we demonstrate that by leveraging the compositional structure of interconnected nonlinear systems, it is possible to verify neural Lyapunov functions for high-dimensional systems beyond the capabilities of current satisfiability modulo theories (SMT) solvers using a monolithic approach. Our numerical examples employ neural Lyapunov functions trained by solving Zubov’s partial differential equation (PDE), which characterizes the domain of attraction for individual subsystems. These examples show a performance advantage over sums-of-squares (SOS) polynomial Lyapunov functions derived from semidefinite programming.

Index Terms—Learning, formal verification, neural networks, nonlinear systems, stability analysis, interconnection, curse of dimensionality

I. INTRODUCTION

One of the longstanding challenges in nonlinear control is the construction of Lyapunov functions for stability analysis and controller design. Since Lyapunov’s seminal paper over a century ago [25], there has been an ongoing quest for constructive methods for the design of Lyapunov functions. Both analytical [15], [33] and computational [11], [12] strategies have been explored.

Among computational approaches for finding Lyapunov functions, sums-of-squares (SOS) techniques have perhaps received the most attention [16], [28]–[30], [35], [36]. Not only can these techniques provide local stability analysis, but they can also offer regions of attraction estimates [28], [35], [36]. Using semidefinite programming (SDP), the region of attraction can be enlarged by expanding the level set of a certain “shape function” contained in the region of attraction estimate. While this is computationally appealing, the selection of such shape functions beyond the obvious choices of norm functions remains somewhat ad hoc [18].

In recent years, progress in machine learning and neural networks has begun to transform the realm of computational studies. Many authors have investigated the use of neural

networks for computing Lyapunov functions (see, e.g., [1], [7], [9], [13], [14], [17], [39], and [8] for a recent survey). Perhaps one of the most notable differences between SDP-based synthesis of SOS Lyapunov functions and the training of neural Lyapunov functions is that neural network Lyapunov functions are not always guaranteed to be Lyapunov functions. Subsequent verification is required to ensure they satisfy Lyapunov conditions, e.g., using satisfiability modulo theories (SMT) solvers [2], [6]. This process is time-consuming, especially when seeking a maximal Lyapunov function [22], as accurate numerical verification becomes more challenging near the stability region’s boundary. A neural network Lyapunov function offers the advantage of universal approximation and embracing non-convex optimization, unlike the SDP’s limitation to convex optimization. It can sometimes capture regions of attraction better and is capable of dealing with non-polynomial dynamics. Moreover, it can leverage the abundant machine learning infrastructure, including neural network architectures, optimization algorithms, and graphics processing unit (GPU) computing. Overcoming the curse of dimensionality in verifying neural Lyapunov functions remains a significant challenge.

In this work, we demonstrate that by exploiting the compositional structure of interconnected nonlinear systems, where each subsystem admits a neural Lyapunov function in the absence of interaction, one can verify a vector neural Lyapunov function for the interconnected high-dimensional system to determine a region of attraction. This approach can address problems beyond the capabilities of current SMT solvers [10] using a monolithic approach. Specifically, we conduct numerical experiments using neural Lyapunov functions trained by solving Zubov’s partial differential equation (PDE), which characterizes the domain of attraction for individual subsystems [17], [22], [40]. We show that these functions outperform SOS polynomial Lyapunov functions obtained through semidefinite programming in terms of capturing the region of attraction of the interconnected system. An extended version of the paper, including an appendix for the omitted technical details, can be found in [23].

II. PRELIMINARIES

A. Interconnected system

Consider a network of nonlinear systems of the form

$$\dot{x}_i = f_i(x_i) + \sum_{j \neq i} G_{ij}(x_i, x_j), \quad (1)$$

where each $f_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{n_i}$ and $G_{ij} : \mathbb{R}^{n_i} \times \mathbb{R}^{n_j} \rightarrow \mathbb{R}^{n_i}$ are assumed to be locally Lipschitz and $i, j \in \{1, \dots, l\}$,

This research was supported in part by an NSERC Discover Grant and the Canada Research Chairs program. This research was enabled in part by support provided by the Digital Research Alliance of Canada (alliance.ca).

Jun Liu, Maxwell Fitzsimmons, and Ruikun Zhou are with the Department of Applied Mathematics, Faculty of Mathematics, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada. Emails: j.liu@uwaterloo.ca, mfitzsimmons@uwaterloo.ca, ruikun.zhou@uwaterloo.ca

Yiming Meng is with the Coordinated Science Laboratory, University of Illinois Urbana-Champaign, Urbana, IL 61801, USA. Emails: ymmeng@illinois.edu, yiming.meng@uwaterloo.ca

which indicates the collection of subsystems in the network. It is assumed that $f_i(0) = 0$ and $G_{ij}(0, 0) = 0$. Hence, the origin is an equilibrium point for each individual subsystem in the absence of the interconnection terms G_{ij} and for the overall interconnected system.

We use $x = (x_1, \dots, x_l) \in \mathbb{R}^{n_1} \times \dots \times \mathbb{R}^{n_l} = \mathbb{R}^N$, where $N = n_1 + n_2 + \dots + n_l$, to denote the state vector of the networked system (1) and $\phi(t, x)$ to indicate the solution to (1) starting from the initial condition $\phi(0) = x$. We also use $\phi_i(t, x)$ to denote the solution trajectory of the i th subsystem.

Definition 1 (Domain of Attraction): Suppose that the origin is asymptotically stable for (1), the *domain of attraction* of the origin for (1) is defined as

$$\mathcal{D} := \left\{ x \in \mathbb{R}^N : \lim_{t \rightarrow \infty} \phi(t, x) = 0 \right\}.$$

Any invariant subset of \mathcal{D} is called a *region of attraction*.

B. Problem formulation

A common approach to computing regions of attraction is using sub-level sets of Lyapunov functions. Suppose that we can compute a Lyapunov function for each individual subsystem in the absence of the interconnection terms G_{ij} , namely subsystems of the form

$$\dot{x}_i = f_i(x_i), \quad i = 1, \dots, l, \quad (2)$$

such that sub-level sets of the form

$$\mathcal{V}_i(v_i) := \{x_i \in \mathbb{R}^{n_i} : V_i(x_i) \leq v_i\} \quad (3)$$

are verified regions of attraction for individual subsystems in (2), where v_1, \dots, v_l are positive constants.

Clearly, the absence of the interconnection, i.e., $G_{ij} \equiv 0$, implies that the set $\Omega_1 \times \dots \times \Omega_l$ is a region of attraction for (1). However, with interconnection, this is no longer the case. The main objective of this paper is to formulate compositionally verifiable conditions to certify regions of attraction for the interconnected system (1).

Problem 1: Given regions of attraction for individual subsystems provided by (3), verify regions of attraction for the interconnected system (1).

III. STABILITY AND REACHABILITY ANALYSIS USING VECTOR LYAPUNOV FUNCTIONS

Consider a vector Lyapunov function $V : \mathbb{R}^N \rightarrow \mathbb{R}^l$ by

$$V(x) = (V_1(x_1), \dots, V_l(x_l)), \quad (4)$$

where $V_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R}$ are scalar Lyapunov functions for subsystems in (2).

A. Local stability analysis

We assume that each f_i is continuously differentiable and the origin is exponentially stable for each individual subsystem (2). Let $A_i = Df_i(0)$, where Df_i is the Jacobian of f_i . By our assumption, A_i is a Hurwitz matrix. Let $Q_i > 0$ be any positive definite matrix and let $P_i > 0$ solve the Lyapunov equation

$$P_i A_i + A_i^T P_i = -Q_i. \quad (5)$$

Rewrite (2) as

$$\dot{x}_i = A_i x_i + g_i(x_i), \quad (6)$$

where $g_i(x_i) = f_i(x_i) - A_i x_i$ satisfies $\lim_{x_i \rightarrow 0} \frac{\|g_i(x_i)\|}{\|x_i\|} = 0$.

We can analyze local stability of the interconnected system (1) using the following proposition.

Proposition 1: Let P_i and Q_i satisfy (5). For each $i = 1, \dots, l$ and $p > 0$, define the set

$$\mathcal{P}_i(p) := \{x \in \mathbb{R}^{n_i} : x^T P_i x \leq p\}.$$

Suppose that there exists a positive vector $c = (c_1, \dots, c_l) \in \mathbb{R}^l$ and a matrix of nonnegative elements $R = (r_{ij})$ such that the following inequalities hold:

$$\|P_i Dg_i(x_i)\| \leq r_{ii}, \quad (7)$$

$$\|P_i D G_{ij}(x_i, x_j)\| \leq r_{ij}, \quad (8)$$

for all $x_i \in \mathcal{P}_i(c_i)$ and $x_j \in \mathcal{P}_j(c_j)$. Define $\Lambda = (\lambda_{ij})$ by

$$\lambda_{ii} = -\frac{\lambda_{\min}(Q_i)}{\lambda_{\max}(P_i)} + \frac{2(r_{ii} + \sum_{j \neq i} r_{ij})}{\lambda_{\min}(P_i)}, \quad (9)$$

$$\lambda_{ij} = \frac{r_{ij}}{\lambda_{\min}(P_j)}, \quad j \neq i. \quad (10)$$

If there exists a positive vector $p = (p_1, \dots, p_l) \in \mathbb{R}^l$ such that $p \leq c$ and $\Lambda p < 0$ (componentwise), then the set

$$\mathcal{P}(p) := \mathcal{P}_1(p_1) \times \dots \times \mathcal{P}_l(p_l),$$

is a region of attraction for the interconnected system (1).

The proof can be found in the appendix of [23].

Remark 1: While we assumed that G_{ij} takes the form in (1), a more general form of interconnection, $G(x)$, is permissible for the analysis in Proposition 1 to proceed. We opted for the summation form to exploit the compositional nature of (7) and (8). If a decomposition is available through other inequality estimates, the results of Proposition 1 can be applied straightforwardly. Moreover, the gain bounds (9) and (10) may be further improved (see the Appendix of [23]).

Remark 2: The condition that there exists some $p > 0$ such that $\Lambda p < 0$ or $(-\Lambda)p > 0$ is one of many conditions equivalent to $-\Lambda$ being a nonsingular M -matrix (condition K_{33} in [31]), provided $-\Lambda$ is a Z -matrix (square matrices with nonpositive off-diagonal entries). Another equivalent condition is that Λ is Hurwitz (condition J_{29} in [31]).

Remark 3: Vector Lyapunov functions and the comparison lemma have been well-known since the 1960s [3], [4], [26]. Their application in analyzing large-scale systems was popularized in the 1970s and 80s. For interested readers, see a survey in [27] and the books [34], [37]. ROA estimates have received less attention, but prior work exists [5], [38]. Our contribution is to formulate this in a compositional form that is readily verifiable by numerical SMT solvers, such as dReal [10], even with numerical errors around the origin. Specifically, we examine the linearization around the origin and use the robustness implied by linear system stability to bound high-order terms around the origin. One advantage of using a vector Lyapunov function, as opposed to a weighted sum of scalar Lyapunov functions, for stability and

reachability analysis is that it offers invariance certification for hyperrectangular sets (in \mathbb{R}^l) rather than ellipsoids or rhombi, a benefit noted in [38].

B. Reachability analysis

The local analysis in the previous section is inherently conservative because of the use of the matrix norm and a quadratic Lyapunov function for an interconnected nonlinear system. The next result provides verifiable conditions to expand the region of attraction through reachability analysis using vector Lyapunov functions.

Proposition 2: Suppose that there exist Lyapunov functions $V_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R}$, positive vectors $c = (c_1, \dots, c_l) \in \mathbb{R}^l$ and $v = (v_1, \dots, v_l) \in \mathbb{R}^l$, and a constant $\varepsilon < 0$ such that $c \leq v$ and the following inequalities hold:

$$\nabla V_i(x_i) \left[f_i(x_i) + \sum_{j \neq i} G_{ij}(x_i, x_j) \right] \leq \varepsilon < 0 \quad (11)$$

for all $x_i \in \{x_i \in \mathbb{R}^{n_i} : c_i \leq V_i(x_i) \leq v_i\}$ and all $x_j \in \mathcal{V}_j(v_j) := \{x_j \in \mathbb{R}^{n_j} : V_j(x_j) \leq v_j\}$. Then solutions of the interconnected system (1) starting in $\mathcal{V}(v) = \mathcal{V}_1(v_1) \times \dots \times \mathcal{V}_l(v_l)$ reach $\mathcal{V}(c)$ in finite time and remain there afterwards.

The proof can be found in the appendix of [23].

IV. TRAINING VECTOR NEURAL LYAPUNOV FUNCTIONS AND SMT VERIFICATION

In this section, we describe the proposed computational approach for training and verifying neural vector Lyapunov functions for interconnected systems. Both the learning and verification are conducted in a compositional fashion so that it can leverage the compositional structure of the interconnected system to scale the approach to tackle problems beyond the reach of monolithic approaches.

A. Training neural Lyapunov functions for subsystems

We following the approach presented in [21], [22], [24] for training neural Lyapunov functions for individual subsystems using Zubov's equation and physics-informed neural networks [20], [32]. The main idea is to solve Zubov's PDE

$$\dot{W}(x) := \nabla W(x) \cdot f(x) = -\Psi(x)(1 - W(x)), \quad (12)$$

where W is a positive definite function to be solved and Ψ is also a positive definite function. Two particular choices [21], [22] of Ψ are $\Psi_1(x) = \alpha \|x\|^2$ or $\Psi_2(x) = \alpha(1 + W(x)) \|x\|^2$ for some positive constant $\alpha > 0$. For numerical examples in this paper, we choose $\Psi(x) = \alpha(1 + W(x)) \|x\|^2$ with $\alpha = 0.1$, which corresponds to taking a transform $\beta(s) = \tanh(\alpha s)$ of a usual Lyapunov function [17].

Let $W_{\text{NN}}(x; \theta)$ be a neural network solution to Zubov's PDE (12) on a domain $\Omega \subseteq \mathbb{R}^N$. Consider the training loss

$$\mathcal{L}(\theta) = L_r(\theta) + L_b(\theta) + L_d(\theta), \quad (13)$$

where L_r is the residual error of the PDE, evaluated over a set of collocation points $\{x_i\}_{i=1}^{N_r} \subseteq \Omega$ as

$$L_r = \frac{1}{N_c} \sum_{i=1}^{N_c} (\nabla_x W_{\text{NN}}(x_i; \theta) f(x_i) + \Psi(x_i)(1 - W_{\text{NN}}(x_i; \theta)))^2. \quad (14)$$

The loss L_b captures boundary conditions. As in [21], [22], we can encourage the following inequality near the origin:

$$\beta(c_1 \|x\|^2) \leq W(x) \leq \beta(c_2 \|x\|^2), \quad (15)$$

where $\beta(s) = \tanh(\alpha s)$ as discussed above.

Finally, the term $L_d(\theta)$ can capture a data loss of the form

$$L_d(\theta) = \frac{1}{N_d} \sum_{i=1}^{N_d} (W_{\text{NN}}(y_i; \theta) - \hat{W}(y_i))^2, \quad (16)$$

where $\{\hat{W}(y_i)\}_{i=1}^{N_d}$ is a set of data points, which can be obtained by forward integration of the ODE defining (2) [17].

B. Verification of stability and reachability using vector neural Lyapunov functions and SMT solvers

We aim to use satisfiability modulo theories (SMT) solvers to verify the inequalities (7), (8), and (11) in terms of quadratic Lyapunov functions and the trained neural Lyapunov functions to certify local stability and regions of attraction. The procedure proceeds as follows.

(1) Local stability: We use quadratic Lyapunov functions, obtained by solving Lyapunov equations (5), to verify inequalities (7) and (8). The matrix Λ is computed according to (9) and (10). The choice of initial c is arbitrary, as long as it is sufficiently small such that the matrix $-\Lambda$ becomes a nonsingular M -matrix [31]. This is always achievable because as r_{ij} becomes sufficiently small, the matrix Λ will be diagonally dominating. Once the M -matrix condition is met, there always exists some $p > 0$ such that $\Lambda p < 0$ [31].

For simplicity, we can use the largest vector c obtained by the conditions in [21], [22] for local stability of individual subsystems and scale it down until inequalities (7) and (8) are verified such that Λ defined by (9) and (10) satisfies $\Lambda c < 0$. As a result, we have verified a local stability region

$$\mathcal{P}(p) := \mathcal{P}_1(p_1) \times \dots \times \mathcal{P}_l(p_l),$$

where $\mathcal{P}_i(p_i) := \{x_i \in \mathbb{R}^{n_i} : x_i^T P_i x_i \leq p_i\}$, as defined in Proposition 1.

(2) Reachability by quadratic Lyapunov functions: The local stability region can be very small due to the use of matrix norms for estimates. We can employ a successive procedure to enlarge the region of attraction by Proposition 2 as follows. Suppose there exists a sequence of vectors c_P^m ($m = 1, 2, \dots, k$) such that Proposition 2 holds with $V_i(x_i) = x_i^T P_i x_i$, $c = c_P^m$, and $v = c_P^{m+1}$ for each $m = 1, \dots, k - 1$. It follows that if $c_P^1 < p$, where p is from **Step 1**, then the set $\mathcal{P}(c_P^k)$ is an enlarged region of attraction for the interconnected system (1).

(3) Reachability by general vector Lyapunov functions: Following **Step 2**, we employ SMT solvers to verify that the neural vector neural Lyapunov $V(x) = (V_1(x_1), \dots, V_l(x_l))$ trained according to Section IV-A can satisfy Proposition 2 with $c = c_V^m$, and $v = c_V^{m+1}$, along with a sequence of vectors c_V^m ($m = 1, 2, \dots, q$) such that the following set containment condition is met

$$\mathcal{V}(c_V^1) := \{x \in \mathbb{R}^N : V(x) \leq c_V^1\} \subseteq \mathcal{P}(c_P^k),$$

where c_P^k is from **Step 2**. Following this, we potentially obtain a further enlarged region of attraction $\mathcal{V}(c_V^q)$.

C. Compositional nature of SMT verification

The training process is naturally compositional because the neural Lyapunov functions are trained for individual systems separately. In this subsection, we discuss how verification of the aforementioned Lyapunov conditions in terms of inequalities can also be verified in a compositional fashion.

Local stability is done through verifying (7) and (8), which is compositional because (7) only involves one subsystem and (8) only involves two. By examining the nature of G_{ij} in concrete examples, we may be able to further decompose the verification of (8).

The search for the sequences $\{c_P^m\}$ and $\{c_V^m\}$ in **Steps 2 and 3** of Section IV-B can be formulated as successively solving a minimization problem of the form

$$\begin{aligned} & \underset{c_i}{\text{minimize}} && c_i \\ & \text{subject to} && (11), \end{aligned} \quad (17)$$

where v is given. We can solve this problem for all $i = 1, \dots, l$ and then set $v = c$ and continue until the value of c cannot be improved further by predefined threshold.

The constraint given by the inequality (11) can be further decomposed. We can verify the existence of constants g_{ij}

$$\nabla V_i(x_i)G_{ij}(x_i, x_j) \leq g_{ij} \quad (18)$$

for all $x_i \in \mathcal{V}_i(v_i)$ and $x_j \in \mathcal{V}_j(v_j)$ and replace (11) with

$$\nabla V_i(x_i)f_i(x_i) + \sum_{j \neq i} g_{ij} \leq \varepsilon < 0. \quad (19)$$

The compositional nature of (18) and (19) is evident. Both of them can be verified readily by numerical SMT solvers. We use dReal [10] for verification in the numerical examples in this paper due to its δ -completeness guarantees. Here, δ is a user-defined precision parameter. Verification of a given formula is guaranteed to succeed if the “ δ -weakening” of the negation of the formula is unsatisfiable.

V. NUMERICAL RESULTS

A. Networked Van der Pol oscillators

Inspired by [19], we consider a network of reversed Van der Pol equations of the form

$$\begin{aligned} \dot{x}_{i1} &= -x_{i2}, \\ \dot{x}_{i2} &= x_{i1} - \mu_i(1 - x_{i1}^2)x_{i2} + \sum_{j \neq i} \mu_{ij}x_{i1}x_{j2}, \end{aligned}$$

where $\mu_i \in (0.5, 2.5)$ and μ_{ij} represents the interconnection strength. The parameters μ_i are randomly generated and take the following values for $i = 1, \dots, 10$:

$$[1.25, 2.4, 1.96, 1.7, 0.81, 0.81, 0.62, 2.23, 1.7, 1.92].$$

We set that $\mu_{ij} \in (-0.1, 0.1)$ and the number of nonzero entries in $\{\mu_{ij}\}$ for each i is fewer than 3 or 4 (referred to as density below). We choose the number of total subsystems $l = 10$. Two network topologies are depicted in Fig. 1

for density equal to 3 and 4. The total dimension of the interconnected system is therefore 20, which is beyond the capability of current SMT or SDP-based synthesis of Lyapunov functions if a monolithic approach is taken. Using the approach proposed in this paper, we are able to train neural vector Lyapunov functions and verify regions of attraction. Comparisons with sums-of-squares (SOS) synthesis demonstrate that the neural Lyapunov functions, computed by solving Zubov’s equation, outperform the “expanding interior” approach taken by SOS design.

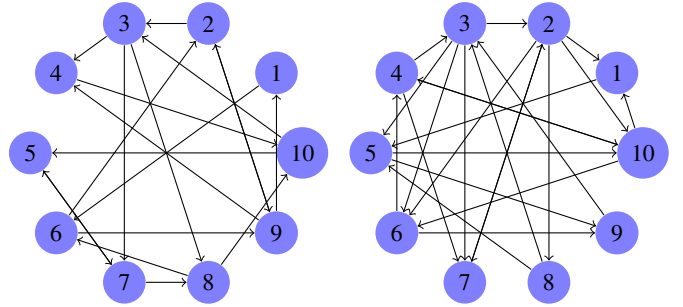


Fig. 1: Two networks with different densities of interconnections by varying the number of nonzero entries in the set μ_{ij} for the Van der Pol network.

B. Hyperparameters

The domain is $[-2.5, 2.5] \times [-5.5, 5.5]$. The number of collocation points $N = 300,000$ with a batch size of 32 for training. We also generate $N_d = 3,000$ data points and use the same batch size to evaluate the data loss (16). We train for a maximum of 20 epochs with a learning rate of 0.001.

C. Results and discussions

We trained feedforward neural networks with $\tanh(\cdot)$ activation as Lyapunov functions for the individual subsystems using the approach proposed in [21], [22] and outlined in Section IV-A. We then compositionally verified regions of attraction using the approach detailed in Section IV-B.

For density equal 3 (network depicted on the left of Fig. 1), the regions of attraction for individual subsystems, certified using SMT solvers for neural Lyapunov functions and SOS Lyapunov functions using semidefinite programming, are shown in 2. It can be observed that, for individual systems, the verified level sets defining regions of attraction are comparable between neural and SOS Lyapunov functions. However, the level sets for the interconnected system certified by neural Lyapunov functions are significantly better than those of the SOS Lyapunov functions. In fact, the ROA for the interconnected system certified by the SOS Lyapunov functions is even worse than that obtained using quadratic Lyapunov functions. One explanation for this is that the “interior expanding” approach [28] expands the region of attraction in a somewhat ad hoc manner, whereas the Zubov approach solves a PDE that characterizes the region of attraction. The PDE used for training is consistent among all subsystems. This might be beneficial as they provide gains

Density	SOS (scale)	Layer	Width	Neural (scale)	Neural (sub-level sets)	Verification Time (s)
3	0.15	2	20	0.27	[0.21, 0.18, 0.19, 0.19, 0.24, 0.23, 0.24, 0.16, 0.19, 0.18]	18,304
3	—	2	30	0.33	[0.26, 0.22, 0.23, 0.24, 0.30, 0.29, 0.30, 0.20, 0.24, 0.22]	38,521
3	—	3	10	0.29	[0.24, 0.19, 0.19, 0.22, 0.26, 0.26, 0.27, 0.19, 0.21, 0.21]	46,611
4	0.09	2	20	0	0	2,952
4	—	2	30	0.23	[0.18, 0.15, 0.16, 0.16, 0.21, 0.20, 0.21, 0.14, 0.17, 0.15]	35,831
4	—	3	10	0.21	[0.18, 0.14, 0.14, 0.16, 0.19, 0.19, 0.20, 0.14, 0.15, 0.15]	35,736

TABLE I: Verification results for various neural network architectures: “scale” denotes the largest verifiable scalar factor used to scale down the initially verified sub-level sets for individual systems. A scale equal to 0 indicates a failure to verify a stability region. Verification time includes bisection and all iterations involved in Step 2 of Section IV-B.

(c.f. equation (18)) that are more compatible among subsystems. However, this is just an intuitive explanation, and more research is likely needed before drawing firm conclusions. Moreover, for simplicity, we only considered scaled versions of the originally verified level sets for quadratic, neural, and SOS Lyapunov functions, respectively. Initializing Step 2 in Section IV-B with different initial level sets, as done in [19], might yield different results. Further comparisons will be conducted in future work.

We also conducted numerical experiments for the network depicted on the right of Fig. 1. We summarize all the verification results in Table I.

The computation time is reported as follows. To train a neural network on a 2020 MacBook Pro with a 2 GHz Quad-Core Intel Core i5 without any GPUs, it takes approximately 500 seconds to complete the training with the hyperparameters set as above. For verification, it takes about 60 seconds to verify a two-hidden-layer neural network with 20 neurons in each layer, roughly 200 seconds to verify a two-hidden-layer neural network with 30 neurons in each layer, and 2000 seconds to verify a three-hidden-layer neural network with 10 neurons each. We note that verification can be expedited if multiple cores are utilized, as verification in dReal leverages interval analysis, which is easily parallelizable. When verifying vector Lyapunov functions for the interconnected system, completing Step 2 in Section IV-B might require multiple iterations to reach the desired level sets. Implementing a bisection procedure to ascertain the optimal scaling down factor from the initially verified level sets can further extend computational time. The cumulative verification time for these processes is presented in Table I. Verification time was captured on an Intel(R) Xeon(R) CPU E5-2683 v4 @ 2.10GHz with 32 cores, a single CPU node through the Digital Research Alliance of Canada. The code for reproducing the example can be found at <https://github.com/j49liu/acc24-compositional-neural-lyapunov>. Future development for this line of research will be supported by the tool LyZNet, which targets physics-informed learning of Lyapunov functions with formal guarantees provided by SMT verification [24].

VI. CONCLUSIONS

This paper introduces a compositional method to train and verify neural Lyapunov functions. Leveraging the compositional structure of interconnected nonlinear systems, we verified neural Lyapunov functions for high-dimensional systems, outperforming current SMT solvers with a monolithic method. Numerical results demonstrate that when trained using Zubov’s PDE and verified compositionally with SMT solvers, neural Lyapunov functions excel over SOS Lyapunov functions obtained from semidefinite programming. Future directions will explore compositional strategies for neural Lyapunov control design.

REFERENCES

- [1] Alessandro Abate, Daniele Ahmed, Mirco Giacobbe, and Andrea Peruffo. Formal synthesis of Lyapunov neural networks. *IEEE Control Systems Letters*, 5(3):773–778, 2020.
- [2] Daniele Ahmed, Andrea Peruffo, and Alessandro Abate. Automated and sound synthesis of Lyapunov functions with smt solvers. In *Proc. of TACAS*, pages 97–114. Springer, 2020.
- [3] Fo N Bailey. The application of Lyapunov’s second method to interconnected systems. *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, 3(3):443–462, 1965.
- [4] Richard Bellman. Vector Lyapunov functions. *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, 1(1):32–34, 1962.
- [5] G Bitsoris and Ch Burgat. Stability analysis of complex discrete systems with locally and globally stable subsystems. *International Journal of Control*, 25(3):413–424, 1977.
- [6] Ya-Chien Chang, Nima Roohi, and Sicun Gao. Neural Lyapunov control. *Advances in Neural Information Processing Systems*, 32, 2019.
- [7] Hongkai Dai, Benoit Landry, Lujie Yang, Marco Pavone, and Russ Tedrake. Lyapunov-stable neural-network control. In *Proc. of Robotics: Science and Systems*, 2021.
- [8] Charles Dawson, Sicun Gao, and Chuchu Fan. Safe control with learned certificates: A survey of neural Lyapunov, barrier, and contraction methods for robotics and control. *IEEE Transactions on Robotics*, 2023.
- [9] Nathan Gaby, Fumin Zhang, and Xiaojing Ye. Lyapunov-net: A deep neural network architecture for Lyapunov function approximation. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 2091–2096. IEEE, 2022.
- [10] Sicun Gao, Soonho Kong, and Edmund M Clarke. dreal: an SMT solver for nonlinear theories over the reals. In *Proc. of CADE*, pages 208–214, 2013.
- [11] Peter Giesl. *Construction of Global Lyapunov Functions Using Radial Basis Functions*, volume 1904. Springer, 2007.
- [12] Peter Giesl and Sigurdur Hafstein. Review on computational methods for Lyapunov functions. *Discrete & Continuous Dynamical Systems-B*, 20(8):2291, 2015.
- [13] Lars Grüne. Computing Lyapunov functions using deep neural networks. *Journal of Computational Dynamics*, 8(2), 2021.

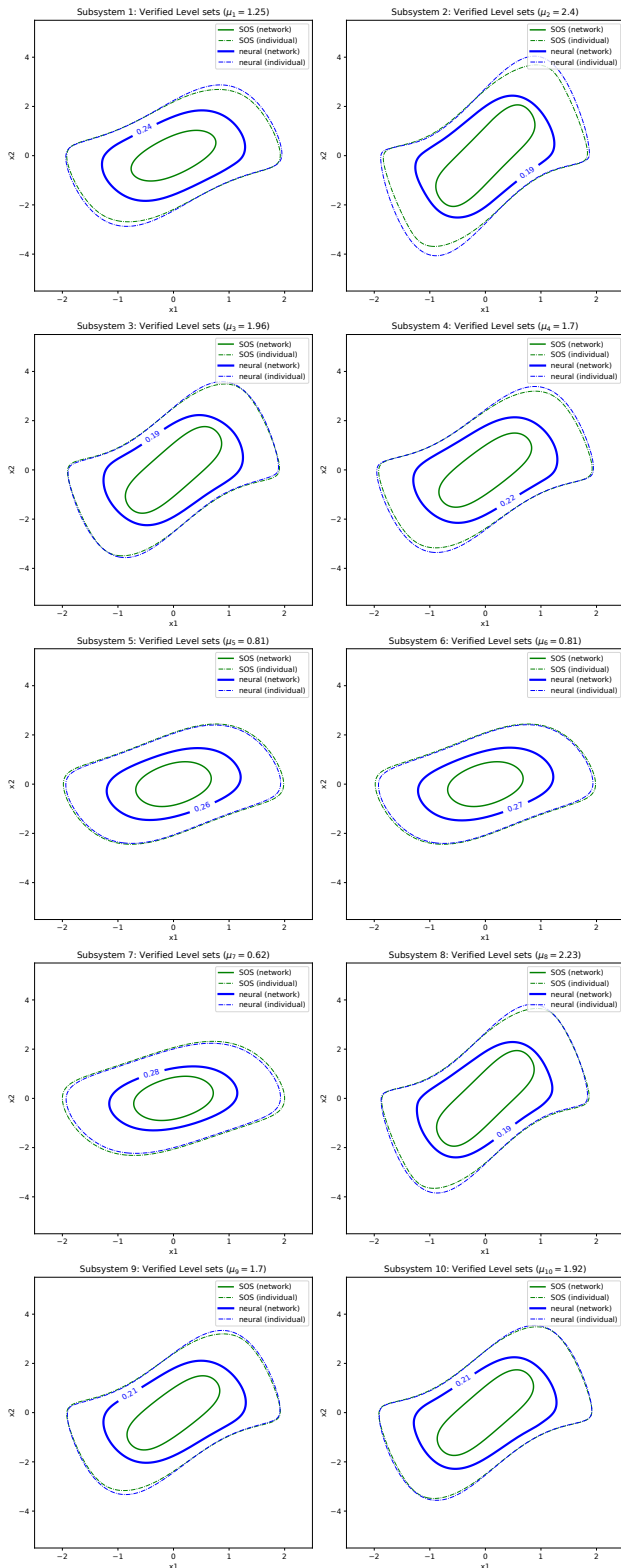


Fig. 2: Neural stability analysis on a 20-dimensional interconnected system: sub-level sets of neural Lyapunov functions [22], which define regions of attraction, are verified by the SMT solver dReal [10] using the approach described in Section IV-B. The thick solid lines represent the regions of attraction for the interconnected system, while the thin dashed lines indicate those for individual subsystems.

[14] Lars Grüne. Overcoming the curse of dimensionality for approximating Lyapunov functions with deep neural networks under a small-gain condition. *IFAC-PapersOnLine*, 54(9):317–322, 2021.

[15] Wassim M Haddad and VijaySekhar Chellaboina. *Nonlinear Dynamical Systems and Control: A Lyapunov-based Approach*. Princeton university press, 2008.

[16] Morgan Jones and Matthew M Peet. Converse Lyapunov functions and converging inner approximations to maximal regions of attraction of nonlinear systems. In *Proc. of CDC*, pages 5312–5319. IEEE, 2021.

[17] Wei Kang, Kai Sun, and Liang Xu. Data-driven computational methods for the domain of attraction and Zubov’s equation. *arXiv preprint arXiv:2112.14415*, 2021.

[18] Larissa Khodadadi, Behzad Samadi, and Hamid Khaloozadeh. Estimation of region of attraction for polynomial nonlinear systems: A numerical method. *ISA Transactions*, 53(1):25–32, 2014.

[19] Soumya Kundu and Marian Anghel. A sum-of-squares approach to the stability and control of interconnected systems using vector Lyapunov functions. In *Proc. of ACC*, pages 5022–5028. IEEE, 2015.

[20] Isaac E Lagaris, Aristidis Likas, and Dimitrios I Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000, 1998.

[21] Jun Liu, Yiming Meng, Maxwell Fitzsimmons, and Ruikun Zhou. Physics-informed neural network Lyapunov functions: PDE characterization, learning, and verification. *arXiv:2312.09131*, 2023.

[22] Jun Liu, Yiming Meng, Maxwell Fitzsimmons, and Ruikun Zhou. Towards learning and verifying maximal neural Lyapunov functions. In *Proc. of CDC*, 2023.

[23] Jun Liu, Yiming Meng, Maxwell Fitzsimmons, and Ruikun Zhou. Compositionally verifiable vector neural Lyapunov functions for stability analysis of interconnected nonlinear systems. *arXiv*, 2024.

[24] Jun Liu, Yiming Meng, Maxwell Fitzsimmons, and Ruikun Zhou. LyZNet: A lightweight python tool for learning and verifying neural Lyapunov functions and regions of attraction. In *Proc. of HSCC*, 2024.

[25] Aleksandr Mikhailovich Lyapunov. The general problem of the stability of motion. *International Journal of Control*, 55(3):531–534, 1992.

[26] VM Matrosov. On the theory of stability of motion. *Journal of Applied Mathematics and Mechanics*, 26(6):1506–1522, 1962.

[27] A Michel. On the status of stability of interconnected systems. *IEEE Transactions on Circuits and Systems*, 30(6):326–340, 1983.

[28] Andy Packard, Ufuk Topcu, Peter J Seiler Jr, and Gary Balas. Help on SOS. *IEEE Control Systems Magazine*, 30(4):18–23, 2010.

[29] Antonis Papachristodoulou and Stephen Prajna. On the construction of Lyapunov functions using the sum of squares decomposition. In *Proc. of CDC*, volume 3, pages 3482–3487. IEEE, 2002.

[30] Antonis Papachristodoulou and Stephen Prajna. A tutorial on sum of squares techniques for systems analysis. In *Proc. of ACC*, pages 2686–2700. IEEE, 2005.

[31] Robert J Plemmons. m -matrix characterizations. I—nonsingular m -matrices. *Linear Algebra and Its Applications*, 18(2):175–188, 1977.

[32] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[33] Rodolphe Sepulchre, Mrdjan Jankovic, and Petar V Kokotovic. *Constructive Nonlinear Control*. Springer, 2012.

[34] Dragoslav D Siljak. *Large-Scale Dynamic Systems: Stability and Structure*. Dover, 1978.

[35] Weehong Tan and Andrew Packard. Stability region analysis using polynomial and composite polynomial Lyapunov functions and sum-of-squares programming. *IEEE Transactions on Automatic Control*, 53(2):565–571, 2008.

[36] Ufuk Topcu, Andrew Packard, and Peter Seiler. Local stability analysis using simulations and sum-of-squares programming. *Automatica*, 44(10):2669–2675, 2008.

[37] Mathukumalli Vidyasagar. *Input-Output Analysis of Large-Scale Interconnected Systems: Decomposition, Well-Posedness and Stability*. Springer, 1981.

[38] Stein Weissenberger. Stability regions of large-scale systems. *Automatica*, 9(6):653–663, 1973.

[39] Ruikun Zhou, Thanin Quartz, Hans De Sterck, and Jun Liu. Neural Lyapunov control of unknown nonlinear systems with stability guarantees. In *Advances in Neural Information Processing Systems*, 2022.

[40] V. I. Zubov. *Methods of A. M. Lyapunov and Their Application*. Noordhoff, 1964.