ZUBOV-NET: ANCHORED NEURAL NETWORKS FOR LEARNING MAXIMAL LYAPUNOV FUNCTIONS WITH FORMAL GUARANTEES

Anonymous authors
Paper under double-blind review

ABSTRACT

Lyapunov functions are a central tool for certifying the stability of physical and engineered systems, yet constructing them remains a long-standing computational challenge. Neural networks can learn Lyapunov functions from data, but the learned functions rarely come with formal guarantees. We introduce *Zubov-Net*, a neural architecture whose outputs are guaranteed to be Lyapunov functions near the equilibrium and whose sublevel sets can cover any compact subset of the region of attraction (ROA). Zubov-Net multiplies a known local Lyapunov function, the anchor, by a learned positive correction. Because this design enforces the local Lyapunov conditions for every choice of network parameters, training only has to learn the global shape of the basin from trajectory data and a well-scaled residual of Zubov's equation. We prove a universal approximation theorem for anchored networks: they can approximate a maximal Lyapunov function arbitrarily well on each of its sublevel sets. Across a range of benchmarks with two to twenty state variables, the formally verified ROA estimates of Zubov-Net are consistently, and often substantially, larger than those of the baseline certificates.

1 INTRODUCTION

Lyapunov functions are the cornerstone of stability analysis for nonlinear dynamical systems (Khalil, 2002). Since Lyapunov (1992) introduced them over a century ago, they have remained a central tool in systems and control, and the same idea now underpins safety-critical robotics and learning-based control of physical systems (Ames et al., 2017; 2014; Berkenkamp et al., 2017; Dawson et al., 2023). A Lyapunov function does more than certify local stability: its sublevel sets delineate the region of attraction (ROA), the set of initial states from which trajectories converge to the equilibrium. More precise knowledge of the ROA matters in practice, because it describes how far a system may deviate from its desired operating point and still recover. For example, in power systems the ROA collects the post-fault states from which generators resynchronize and the grid returns to stable operation (Chiang et al., 1988); in robotics, the configurations and velocities from which feedback recovers a desired posture or motion (Tedrake et al., 2010); in aircraft and spacecraft control, the position, attitude, and velocity errors from which the vehicle returns to its target trajectory (Chakraborty et al., 2011); in chemical and industrial processes, the temperatures and concentrations from which a unit returns to its operating condition (El-Farra & Christofides, 2001); and in biological models such as tumour growth, the cell populations from which treatment can restore tumour dormancy (Merola et al., 2008). Computing Lyapunov functions, however, remains a fundamental challenge (Giesl & Hafstein, 2015), and computing ones whose sublevel sets fill a large part of the ROA is harder still. Sum-of-squares (SOS) programming made the search systematic for polynomial certificates (Parrilo, 2000; Topcu et al., 2008) and remains the standard benchmark for ROAs of polynomial systems. It, however, applies only to polynomial dynamics or to polynomial approximations of non-polynomial ones, and it offers no principled way to approximate the entire ROA. This second limitation is visible in standard benchmarks (Chesi et al., 2005; Khodadadi et al., 2014; Tan & Packard, 2008), which we revisit in Section 6.

Motivated by these challenges, learning-based Lyapunov certificates have received considerable attention. Neural networks apply to non-polynomial dynamics directly, and their universal approximation property lets them represent the curved geometry of the ROA. Training is flexible as well: it can

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

fit a candidate to simulated trajectories (Richards et al., 2018), to the Lyapunov inequalities under counterexample-guided search (Chang et al., 2019; Grüne, 2021; Abate et al., 2021; Dai et al., 2021; Zhou et al., 2022), or to Zubov's equation, whose solution encodes the entire ROA (Liu et al., 2023; 2025; Li et al., 2025). Neither universal approximation nor training, however, establishes that the learned function is a Lyapunov function. Formal verification can close this gap in principle, whether through satisfiability modulo theories (Gao et al., 2013) or through bound propagation with branch and bound (Zhang et al., 2018; Wang et al., 2021), and several pipelines already combine the two (Chang et al., 2019; Abate et al., 2021; Zhou et al., 2022; Liu et al., 2024b). Two difficulties remain. At the equilibrium, both the candidate and its Lie derivative vanish, so any approximation error can destroy positive definiteness of the one or negative definiteness of the other. Near the boundary of the ROA or of the domain, verification becomes harder, and conditions that ignore the direction of the flow on the boundary return needlessly conservative certified sets.

We address these limitations with an architecture that starts from a known local Lyapunov function and learns a correction toward a global target: a *maximal* Lyapunov function, whose finite sublevel sets are compact subsets of the ROA and together fill it (Zubov, 1964; Vannelli & Vidyasagar, 1985). We call the resulting network *Zubov-Net*, because this target solves Zubov's equation, whose solution characterizes the ROA (Zubov, 1964). More specifically, given a positive definite local Lyapunov function V, which we call the *anchor* (Assumption 1 states what an anchor must satisfy, and both quadratic and homogeneous anchors qualify), Zubov-Net represents the unbounded target and its bounded transform as

$$G_\theta(x) = N_\theta(x) - N_\theta(0), \qquad H_\theta(x) = V(x)\exp(G_\theta(x)), \qquad W_\theta(x) = \tanh H_\theta(x), \tag{1}$$

where N_θ is an unconstrained smooth feedforward network. Two features of this construction address the two difficulties above. First, H_θ and W_θ inherit from V the property of being strict Lyapunov functions near the equilibrium, for every parameter vector (Proposition 1), while the multiplicative correction e^{G_θ} remains free to reshape distant sublevel sets, which is what brings a maximal Lyapunov function within reach (Theorem 1). Second, the $\tanh$ transform matches the bounded form of Zubov's equation used in training and simplifies verification, because the decrease condition is equivalent to the sign of a factor that contains neither the saturating output nor the exponential (Section 5). Empirically, across benchmarks from low to high dimension, the ROA estimates that Zubov-Net certifies are consistently, and often substantially, larger than those from the baselines, mostly formally verified SOS certificates.

In summary, our contributions are fourfold.

1. We propose anchored neural networks for learning maximal Lyapunov functions. The Lyapunov conditions hold in a neighborhood of the origin by construction, and the network learns the maximal Lyapunov function away from it.
2. We prove that anchored networks can approximate a maximal Lyapunov function and its derivatives uniformly on every compact subset of the constrained ROA, and that the union of their certified sublevel sets covers the entire constrained ROA.
3. We introduce a viability-aware verification condition that can use any part of the domain boundary along which the flow already points strictly inward. The condition also exploits the architecture: stating it in terms of the latent function H_θ removes the final $\tanh$ and the exponential from the expression sent to the verifier, which reduces the cost of verification without sacrificing approximation power.
4. We design unified recipes for training and verifying a suite of 22 comparison systems from two to twenty dimensions, and show that the architecture yields markedly less conservative certified ROAs than state-of-the-art SOS and neural Lyapunov functions. A separate formally verified cubic-leading example exercises the homogeneous-anchor branch.

To the best of our knowledge, this is the first systematic study of neural architectures for learning maximal Lyapunov functions that combines an approximation guarantee with extensive benchmark evidence: on every model we compare formally, the certified ROA of Zubov-Net is larger than the one certified by SOS. The remaining sections are organized as follows. Section 2 reviews related work. Section 3 defines the architecture and discusses the training procedure. Section 4 proves the universal approximation results. Section 5 presents verifiable sufficient conditions for a learned Lyapunov function to certify a ROA. Section 6 reports the experiments.

2 BACKGROUND AND RELATED WORK

Maximal Lyapunov functions and Zubov's equation. A local Lyapunov function proves attraction only on the sublevel sets where the Lyapunov inequalities hold. A *maximal* Lyapunov function is a proper Lyapunov function that is valid on the entire ROA: each of its finite sublevel sets is a compact subset of the ROA, and the union of these sublevel sets is the ROA itself (Zubov, 1964; Vannelli & Vidyasagar, 1985; Kellett, 2015). Under mild technical conditions, such a function can be constructed by integrating a stage cost along the trajectory that starts at the given state. If the stage cost is positive definite, the resulting function is a Lyapunov function, and it approaches infinity as the initial state approaches the boundary of the ROA. Composing it with the hyperbolic tangent, or with the Kruzhkov transform $W = 1 - e^{-U}$, removes this singularity and yields a bounded function that approaches one at the boundary and satisfies a first-order partial differential equation, known as *Zubov's equation*, whose solution characterizes the ROA (Zubov, 1964). Classical numerical approaches formulate this equation in the viscosity sense and solve it by grid discretization (Camilli et al., 2001a;b). Recent work instead trains neural networks to approximate its solution, either from trajectory data (Kang et al., 2024) or with physics-informed losses (Raissi et al., 2019; Liu et al., 2023; 2025) for computing Lyapunov functions.

Polynomial certificates. Sum-of-squares (SOS) relaxations have been studied extensively for approximating the ROA of polynomial systems (Parrilo, 2000; Papachristodoulou & Prajna, 2002; Tan & Packard, 2008; Topcu et al., 2008; Khodadadi et al., 2014). The resulting convex programs are attractive, but they apply only to polynomial dynamics, or to polynomial approximations of non-polynomial ones; in the latter case, a certificate for the approximation does not by itself certify the original system. Furthermore, the program size grows rapidly with the polynomial degree, and polynomial sublevel sets generally struggle to capture the exact geometry of the ROA. We use these approaches mainly as benchmarks.

Learning Lyapunov functions. Neural networks are flexible function approximators, and their universal approximation properties make them natural candidates for Lyapunov functions. They have been trained to approximate Lyapunov functions from trajectories (Richards et al., 2018) and to solve the Lyapunov partial differential inequalities that characterize them (Grüne, 2021). Other work alternates training with counterexample generation and formal verification (Chang et al., 2019; Abate et al., 2021; Dai et al., 2021; Zhou et al., 2022). As noted above, Zubov's equation has also served as an explicit training criterion (Kang et al., 2024; Liu et al., 2023; 2025; Li et al., 2025). We emphasize in particular the work that builds the Lyapunov conditions into the architecture: Gaby et al. (2022) augment the output with a positive term to ensure positive definiteness, Fitzsimmons & Liu (2024) give a network structure with verifiable sufficient conditions for homogeneous vector fields, and Barreau & Bastianello (2024) add neural higher-order terms to a local quadratic form. The last work, however, trains only a clipped and regularized variant of its proposed architecture, and verifies no trained network formally. In contrast, we train and verify the architecture itself, prove universal approximation of maximal Lyapunov functions, and compare with SOS certificates systematically across dimensions. Zubov-Net also admits anchors more general than a local quadratic Lyapunov function, such as the homogeneous Lyapunov functions of Fitzsimmons & Liu (2024), or any local Lyapunov function with a certified decay margin near the equilibrium.

Formal verification of neural Lyapunov functions. Formal neural Lyapunov synthesis has used a range of techniques, including satisfiability modulo theories, mixed-integer programming, and bound propagation (Abate et al., 2021; Dai et al., 2021; Liu & Fitzsimmons, 2025; Liu et al., 2024b). Notably, CROWN and α-β-CROWN combine differentiable bounds with complete branch-and-bound search (Zhang et al., 2018; Xu et al., 2021; Wang et al., 2021). We verify Zubov-Net with α-β-CROWN, and state the verification conditions in terms of the latent term H_θ to reduce the cost of each query. Our conditions are also viability-aware: a certified sublevel set may meet constraint boundaries along which the flow points strictly inward, because invariance depends on the vector field at the active constraints rather than on clearance from every face (Blanchini, 1999). In contrast, maximal Lyapunov–barrier functions couple Zubov PDEs to state constraints by treating each constraint as an obstacle to be avoided (Meng & Liu, 2026). Our experiments further extend formal verification of Zubov-Net to networked systems, in the spirit of sparse and compositional Lyapunov inequalities (Zhang et al., 2023; Liu et al., 2024a).

3 ANCHORED NEURAL NETWORKS FOR LEARNING LYAPUNOV FUNCTIONS

This section defines the anchored architecture and describes how the correction network is trained.

3.1 CONSTRAINED REGIONS OF ATTRACTION

Consider $\dot{x} = f(x)$, where $f : \mathbb{R}^n \to \mathbb{R}^n$ is C^1 and $f(0) = 0$, and let ϕ denote its flow. The domain of interest is defined by

$$\mathcal{D} = \{x \in \mathbb{R}^n : h_i(x) \le 1,\ i = 1, \dots, m\}, \tag{2}$$

where each h_i is C^1 and $0 \in \operatorname{int} \mathcal{D}$. For example, affine h_i's can describe box domains as well as half-space constraints. We are interested in the constrained ROA relative to $\mathcal{D}$ defined as

$$\mathcal{A}_{\mathcal{D}} = \{x_0 \in \mathcal{D} : \phi(t, x_0) \in \mathcal{D} \text{ for all } t \ge 0 \text{ and } \phi(t, x_0) \to 0\}. \tag{3}$$

A trajectory that leaves $\mathcal{D}$ and later returns does not belong to $\mathcal{A}_{\mathcal{D}}$. Thus $\mathcal{A}_{\mathcal{D}}$ is the safe ROA when $\mathcal{D}$ specifies the safe set.

3.2 LOCAL LYAPUNOV FUNCTIONS AS ANCHORS

The architecture starts from a known local Lyapunov function V, which we call the *anchor*. Any certified local Lyapunov function can serve as V. We use two concrete constructions.

Quadratic anchor. If $A = Df(0)$ is Hurwitz, i.e., all eigenvalues of A have negative real parts, we can take $V(x) = x^\top P x$, where $P \succ 0$ solves the celebrated Lyapunov equation $A^\top P + PA = -Q$ for a chosen $Q \succ 0$. This is the quadratic branch used by the comparison experiments in Section 6.

Homogeneous anchor. Suppose instead that the local expansion of f begins with

$$f(x) = f_d(x) + O(\|x\|^{d+1}), \qquad f_d(\lambda x) = \lambda^d f_d(x) \quad (\lambda > 0), \tag{4}$$

where $f_d : \mathbb{R}^n \to \mathbb{R}^n$ is the first nonzero homogeneous term and $d \ge 1$ is its degree. If the homogeneous system $\dot{x} = f_d(x)$ is asymptotically stable, Rosier (1992, Theorem 2) gives, for every $p > 1$, a positive definite, proper function $V_p \in C^1(\mathbb{R}^n) \cap C^\infty(\mathbb{R}^n \setminus \{0\})$ and a constant $\alpha > 0$ such that

$$V_p(\lambda x) = \lambda^p V_p(x) \quad (\lambda > 0), \qquad \nabla V_p(x)^\top f_d(x) \le -\alpha \|x\|^{p+d-1} \quad (x \ne 0). \tag{5}$$

We can use $V = V_p$ as the anchor. The higher-order remainder in equation (4) perturbs the Lie derivative only at order $\|x\|^{p+d}$, so V_p remains a strict local Lyapunov function.

Fitzsimmons & Liu (2024) construct homogeneous neural Lyapunov functions by learning an angular factor $R(u)$ on the unit sphere and extending it radially as $\|x\|^p R(x/\|x\|)$. We use the positivity-preserving specialization $R(u) = \exp(s(u))$ with $s(u) = a_0 + w^\top \tanh(Au + b)$:

$$V_p(x) = \|x\|^p \exp\left(s\left(\frac{x}{\|x\|}\right)\right), \qquad x \ne 0, \tag{6}$$

with $V_p(0) = 0$. The exponential makes the candidate positive away from the origin, while the angular argument makes it homogeneous of degree p. Unlike a quadratic anchor that is algebraically determined by solving the Lyapunov equation, the homogeneous network V_p is trained and then formally verified against equation (5) before it can be used as the anchor.

3.3 ANCHORED NEURAL NETWORK ARCHITECTURE

Let $N_\theta : \mathbb{R}^n \to \mathbb{R}$ be a smooth feedforward network and set

$$G_\theta(x) = N_\theta(x) - N_\theta(0), \quad H_\theta(x) = V(x) e^{G_\theta(x)}, \quad W_\theta(x) = \tanh H_\theta(x). \tag{7}$$

By construction, $H_\theta(0) = 0$ and $H_\theta(x) > 0$ wherever $V(x) > 0$. Smoothness and $G_\theta(0) = 0$ give $e^{G_\theta(x)} = 1 + O(\|x\|)$, so $H_\theta(x)/V(x) \to 1$ as $x \to 0$. If $V = V_p$ is homogeneous of degree p, then $H_\theta(x) = V_p(x) + O(\|x\|^{p+1})$: the correction preserves the anchor's leading order while allowing correction away from the origin to learn a maximal Lyapunov function.

The bounded output $W_\theta \in [0, 1)$ matches the range of the transformed Zubov target and is convenient for verification. Its derivative along trajectories factors as

$$\mathcal{L}_f W_\theta = (1 - W_\theta^2)\, e^{G_\theta} \underbrace{(\nabla V + V \nabla G_\theta)^\top f}_{R_\theta}. \tag{8}$$

Because both factors preceding R_θ are strictly positive, the sign of $\mathcal{L}_f W_\theta$ is exactly the sign of R_θ. Formal verification of the Lyapunov decrease condition $\mathcal{L}_f W_\theta < 0$ therefore reduces to checking $R_\theta < 0$ (Section 5), which eliminates the exponential term and the potentially saturated factor $(1 - W_\theta^2)$ near the ROA boundary from formal verification.

3.4 Learning a maximal Lyapunov function

Choose a continuous running cost ω such that $\omega(0) = 0$ and $\omega(x) > 0$ for $x \neq 0$. Under standard assumptions, a maximal Lyapunov function (Vannelli & Vidyasagar, 1985; Kang et al., 2024; Liu et al., 2025) on $\mathcal{A}_\mathcal{D}$ is given by

$$U(x) = \int_0^\infty \omega(\phi(t, x))\, dt, \tag{9}$$

and satisfies the PDE

$$\nabla U(x)^\top f(x) = -\omega(x). \tag{10}$$

The corresponding bounded target $W = \tanh U$ satisfies the Zubov PDE (Liu et al., 2025)

$$\nabla W(x)^\top f(x) = -\omega(x)(1 - W(x)^2). \tag{11}$$

We train H_θ to approximate U, so $W_\theta = \tanh H_\theta$ approximates W. For $x \neq 0$, substituting $H_\theta = V e^{G_\theta}$ into equation (10) and dividing by ω gives the relative residual

$$\rho_\theta(x) = 1 + \frac{e^{G_\theta(x)}(\nabla V(x) + V(x)\nabla G_\theta(x))^\top f(x)}{\omega(x)} = 1 + \frac{e^{G_\theta(x)} R_\theta(x)}{\omega(x)}. \tag{12}$$

Thus $\rho_\theta = 0$ is the target PDE, while $\rho_\theta < 1$ holds exactly when $\mathcal{L}_f H_\theta < 0$. This residual omits the factor $1 - W_\theta^2$, which becomes small near the ROA boundary (Liu et al., 2025).

Two simple choices of running cost follow from the anchors. For a quadratic anchor, we use $\omega(x) = x^\top Q x$. For a homogeneous anchor, one may use $\omega(x) = -\nabla V_p(x)^\top f_d(x) > 0$, where positivity follows from equation (5).

We generate training data from simulated trajectories. For a state x_j on a trajectory that converges inside $\mathcal{D}$, let $\widehat{J}_j$ denote an approximation of $U(x_j)$ by numerical integration. We minimize

$$\mathcal{L}(\theta) = \mathrm{MSE}\left(G_\theta(x_j), \log \frac{\widehat{J}_j}{V(x_j)}\right) + \lambda_p\, \mathrm{MSE}(\rho_\theta(x_j), 0) + \lambda_m\, \mathrm{mean}\,[\rho_\theta(x_j) - \eta]_+^2. \tag{13}$$

The data term fits $H_\theta(x_j)$ to the trajectory estimate $\widehat{J}_j$ through the log-ratio $\log(\widehat{J}_j / V(x_j))$. The PDE term makes ρ_θ close to zero at the training points. The margin term penalizes $\rho_\theta > \eta$; choosing $\eta < 1$ encourages strict decrease with a buffer. The quadratic-anchor experiments introduce these terms in three stages: data, PDE ($\lambda_p = 1$), and margin ($\lambda_m = 100$, $\eta = 0.5$). If a trajectory leaves $\mathcal{D}$, we cap the value of its sampled states at a large constant, which corresponds to the label $W \approx 1$.

Counterexample-guided continuation. After initial training, a sampled screen estimates the largest sublevel on which the Lyapunov conditions hold at the sampled points and identifies states where one of them fails. We simulate trajectories from those states. If a trajectory yields a finite estimate of U, we add its samples to the training data and resume training. This loop continues within a fixed computational budget. It guides training but does not replace formal verification (Section 5).

4 Universal approximation and maximality guarantees

This section establishes some mathematical guarantees of the proposed anchored neural architecture for approximating maximal Lyapunov functions. We state only the main results; their proofs are given in Appendix A.

Assumption 1 (Certified local anchor)**.** *The function $V : \mathbb{R}^n \to \mathbb{R}$ is positive definite, continuously differentiable, and twice continuously differentiable on $\mathbb{R}^n \setminus \{0\}$. There are constants $p > 1$, $d \geq 1$, $r_0 > 0$, and $\alpha, C_V, C_f > 0$ such that, for $0 < \|x\| \leq r_0$,*

$$\mathcal{L}_f V(x) \leq -\alpha \|x\|^{p+d-1}, \qquad V(x) \leq C_V \|x\|^p, \qquad \|f(x)\| \leq C_f \|x\|^d. \tag{14}$$

Both anchor constructions in Section 3 satisfy Assumption 1.

Our first result states that, by construction, the anchored function is a valid local Lyapunov function for every choice of the network parameters.

Proposition 1 (Local Lyapunov property for every parameter)**.** *Let Assumption 1 hold, and let G_θ be C^1 with $G_\theta(0) = 0$. Then $H_\theta = Ve^{G_\theta}$ and $W_\theta = \tanh H_\theta$ are positive definite, and both decrease strictly along every nonzero trajectory in some neighborhood of the origin.*

Our main result needs one hypothesis beyond Assumption 1: $\mathcal{A}_\mathcal{D}$ is *relatively open* in $\mathcal{D}$, that is, every point of $\mathcal{A}_\mathcal{D}$ has a neighborhood whose intersection with $\mathcal{D}$ lies in $\mathcal{A}_\mathcal{D}$. This excludes trajectories in $\mathcal{A}_\mathcal{D}$ that touch $\partial\mathcal{D}$ tangentially, since arbitrarily small perturbations of such a trajectory leave $\mathcal{D}$. It holds whenever $\mathcal{D}$ is positively invariant, and whenever the flow points strictly inward at every point of $\mathcal{A}_\mathcal{D} \cap \partial\mathcal{D}$ (Lemma 2), the same inward-flow condition that the verifier checks in Section 5.

Theorem 1 (Certified sublevel sets exhaust the constrained ROA)**.** *Let Assumption 1 hold, suppose that $\mathcal{A}_\mathcal{D}$ is relatively open in $\mathcal{D}$, and let the activation function be C^1 and nonpolynomial. Then for every compact set $K \subseteq \mathcal{A}_\mathcal{D}$ there exist an anchored network $H_\theta = Ve^{G_\theta}$ and a level $c > 0$ such that the connected component $S_{\theta,c}$ of $\{x \in \mathcal{D} : H_\theta(x) \leq c\}$ that contains the origin satisfies (i) $K \subseteq S_{\theta,c} \subseteq \mathcal{A}_\mathcal{D}$, (ii) $\mathcal{L}_f H_\theta < 0$ on $S_{\theta,c} \setminus \{0\}$, and (iii) $S_{\theta,c}$ is positively invariant. Consequently, there are sets $S_1 \subseteq S_2 \subseteq \cdots$ of this form whose union is $\mathcal{A}_\mathcal{D}$.*

Conditions (ii) and (iii) are what Section 5 verifies, with explicit margins, for a trained candidate; condition (i) says that every compact subset of the constrained ROA can be contained in a verified ROA obtained from a learned candidate. Since $\tanh$ is increasing, the theorem applies verbatim to $W_\theta = \tanh H_\theta$ with the level $\tanh c$.

Proof idea. The proof constructs a maximal Lyapunov function U of the form equation (9) whose running cost equals $-\mathcal{L}_f V$ near the origin. Then $U = V$ there, so the log-ratio $\log(U/V)$, the quantity fitted by the data term in equation (13), is C^1 on every compact sublevel set of U, including across the origin (Lemma 1). Universal approximation of this log-ratio in C^1 (Hornik et al., 1990; Pinkus, 1999) gives anchored networks that approximate U and ∇U on a compact sublevel set $K_b \supseteq K$. Strict decrease of H_θ along trajectories near the origin follows from Proposition 1, where the neighborhood can be made independent of the approximation accuracy. Elsewhere on K_b, strict decrease follows from the uniform decrease of U. To show that the union of sublevel sets can form $\mathcal{A}_\mathcal{D}$, we show that $S_{\theta,c}$ can be squeezed between two sublevel sets of U. This is where relative openness of $\mathcal{A}_\mathcal{D}$ is used. □

We make three remarks on this result. (1) The theorem is about representability; it does not assert that training finds the network. To simplify the proof, we use a matched running cost, whereas actual training may use any positive definite cost, such as those in Section 3. (2) The universality result is semiglobal in the sense that we do not seek a single finite network that captures the boundary of $\mathcal{A}_\mathcal{D}$; instead, a union of neural sublevel sets recovers the whole constrained ROA. (3) The validity of the anchored Lyapunov function is established on a neighborhood of the origin, but the proof does not provide an explicit estimate of its size. We therefore rely on formal verification, discussed in the next section, to certify a constrained ROA rigorously.

5 Boundary-aware formal verification

Theorem 1 shows that good candidates exist and that the proposed anchored neural architecture is capable of representing them. Section 3 discussed the training mechanism tailored to the anchored neural architecture. In this section, we show how to formally verify a trained candidate so that it certifies a constrained ROA, and does so in a non-conservative fashion.

The usual bounded-domain certificate requires a sublevel set to be completely contained in $\mathcal{D}$ (see, e.g., Liu et al., 2025). This is sufficient but can be needlessly conservative: a physical constraint

or a given domain may already form part of the boundary of an invariant set. For $x \in \mathcal{D}$, let $I(x) = \{i : h_i(x) = 1\}$ denote the set of constraints in equation (2) that are active at x. We propose the following proposition, which is the one implemented by our verifier.

Proposition 2 (Two-stage verification with viable boundaries)**.** *Let $C_0 \subset \mathcal{D}$ be a positively invariant neighborhood of the origin on which convergence to the origin is certified. Let W be C^1 near $\mathcal{D}$, and let $0 < c_{\text{in}} < c_{\text{out}}$. Suppose that there are constants $\epsilon > 0$ and $\delta_i > 0$ such that*

$$\{x \in \mathcal{D} : W(x) \le c_{\text{in}}\} \subseteq C_0, \tag{15}$$

$$\mathcal{L}_f W(x) \le -\epsilon, \qquad \text{for all } x \in \mathcal{D} \text{ with } c_{\text{in}} \le W(x) \le c_{\text{out}}, \tag{16}$$

$$W(x) > c_{\text{out}} \quad \text{or} \quad \mathcal{L}_f h_i(x) \le -\delta_i, \qquad \text{for all } i \in I(x) \text{ and for all } x \in \mathcal{D} \text{ with } I(x) \neq \emptyset. \tag{17}$$

Then $S = \mathcal{D} \cap \{W \le c_{\text{out}}\}$ is positively invariant, and every trajectory starting in S enters C_0 in finite time and converges to the origin. In particular, $S \subseteq \mathcal{A}_{\mathcal{D}}$.

We essentially certify two properties of W: (1) *containment*, equation (15) shows that an inner sublevel set of W is contained in a certified local ROA C_0; and (2) *reach*, equation (16) shows that every trajectory in the outer sublevel set reaches the inner one within time $(c_{\text{out}} - c_{\text{in}})/\epsilon$. The boundary condition equation (17) is essentially the strict form of Nagumo's condition (Blanchini, 1999), which ensures that trajectories remain inside $\mathcal{D}$. We use strict inequalities because both SMT solvers (Gao et al., 2013) and neural network verifiers can only effectively verify strict inequalities. This also makes the proof, given in Appendix A, more or less elementary.

Next, we show that the architecture also enables us to further simplify the verification conditions.

Proposition 3 (Exact pre-transform verification)**.** *Let $c \in (0, 1)$ and write $\gamma(c) = \operatorname{atanh}(c)$. For the anchored candidate in equation (7) and every $x \in \mathbb{R}^n$,*

$$W_\theta(x) \le c \iff H_\theta(x) \le \gamma(c), \tag{18}$$

$$\mathcal{L}_f W_\theta(x) < 0 \iff R_\theta(x) = (\nabla V + V \nabla G_\theta)^\top f < 0. \tag{19}$$

Furthermore, if

$$G_\theta(x) > \log\left(\frac{\gamma(c_{\text{in}})}{c_V}\right) \qquad \text{for all } x \in \mathcal{D} \text{ with } V(x) \ge c_V, \tag{20}$$

then $\{x \in \mathcal{D} : W_\theta(x) \le c_{\text{in}}\} \subseteq \{V < c_V\}$. Finally, if $R_\theta \le -\epsilon_R$ on the annulus $Q = \{x \in \mathcal{D} : c_{\text{in}} \le W_\theta(x) \le c_{\text{out}}\}$ for some $\epsilon_R > 0$, then equation (16) holds with

$$\epsilon = \epsilon_R \min_{x \in Q} (1 - W_\theta(x)^2)\, e^{G_\theta(x)} > 0. \tag{21}$$

Proposition 3 effectively removes the final $\tanh$ transformation from the verification conditions, both for the level sets and for the Lie derivative condition; the factor $(1 - W_\theta^2)e^{G_\theta}$ removed from the derivative vanishes near the ROA boundary and is potentially more difficult to bound. Verification of all the benchmark examples in Section 6 uses the simplified conditions.

6 Experiments

The experiments are designed to answer the following questions: (1) Can one fixed training recipe produce formally verified sets larger than those certified by SOS across standard low-dimensional benchmarks? (2) Does the approach extend to structured systems of higher dimension? (3) What does the anchored architecture contribute, and can it incorporate anchors more general than quadratic, or even polynomial, ones?

Selection of benchmarks. The benchmarks include standard examples from the literature (Vannelli & Vidyasagar, 1985; Chesi et al., 2005; Topcu et al., 2008; Khodadadi et al., 2014; Ahmadi et al., 2011). We label planar systems P1–P12, three-dimensional systems T1–T4, and structured higher-dimensional systems S1–S6; Appendix C gives their domains, architectures, and the scope of each formal comparison. We also include a planar system that does not admit a polynomial Lyapunov function, to show that we can incorporate anchors more general than quadratic or polynomial ones.

Table 1: Planar results under one common recipe.

case	system	SOS	Zubov-Net	gain
P1	Van der Pol, $\mu = 1$	13.0422	**13.5051**	1.04×
P2	two-machine swing	1.8415	**9.5674**	5.20×
P3	semialgebraic basin	4.8373	**23.8470**	4.93×
P4	shifted Lotka–Volterra	4.5907	**8.5928**	1.87×
P5	Chesi S1	40.3711	**55.7803**	1.38×
P6	Chesi S4	3.1662	**4.1184**	1.30×
P7	Khodadadi quadratic	92.8184	**183.4942**	1.98×
P8	Stepanova model	0.1277	**0.4229**	3.31×
P9	saddle oscillator	18.3470	**48.5421**	2.65×
P10	Van der Pol, $\mu = 3$	14.6459	**16.9949**	1.16×
P11	Ahmadi triangular	40.1783	**50.9451**	1.27×
P12	Chesi S2 cubic	84.9774	**111.0029**	1.31×

Table 2: Three-dimensional results under one common recipe for the examples E3–E6 of Topcu et al. (2008); the SOS volumes of T2–T4 are directly quoted from the source.

case	SOS	Zubov-Net	gain
T1	125.829	**216.914**	1.72×
T2	4.189	**4.690**	1.12×
T3	260.962	**724.785**	2.78×
T4	146.608	**953.809**	6.51×

Table 3: Plain MLP versus anchored network on the 5D aircraft (S3).

	plain	anchored
violations near origin	27.1%	**0%**
verified volume	8,567	**12,168**

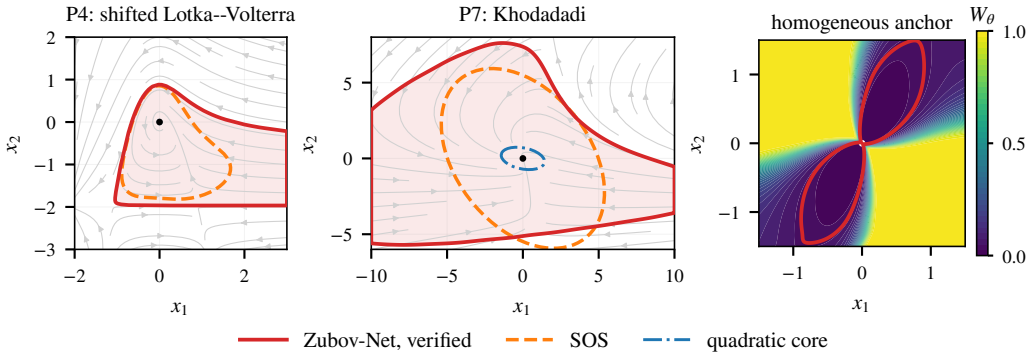

Figure 1: Formally verified certificates. Left and middle: P4 and P7 under the common recipe. Right: a planar system without a polynomial Lyapunov function (Liu & Fitzsimmons, 2026), for which Zubov-Net with a homogeneous anchor certifies a region of attraction.

Reported results. All Zubov-Net results are formally verified sublevel sets; we report their volumes in physical coordinates, estimated with scrambled Sobol samples after the level has been certified. Baselines are published SOS certificates that we formally verify against the model, or published volumes where no certificate is available; if neither exists, we synthesize an SOS certificate with standard tools. The Pendubot example (S2) is synthesized and verified for a cubic Taylor approximation, because verification of the SOS certificate against the exact trigonometric model fails.

Training recipes. All 22 comparison experiments use quadratic anchors and $\tanh$ networks. All planar cases use the following settings: two hidden layers of width 30, 20,000 Adam updates on trajectory data, 20,000 with the PDE loss, and one fixed four-stage L-BFGS schedule; every scheduled checkpoint is screened, the largest eligible one is selected, and one formal verification is run. All three-dimensional cases use one larger adaptive recipe: two hidden layers of width 48, 45,000 initial updates, and at most six repair rounds, each of which takes one pass of the counterexample-guided continuation of Section 3 with 15,000 additional training updates. Cases of dimension four and higher use architectures adapted to dimension and sparsity.

6.1 Common recipes in two and three dimensions

Table 1 contains every planar case. The common recipe on Zubov-Net exceeds every formally verified polynomial baseline. The gain is modest when the polynomial already follows the basin well (P1) and large for asymmetric or nonellipsoidal basins (P2, P3, and P8). P4 illustrates the advantage of

Table 4: Structured higher-dimensional results; Appendix C states the scope of each comparison.

case	n	system	baseline	base vol.	Zubov-Net	gain
S1	4	full swing two-machine	quartic SOS, formal	72.493	**920.920**	$12.70\times$
S2	4	Pendubot	quartic SOS, Taylor model	0.448	**1.190**	$2.66\times$
S3	5	aircraft	quartic SOS, formal	7301.617	**12167.835**	$1.67\times$
S4	6	three-machine	quartic SOS, screened	165.75	**249.09**	$1.50\times$
S5	10	microgrid	sparse SOS, formal	0.00201	**40.719**	2.03×10^4
S6	20	VdP network	frozen neural	2.094×10^8	$\mathbf{2.365\times10^8}$	$1.13\times$

boundary-aware verification (Figure 1, left): the verified set uses the part of the box boundary where the vector field already points inward. Zubov-Net under a common recipe also uniformly outperforms SOS certificates (Table 2). The baseline for T1 is formally verified, whereas for T2–T4 no certificate was available, so we compare against the published volumes.

6.2 Structured systems from four to twenty dimensions

Table 4 summarizes the results for structured systems of higher dimensions. More details on the certified sets are shown in Appendix C. The 4D swing model (S1) is verified against the exact sine dynamics, not against the cubic model used for polynomial synthesis. The 10D microgrid (S5) uses five two-state anchored networks and a sparse verification procedure; the SOS certificate is synthesized and verified under the same decomposition. The 20D system (S6) verifies ten two-state anchored networks compositionally and shows improvement over the frozen neural models published by Liu et al. (2024a). In all cases, the results for Zubov-Net show improvements, including $12.70\times$ for the exact swing model, $1.67\times$ for the aircraft, and more than 2×10^4 for the sparse microgrid (Appendix C puts the last ratio in context).

6.3 Ablation on anchored architecture and non-polynomial anchors

We use the 5D aircraft example (S3) to illustrate the contribution of the anchor. A plain MLP of the same size, trained under the same conditions, violates $\mathcal{L}_f W < 0$ at 27.1% of 262,144 points sampled near the origin, whereas the anchored network has no violation, as predicted by Proposition 1. The anchored network also shows a moderate (42%) improvement in the certified ROA volume, compared with the affine-corrected version of the plain network. The results are summarized in Table 3.

Our last experiment aims to show that quadratic or polynomial anchors are not required. We consider a planar system that has been shown to have no polynomial (or even analytic) Lyapunov function (Liu & Fitzsimmons, 2026). We train a degree-two homogeneous anchor of the form in equation (6) and a two-layer width-32 correction network N_θ in equation (7), and use dReal to verify the sublevel set $\{W_\theta \leq 0.04154\}$ on $\mathcal{D} = [-1.5, 1.5]^2$ (see Appendix B).

7 Discussion and conclusion

We developed Zubov-Net, an architecture that turns a local Lyapunov certificate, the anchor, into a Lyapunov function that is *correct by construction* near the equilibrium and, at the same time, capable of learning a maximal Lyapunov function away from it. We proved that the certified sublevel sets of anchored networks can contain any compact subset of the constrained region of attraction. Formal verification is employed to rigorously check the learned candidates, and to make the certified sets less conservative we proposed boundary-aware sufficient conditions for positive invariance. Extensive comparisons with sum-of-squares polynomial Lyapunov certificates show that the proposed architecture systematically outperforms them on both low- and high-dimensional examples.

We also showed that the proposed architecture simplifies the verification conditions. Nonetheless, formal verification of neural networks, including the anchored networks proposed here, remains the bottleneck for providing formal guarantees on learned Lyapunov functions. Other directions of interest, which are also limitations of the present work, include extensions to nonsmooth Lyapunov functions and to control Lyapunov functions, where nonsmoothness may be ubiquitous, and compositional verification techniques in which the decomposition is learned rather than given.

REFERENCES

Alessandro Abate, Daniele Ahmed, Mirco Giacobbe, and Andrea Peruffo. Formal synthesis of Lyapunov neural networks. *IEEE Control Systems Letters*, 5(3):773–778, 2021. doi: 10.1109/LCSYS.2020.3005328.

Amir Ali Ahmadi, Miroslav Krstic, and Pablo A. Parrilo. A globally asymptotically stable polynomial vector field with no polynomial Lyapunov function. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 7579–7580, 2011.

Aaron D. Ames, Kevin Galloway, Koushil Sreenath, and Jessy W. Grizzle. Rapidly exponentially stabilizing control Lyapunov functions and hybrid zero dynamics. *IEEE Transactions on Automatic Control*, 59(4):876–891, 2014.

Aaron D. Ames, Xiangru Xu, Jessy W. Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2017.

Matthieu Barreau and Nicola Bastianello. (Un)supervised learning of maximal Lyapunov functions. *arXiv preprint arXiv:2408.17246*, 2024. Version 1 appeared as "Learning and verifying maximal Taylor-neural Lyapunov functions".

Felix Berkenkamp, Matteo Turchetta, Angela P. Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Franco Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999. doi: 10.1016/S0005-1098(99)00113-2.

Fabio Camilli, Lars Grüne, and Fabian Wirth. A generalization of Zubov's method to perturbed systems. *SIAM Journal on Control and Optimization*, 40(2):496–515, 2001a.

Fabio Camilli, Lars Grüne, and Fabian Wirth. A regularization of Zubov's equation for robust domains of attraction. In *Nonlinear Control in the Year 2000, Volume 1*, volume 258 of *Lecture Notes in Control and Information Sciences*, pp. 277–289. Springer, London, 2001b.

Abhijit Chakraborty, Peter Seiler, and Gary J. Balas. Nonlinear region of attraction analysis for flight control verification and validation. *Control Engineering Practice*, 19(4):335–345, 2011.

Ya-Chien Chang, Nima Roohi, and Sicun Gao. Neural Lyapunov control. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Graziano Chesi, Andrea Garulli, Alberto Tesi, and Antonio Vicino. LMI-based computation of optimal quadratic Lyapunov functions for odd polynomial systems. *International Journal of Robust and Nonlinear Control*, 15(1):35–49, 2005. doi: 10.1002/rnc.967.

Hsiao-Dong Chiang, Morris W. Hirsch, and Felix F. Wu. Stability regions of nonlinear autonomous dynamical systems. *IEEE Transactions on Automatic Control*, 33(1):16–27, 1988.

Hongkai Dai, Benoit Landry, Lujie Yang, Marco Pavone, and Russ Tedrake. Lyapunov-stable neural-network control. In *Robotics: Science and Systems XVII*, 2021. doi: 10.15607/RSS.2021.XVII.063.

Edward J. Davison and E. M. Kurak. A computational method for determining quadratic Lyapunov functions for non-linear systems. *Automatica*, 7(5):627–636, 1971.

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*, 39(3):1749–1767, 2023.

Nael H. El-Farra and Panagiotis D. Christofides. Integrating robustness, optimality and constraints in control of nonlinear processes. *Chemical Engineering Science*, 56(5):1841–1868, 2001.

Maxwell Fitzsimmons and Jun Liu. A neural network approach to finding global Lyapunov functions for homogeneous vector fields. *IEEE Control Systems Letters*, 8:670–675, 2024. doi: 10.1109/LCSYS.2024.3402142.

Nathan Gaby, Fumin Zhang, and Xiaojing Ye. Lyapunov-Net: A deep neural network architecture for Lyapunov function approximation. In *Proceedings of the 61st IEEE Conference on Decision and Control*, pp. 2091–2096, 2022. doi: 10.1109/CDC51059.2022.9993006.

Sicun Gao, Soonho Kong, and Edmund M. Clarke. dReal: An SMT solver for nonlinear theories over the reals. In *Formal Methods in Computer-Aided Design*, pp. 208–214, 2013.

Peter Giesl and Sigurdur Hafstein. Review on computational methods for Lyapunov functions. *Discrete and Continuous Dynamical Systems – Series B*, 20(8):2291–2331, 2015.

Lars Grüne. Computing Lyapunov functions using deep neural networks. *Journal of Computational Dynamics*, 8(2):131–152, 2021. doi: 10.3934/jcd.2021006.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks*, 3(5): 551–560, 1990. doi: 10.1016/0893-6080(90)90005-6.

Wei Kang, Kai Sun, and Liang Xu. Data-driven computational methods for the domain of attraction and Zubov's equation. *IEEE Transactions on Automatic Control*, 69(3):1600–1611, 2024. doi: 10.1109/TAC.2023.3326769.

Christopher M. Kellett. Classical converse theorems in Lyapunov's second method. *Discrete and Continuous Dynamical Systems–B*, 20(8):2333–2360, 2015. doi: 10.3934/dcdsb.2015.20.2333.

Hassan K. Khalil. *Nonlinear Systems*. Prentice Hall, Upper Saddle River, NJ, 3 edition, 2002.

L. Khodadadi, B. Samadi, and H. Khaloozadeh. Estimation of region of attraction for polynomial nonlinear systems: A numerical method. *ISA Transactions*, 53(1):25–32, 2014. doi: 10.1016/j.isatra.2013.08.005.

Haoyu Li, Xiangru Zhong, Bin Hu, and Huan Zhang. Two-stage learning of stabilizing neural controllers via Zubov sampling and iterative domain expansion. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Jun Liu and Maxwell Fitzsimmons. On strict verification of neural Lyapunov functions. *IFAC-PapersOnLine*, 59(19):304–309, 2025. doi: 10.1016/j.ifacol.2025.11.051.

Jun Liu and Maxwell Fitzsimmons. A globally asymptotically stable planar homogeneous polynomial vector field with no polynomial Lyapunov function. *arXiv preprint arXiv:2607.16171*, 2026.

Jun Liu, Yiming Meng, Maxwell Fitzsimmons, and Ruikun Zhou. Towards learning and verifying maximal neural Lyapunov functions. In *Proceedings of the 62nd IEEE Conference on Decision and Control*, pp. 8012–8019, 2023.

Jun Liu, Yiming Meng, Maxwell Fitzsimmons, and Ruikun Zhou. Compositionally verifiable vector neural Lyapunov functions for stability analysis of interconnected nonlinear systems. In *Proceedings of the American Control Conference (ACC)*, 2024a.

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 *Proceedings of the 27th ACM International Conference on Hybrid Systems: Computation and Control*, pp. 1–8, 2024b. doi: 10.1145/3641513.3650134.

Jun Liu, Yiming Meng, Maxwell Fitzsimmons, and Ruikun Zhou. Physics-informed neural network Lyapunov functions: PDE characterization, learning, and verification. *Automatica*, 175:112193, 2025. doi: 10.1016/j.automatica.2025.112193.

A. M. Lyapunov. The general problem of the stability of motion. *International Journal of Control*, 55(3):531–534, 1992. Translated by A. T. Fuller from the 1892 Russian original.

Yiming Meng and Jun Liu. Towards learning and verifying maximal Lyapunov–barrier functions with a Zubov PDE formulation. In *Proceedings of the European Control Conference (ECC)*, 2026.

Alessio Merola, Carlo Cosentino, and Francesco Amato. An insight into tumor dormancy equilibrium via the analysis of its domain of attraction. *Biomedical Signal Processing and Control*, 3(3): 212–219, 2008.

Antonis Papachristodoulou and Stephen Prajna. On the construction of Lyapunov functions using the sum of squares decomposition. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 3, pp. 3482–3487, 2002.

Pablo A. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, 2000.

Allan Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8: 143–195, 1999. doi: 10.1017/S0962492900002919.

Maziar Raissi, Paris Perdikaris, and George Em 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. doi: 10.1016/j.jcp.2018.10.045.

Spencer M. Richards, Felix Berkenkamp, and Andreas Krause. The Lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems. In *Proceedings of the 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pp. 466–476, 2018.

Lionel Rosier. Homogeneous Lyapunov function for homogeneous continuous vector field. *Systems & Control Letters*, 19(6):467–473, 1992. doi: 10.1016/0167-6911(92)90078-7.

Zhenyu Sun et al. Safe stabilization of cancer dynamics via formal methods and Lyapunov control: A case study on the Stepanova model. In *Proceedings of the European Control Conference*, 2026.

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. doi: 10.1109/TAC.2007.914221.

Russ Tedrake, Ian R. Manchester, Mark Tobenkin, and John W. Roberts. LQR-trees: Feedback motion planning via sums-of-squares verification. *The International Journal of Robotics Research*, 29(8):1038–1052, 2010.

Ufuk Topcu, Andrew Packard, and Peter Seiler. Local stability analysis using simulations and sum-of-squares programming. *Automatica*, 44(10):2669–2675, 2008. doi: 10.1016/j.automatica.2008.03.010.

Anthony Vannelli and Mathukumalli Vidyasagar. Maximal Lyapunov functions and domains of attraction for autonomous nonlinear systems. *Automatica*, 21(1):69–80, 1985.

Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J. Zico Kolter. β-CROWN: Efficient bound propagation with per-neuron split constraints for neural network robustness verification. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Felix Wiebe, Shivesh Kumar, Lasse J. Shala, Shubham Vyas, Mahdi Javadi, and Frank Kirchner. Open source dual-purpose Acrobot and Pendubot platform: Benchmarking control algorithms for underactuated robotics. *IEEE Robotics & Automation Magazine*, 31(2):113–124, 2024.

Kaidi Xu, Huan Zhang, Shiqi Wang, Yihan Wang, Suman Jana, Xue Lin, and Cho-Jui Hsieh. Fast and complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers. In *International Conference on Learning Representations*, 2021.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Songyuan Zhang, Yumeng Xiu, Guannan Qu, and Chuchu Fan. Compositional neural certificates for networked dynamical systems. In *Proceedings of the 5th Annual Learning for Dynamics and Control Conference*, volume 211 of *Proceedings of Machine Learning Research*, pp. 272–285, 2023.

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*, volume 35, pp. 29113–29125, 2022.

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

REPRODUCIBILITY STATEMENT

Code will be made publicly available. The implementation builds on the open-source LyZNet toolbox (Liu et al., 2024b). The accompanying repository contains the benchmark definitions, the training and verification scripts, the trained networks and polynomial certificates behind every number reported in Section 6, their proof summaries, and the scripts that produce the tables and figures. Appendix D lists the software versions and gives the details of verification, volume estimation, and the ablation.

AI USE STATEMENT

AI tools assisted with code development for the benchmark evaluations. They also helped polish the language and the structure of the paper. The authors are responsible for all technical results and for the final version of the paper.

A PROOFS

This appendix proves the results of Sections 4 and 5. It is organized as follows.

- Appendix A.1 proves Proposition 1.
- Appendix A.2 constructs the maximal Lyapunov function used in the proof of Theorem 1, establishes its properties, and gives sufficient conditions for the relative openness of $\mathcal{A}_{\mathcal{D}}$.
- Appendix A.3 proves Theorem 1.
- Appendix A.4 proves an elementary invariance lemma.
- Appendices A.5 and A.6 prove Propositions 2 and 3.

Throughout, $\phi(t, x)$ denotes the solution of $\dot{x} = f(x)$ with $\phi(0, x) = x$ on its maximal interval of existence.

A.1 PROOF OF PROPOSITION 1

Because V is positive definite and $e^{G_\theta} > 0$, the function $H_\theta = Ve^{G_\theta}$ is positive definite, and so is $W_\theta = \tanh H_\theta$. It remains to prove strict decrease. Since G_θ is C^1, the constant $L_\theta := \max_{\|x\| \le r_0} \|\nabla G_\theta(x)\|$ is finite. Differentiating the anchored candidate gives the identity

$$\mathcal{L}_f H_\theta(x) = e^{G_\theta(x)} \left[\mathcal{L}_f V(x) + V(x) \nabla G_\theta(x)^\top f(x) \right]. \tag{22}$$

For $0 < \|x\| \le r_0$, Assumption 1 bounds the correction term by $|V \nabla G_\theta^\top f| \le C_V C_f L_\theta \|x\|^{p+d}$. Consequently,

$$\mathcal{L}_f H_\theta(x) \le e^{G_\theta(x)} \|x\|^{p+d-1} \left(-\alpha + C_V C_f L_\theta \|x\| \right), \qquad 0 < \|x\| \le r_0. \tag{23}$$

The right-hand side is strictly negative whenever $0 < \|x\| < \min\{r_0, \alpha/(C_V C_f L_\theta)\}$, where the second entry is interpreted as $+\infty$ if $L_\theta = 0$. Finally, $\mathcal{L}_f W_\theta = (1 - \tanh^2 H_\theta) \mathcal{L}_f H_\theta$, and the factor $1 - \tanh^2 H_\theta$ is strictly positive. Hence W_θ decreases strictly on the same punctured ball. □

A.2 THE MAXIMAL LYAPUNOV FUNCTION USED IN THE PROOF OF THEOREM 1

Throughout this subsection and the next, Assumption 1 holds. Shrinking r_0 if necessary, we assume that the closed ball $\bar{B}_{r_0} = \{x : \|x\| \le r_0\}$ lies in $\operatorname{int} \mathcal{D}$.

Local core. Fix c_V with $0 < c_V < \min_{\|x\|=r_0} V(x)$ and set

$$C_0 = \{x \in \bar{B}_{r_0} : V(x) \le c_V\}, \qquad B_0 = \{x \in \mathbb{R}^n : \|x\| < r_0,\ V(x) < c_V\}. \tag{24}$$

The set B_0 is an open neighborhood of the origin, C_0 is compact, and $B_0 \subseteq C_0 \subseteq \{x : \|x\| < r_0\}$, because the condition $V(x) \le c_V$ excludes the sphere $\|x\| = r_0$. By equation (14), V is nonincreasing along trajectories while they remain in $\bar{B}_{r_0}$. A trajectory starting in C_0 therefore cannot reach the sphere $\|x\| = r_0$, on which $V > c_V$, so C_0 and B_0 are positively invariant. Moreover, every trajectory in C_0 converges to the origin: V decreases along it to a limit $v^* \ge 0$, and if $v^* > 0$ the trajectory would remain in the compact set $\{x \in C_0 : V(x) \ge v^*\}$, which excludes the origin and on which equation (14) gives $\mathcal{L}_f V \le -\mu$ for some $\mu > 0$, forcing $V \to -\infty$. Hence $C_0 \subseteq \mathcal{A}_\mathcal{D}$, and C_0 is a local core of the kind used in Proposition 2.

Matched running cost. Let $\omega : \mathbb{R}^n \to \mathbb{R}$ be continuous, positive definite, continuously differentiable on $\mathbb{R}^n \setminus \{0\}$, and equal to $-\mathcal{L}_f V$ on C_0. Such a cost exists: take $\omega = \chi\,(-\mathcal{L}_f V) + (1-\chi)\|x\|^2$, where $\chi : \mathbb{R}^n \to [0,1]$ is a smooth cutoff that equals one on C_0 and has compact support in $\{x : \|x\| < r_0\}$; on that support, $-\mathcal{L}_f V$ is positive away from the origin by equation (14) and continuously differentiable away from the origin because V is C^2 there. Define

$$U(x) = \int_0^\infty \omega(\phi(t,x))\,dt, \qquad x \in \mathcal{A}_\mathcal{D}, \tag{25}$$

as in equation (9), let $g = \log(U/V)$ on $\mathcal{A}_\mathcal{D} \setminus \{0\}$ with $g(0) = 0$, and let $K_\ell = \{x \in \mathcal{A}_\mathcal{D} : U(x) \le \ell\}$ for $\ell < \infty$.

Lemma 1 (A maximal Lyapunov function with a regular log-ratio)**.** *Under Assumption 1, the following statements hold.*

1. *U is finite and continuous on $\mathcal{A}_\mathcal{D}$, $U = V$ on C_0, and $U(x) > 0$ for $x \neq 0$.*
2. *For every $x \in \mathcal{A}_\mathcal{D}$, the function $t \mapsto U(\phi(t,x))$ is continuously differentiable on $[0,\infty)$ with derivative $-\omega(\phi(t,x))$; in particular, U is nonincreasing along trajectories.*
3. *g vanishes on C_0, and for every compact set $K \subseteq \mathcal{A}_\mathcal{D}$ there is $\tilde{g} \in C^1(\mathbb{R}^n)$ with $\tilde{g} = g$ on K.*
4. *For every $\ell < \infty$, the set K_ℓ is compact and positively invariant, and $\bigcup_{\ell<\infty} K_\ell = \mathcal{A}_\mathcal{D}$.*

Proof. Item 1, first part. Let $y \in C_0$. Its trajectory remains in C_0, where $\omega(\phi(t,y)) = -\frac{d}{dt}V(\phi(t,y))$, and converges to the origin, where V vanishes; integrating gives $U(y) = V(y)$. Now let $x \in \mathcal{A}_\mathcal{D}$. Its trajectory converges to the origin, so $\phi(T,x) \in B_0$ for some $T \ge 0$, and the semigroup property gives

$$U(x) = \int_0^T \omega(\phi(t,x))\,dt + V(\phi(T,x)) < \infty. \tag{26}$$

If $x \neq 0$, then $\phi(t,x) \neq 0$ for all t, so the integrand in equation (25) is positive and $U(x) > 0$.

Item 2. By the semigroup property, $U(\phi(t,x)) = \int_t^\infty \omega(\phi(s,x))\,ds$, and the integrand is continuous in s; the claim is the fundamental theorem of calculus.

Item 3 and continuity of U. On C_0, $U = V$, so $g = 0$. Fix $x^* \in \mathcal{A}_\mathcal{D}$. If $x^* \in B_0$, then $g = 0$ and $U = V$ on the open set B_0, so U is continuous at x^*. Otherwise $x^* \neq 0$; choose T with $\phi(T,x^*) \in B_0$. Since $\phi(T,\cdot)$ is continuously differentiable on an open set containing x^* and B_0 is open, there is an open ball $O \ni x^*$ on which $\phi(T,\cdot)$ is defined and maps into B_0, and the right-hand side of equation (26) defines a function $\tilde{U}$ on O that agrees with U on $O \cap \mathcal{A}_\mathcal{D}$. Because $\phi(t,x^*) \neq 0$ for $t \in [0,T]$, after shrinking O the arcs $\{\phi(t,x) : t \in [0,T]\}$, $x \in O$, lie in a fixed compact subset of $\mathbb{R}^n \setminus \{0\}$, where ω is C^1; hence $\tilde{U} \in C^1(O)$ by differentiation under the integral sign. Shrinking O once more so that $\tilde{U} > 0$ on O and $0 \notin O$, the function $\log(\tilde{U}/V)$ is C^1 on O and equals g on $O \cap \mathcal{A}_\mathcal{D}$. In particular, U is continuous at x^*, which completes item 1. Now let $K \subseteq \mathcal{A}_\mathcal{D}$ be compact. Cover K by finitely many such open sets $O_1, \dots, O_N$ with associated C^1 functions $g_1, \dots, g_N$, each equal to g on $O_j \cap \mathcal{A}_\mathcal{D}$, and let $\chi_0, \chi_1, \dots, \chi_N$ be a smooth partition of unity subordinate to the open

cover $\mathbb{R}^n \setminus K, O_1, \ldots, O_N$ of $\mathbb{R}^n$. Then $\tilde{g} = \sum_{j=1}^{N} \chi_j g_j$ is C^1 on $\mathbb{R}^n$ and, because $\chi_0 = 0$ on K, equals g on K.

Item 4. Positive invariance follows from item 2, and $\bigcup_\ell K_\ell = \mathcal{A}_\mathcal{D}$ from the finiteness of U. Since $K_\ell \subseteq \mathcal{D}$ is bounded, it remains to show that K_ℓ is closed in $\mathbb{R}^n$. Let $x_k \in K_\ell$ converge to y; then $y \in \mathcal{D}$. Suppose that $y \notin \mathcal{A}_\mathcal{D}$. If $\phi(t_1, y) \notin \mathcal{D}$ for some $t_1 \geq 0$, then, because $\mathcal{D}$ is closed and $\phi(t_1, \cdot)$ is continuous, $\phi(t_1, x_k) \notin \mathcal{D}$ for all large k, contradicting $x_k \in \mathcal{A}_\mathcal{D}$; the same argument shows that $\phi(\cdot, y)$ exists for all $t \geq 0$. Hence $\phi(t, y) \in \mathcal{D}$ for all $t \geq 0$. Moreover, $\phi(t, y) \notin B_0$ for all $t \geq 0$, because every trajectory that enters B_0 converges to the origin, which would place y in $\mathcal{A}_\mathcal{D}$. The set $\mathcal{D} \setminus B_0$ is compact and excludes the origin, so $\omega \geq \omega_0$ on it for some $\omega_0 > 0$, and for every $T > 0$, continuous dependence on initial conditions over $[0, T]$ gives

$$\liminf_{k\to\infty} U(x_k) \geq \lim_{k\to\infty} \int_0^T \omega(\phi(t, x_k))\, dt = \int_0^T \omega(\phi(t, y))\, dt \geq \omega_0 T.$$

Thus $U(x_k) \to \infty$, contradicting $U(x_k) \leq \ell$. Therefore $y \in \mathcal{A}_\mathcal{D}$, and $U(y) \leq \ell$ by continuity. □

The next lemma gives the two sufficient conditions for relative openness stated in Section 4. Condition (b) is the boundary condition in equation (17), imposed on the unknown set $\mathcal{A}_\mathcal{D} \cap \partial\mathcal{D}$ rather than on a certified sublevel set.

Lemma 2 (Sufficient conditions for relative openness)**.** *Let Assumption 1 hold. Then $\mathcal{A}_\mathcal{D}$ is relatively open in $\mathcal{D}$ if either (a) $\mathcal{D}$ is positively invariant, or (b) $\mathcal{L}_f h_i(x) < 0$ for every $x \in \mathcal{A}_\mathcal{D} \cap \partial\mathcal{D}$ and every $i \in I(x)$.*

Proof. Let $x \in \mathcal{A}_\mathcal{D}$ and write $\gamma(t) = \phi(t, x)$. Since $\gamma(t) \to 0$, there is $T \geq 0$ with $\gamma(T) \in B_0$, and by continuity of $\phi(T, \cdot)$ there is a neighborhood N of x with $\phi(T, N) \subseteq B_0$; every trajectory starting in N that remains in $\mathcal{D}$ on $[0, T]$ therefore converges to the origin without leaving $\mathcal{D}$. In case (a), every trajectory starting in $N \cap \mathcal{D}$ remains in $\mathcal{D}$, so $N \cap \mathcal{D} \subseteq \mathcal{A}_\mathcal{D}$.

In case (b), we first show that $\gamma(t) \in \operatorname{int}\mathcal{D}$ for all $t > 0$. If $\gamma(t_1) \in \partial\mathcal{D}$ for some $t_1 > 0$, then $\gamma(t_1) \in \mathcal{A}_\mathcal{D} \cap \partial\mathcal{D}$ and some constraint h_i is active there, so $\mathcal{L}_f h_i(\gamma(t_1)) < 0$ and $h_i(\gamma(t)) > 1$ for $t < t_1$ close to t_1, contradicting $\gamma(t) \in \mathcal{D}$. If $x \in \operatorname{int}\mathcal{D}$, the arc $\gamma([0, T])$ is a compact subset of $\operatorname{int}\mathcal{D}$, so after shrinking N the arcs $\phi([0, T], x')$, $x' \in N$, lie in $\operatorname{int}\mathcal{D}$, and $N \subseteq \mathcal{A}_\mathcal{D}$. If $x \in \partial\mathcal{D}$, then $\mathcal{L}_f h_i(x) < 0$ for $i \in I(x)$ and $h_j(x) < 1$ for $j \notin I(x)$. By continuity, these inequalities hold on a neighborhood N_0 of x, and there are $\delta \in (0, T)$ and a neighborhood $N_1 \subseteq N$ of x with $\phi([0, \delta], N_1) \subseteq N_0$. For $x' \in N_1 \cap \mathcal{D}$ and $t \in [0, \delta]$, we have $h_i(\phi(t, x')) \leq h_i(x') \leq 1$ for $i \in I(x)$ and $h_j(\phi(t, x')) < 1$ for $j \notin I(x)$, so $\phi([0, \delta], x') \subseteq \mathcal{D}$. The arc $\gamma([\delta, T])$ is a compact subset of $\operatorname{int}\mathcal{D}$, so after shrinking N_1 the arcs $\phi([\delta, T], x')$, $x' \in N_1$, lie in $\operatorname{int}\mathcal{D}$. Hence $N_1 \cap \mathcal{D} \subseteq \mathcal{A}_\mathcal{D}$. □

A.3 Proof of Theorem 1

Let $K \subseteq \mathcal{A}_\mathcal{D}$ be compact. By Lemma 1(1), U is bounded on K; set $a = 1 + \max_K U$ and $b = a + 1$, so that $K \subseteq K_a \subseteq K_b$, and both sublevel sets are compact by Lemma 1(4). We construct G_θ by approximating g on K_b with an accuracy $\eta \in (0, 1]$, on which Steps 3 and 4 impose three smallness conditions, and we set $c = (a + b)/2$. Step 5 proves the exhaustion claim.

Step 1: approximate the regular log-ratio. By Lemma 1(3), there is $\tilde{g} \in C^1(\mathbb{R}^n)$ with $\tilde{g} = g$ on K_b. Because the activation is C^1 and nonpolynomial, networks with one hidden layer are dense in $C^1(\mathbb{R}^n)$ with respect to uniform convergence of functions and first derivatives on compact sets (Pinkus, 1999, Theorem 4.1); see also Hornik et al. (1990). Hence there is a network N_θ with

$$\|N_\theta - \tilde{g}\|_{C^1(K_b)} := \sup_{K_b} |N_\theta - \tilde{g}| + \sup_{K_b} \|\nabla N_\theta - \nabla \tilde{g}\| < \eta/2. \tag{27}$$

Set $G_\theta = N_\theta - N_\theta(0)$. Since $\tilde{g}(0) = g(0) = 0$, we have $|N_\theta(0)| < \eta/2$. Centering leaves the gradient unchanged and gives

$$\|G_\theta - \tilde{g}\|_{C^1(K_b)} < \eta. \tag{28}$$

Thus subtracting the equilibrium output does not reduce the approximation class.

Step 2: transfer C^1 accuracy through the anchor. The function $\tilde{U} = Ve^{\tilde{g}}$ is C^1 on $\mathbb{R}^n$ and equals U on K_b: on $K_b \setminus \{0\}$ by the definition of g, and at the origin because both sides vanish there. The functions V, ∇V, $\tilde{g}$, and $\nabla\tilde{g}$ are bounded on the compact set K_b, and equation (28) with $\eta \le 1$ bounds G_θ and ∇G_θ on K_b uniformly in η. Write

$$H_\theta - \tilde{U} = V\,(e^{G_\theta} - e^{\tilde{g}}), \qquad \nabla(H_\theta - \tilde{U}) = (e^{G_\theta} - e^{\tilde{g}})(\nabla V + V\nabla\tilde{g}) + e^{G_\theta} V\,(\nabla G_\theta - \nabla\tilde{g}).$$

The mean-value theorem gives $|e^{G_\theta} - e^{\tilde{g}}| \le e^{\max\{G_\theta,\tilde{g}\}}|G_\theta - \tilde{g}|$. Therefore there is a constant C_b, independent of $\eta \in (0, 1]$, such that

$$\|H_\theta - \tilde{U}\|_{C^1(K_b)} \le C_b\eta. \tag{29}$$

Step 3: preserve strict decrease, including near the origin. We first treat a ball whose radius does not depend on η. Since V is continuous with $V(0) = 0$ and B_0 is an open neighborhood of the origin, there is $r_1 > 0$ with $\bar{B}_{r_1} \subseteq B_0$ and $V \le b$ on $\bar{B}_{r_1}$; by Lemma 1(1), $U = V$ on $\bar{B}_{r_1}$, so $\bar{B}_{r_1} \subseteq K_b$. For all $\eta \le 1$, equation (28) gives the common bound

$$\sup_{K_b} \|\nabla G_\theta\| \le 1 + \sup_{K_b} \|\nabla\tilde{g}\| =: L.$$

Choose

$$0 < r_* < \min\left\{r_0,\ r_1,\ \frac{\alpha}{C_V C_f L}\right\}. \tag{30}$$

The derivation of equation (23) applies on $\bar{B}_{r_*} \subseteq K_b$ with L in place of L_θ. It shows that $\mathcal{L}_f H_\theta(x) < 0$ for $0 < \|x\| \le r_*$ and every $\eta \le 1$. Thus the local radius does not collapse as the approximation is refined.

We next treat the remaining set $Q = K_b \cap \{x : \|x\| \ge r_*\}$, which is compact and excludes the origin. For $x \in K_b$, the positive invariance of K_b (Lemma 1(4)) gives $\tilde{U}(\phi(t, x)) = U(\phi(t, x))$ for all $t \ge 0$, so the derivative at $t = 0$ of the C^1 function $t \mapsto \tilde{U}(\phi(t, x))$, which equals $\nabla\tilde{U}(x)^\top f(x)$, coincides with the right derivative $-\omega(x)$ supplied by Lemma 1(2). Hence $\mathcal{L}_f\tilde{U} = -\omega$ on K_b. Set

$$m = \min_Q \omega > 0, \qquad B = \max_Q \|f\| < \infty.$$

By equation (29),

$$\mathcal{L}_f H_\theta = \mathcal{L}_f\tilde{U} + \nabla(H_\theta - \tilde{U})^\top f \le -m + BC_b\eta < 0 \qquad \text{on } Q,$$

provided that $\eta < m/(BC_b)$. Combining the two cases yields $\mathcal{L}_f H_\theta < 0$ on $K_b \setminus \{0\}$.

Step 4: sandwich the origin component. Require, in addition, that $C_b\eta < (b - a)/2$. Then equation (29) gives

$$H_\theta < c \text{ on } K_a, \qquad H_\theta > c \text{ on } \{x \in K_b : U(x) = b\}. \tag{31}$$

The set $S_{\theta,c}$ is a connected component of the compact set $\mathcal{D} \cap \{H_\theta \le c\}$ and is therefore compact.

We first show that $K_a \subseteq S_{\theta,c}$. Let $x \in K_a$. Its trajectory remains in $\mathcal{D}$, converges to the origin, and stays in K_a because U is nonincreasing along it. Appending the limit point 0 yields a path from x to the origin inside K_a. Hence K_a is a connected subset of $\{x \in \mathcal{D} : H_\theta(x) \le c\}$ that contains the origin, and $K_a \subseteq S_{\theta,c}$.

We next show that $S_{\theta,c} \subseteq K_b$. The key observation is that

$$\partial_{\mathcal{D}} K_b \subseteq \{x \in K_b : U(x) = b\}, \tag{32}$$

where $\partial_{\mathcal{D}}$ denotes the boundary relative to $\mathcal{D}$. Indeed, K_b is closed, so $\partial_{\mathcal{D}} K_b \subseteq K_b$. If $x \in K_b$ and $U(x) < b$, then the relative openness of $\mathcal{A}_{\mathcal{D}}$ in $\mathcal{D}$ and the continuity of U provide a relative neighborhood of x in $\mathcal{D}$ that lies in $\mathcal{A}_{\mathcal{D}} \cap \{U < b\} \subseteq K_b$; thus x is a relative interior point of K_b. The inclusion in equation (32) can be strict, for example where the level set $\{U = b\}$ meets $\partial\mathcal{D}$; only the inclusion is needed. Now suppose that $S_{\theta,c} \not\subseteq K_b$. Then $S_{\theta,c}$ is a connected subset of $\mathcal{D}$ that meets both the relative interior of K_b, at the origin, and the relatively open set $\mathcal{D} \setminus K_b$. It must therefore meet $\partial_{\mathcal{D}} K_b$, where $H_\theta > c$ by equation (31) and equation (32). This contradicts $H_\theta \le c$ on $S_{\theta,c}$.

Condition (i) of the theorem now follows from $K \subseteq K_a$ and $K_b \subseteq \mathcal{A}_{\mathcal{D}}$, and (ii) from Step 3. For (iii), let $x \in S_{\theta,c}$ and $t \ge 0$. Since $x \in K_b$ and K_b is positively invariant, the arc $\{\phi(s, x) : 0 \le s \le t\}$ lies in $K_b \subseteq \mathcal{D}$, and H_θ is nonincreasing along it by (ii), so the arc lies in $\{y \in \mathcal{D} : H_\theta(y) \le c\}$. Being connected and containing x, the arc lies in the component $S_{\theta,c}$. Hence $S_{\theta,c}$ is positively invariant.

Step 5: exhaust the constrained ROA. Apply the preceding construction to K_1 to obtain S_1, and then, for $k \geq 2$, to the compact set $S_{k-1} \cup K_k \subseteq \mathcal{A}_\mathcal{D}$ to obtain $S_k \supseteq S_{k-1} \cup K_k$. The sets S_k are nested, each lies in $\mathcal{A}_\mathcal{D}$, and their union contains $\bigcup_k K_k = \mathcal{A}_\mathcal{D}$ by Lemma 1(4). □

A.4 An invariance lemma

The set certified by Proposition 2 is cut out by finitely many C^1 inequalities, and the vector field points strictly inward at every active inequality. In this situation, positive invariance has an elementary proof that needs no constraint qualification. The condition is the strict form of Nagumo's tangent-cone condition (Blanchini, 1999).

Lemma 3 (Strict inward flow implies positive invariance)**.** *Let $\psi_1, \ldots, \psi_k$ be C^1 functions on an open set $O \subseteq \mathbb{R}^n$, and let $M = \{x \in O : \psi_j(x) \leq 0,\ j = 1, \ldots, k\}$. Let $S \subseteq M$ be compact and a union of connected components of M. Suppose that $\mathcal{L}_f \psi_j(x) < 0$ for every $x \in S$ and every j with $\psi_j(x) = 0$. Then every trajectory starting in S exists and remains in S for all $t \geq 0$.*

Proof. Let $x_0 \in S$, let $[0, T)$ be the maximal interval of existence of $\phi(\cdot, x_0)$, and define $t^* = \sup\{t \in [0, T) : \phi(s, x_0) \in S \text{ for all } s \in [0, t]\}$. Suppose that $t^* < T$. Because S is closed, $y := \phi(t^*, x_0) \in S$. If $\psi_j(y) = 0$, then the function $t \mapsto \psi_j(\phi(t, y))$ vanishes at $t = 0$ and has derivative $\mathcal{L}_f \psi_j(y) < 0$ there, so it is negative for all sufficiently small $t > 0$. If $\psi_j(y) < 0$, the same conclusion follows from continuity. Hence there is $\tau > 0$ such that $\phi(t, y) \in M$ for all $t \in [0, \tau]$. This arc is a connected subset of M that contains $y \in S$. It therefore lies in S, because S is a union of connected components of M. Thus $\phi(s, x_0) \in S$ for all $s \in [0, t^* + \tau]$, which contradicts the definition of t^*. Therefore $t^* = T$. Since the trajectory remains in the compact set S on $[0, T)$, it cannot blow up in finite time, and $T = \infty$. □

A.5 Proof of Proposition 2

Let $S = \mathcal{D} \cap \{W \leq c_{\text{out}}\}$, which is compact. We first prove that S is positively invariant. Apply Lemma 3 with $\psi_i = h_i - 1$ for $i = 1, \ldots, m$, $\psi_{m+1} = W - c_{\text{out}}$, and $M = S$. Let $x \in S$. If $W(x) = c_{\text{out}}$, then equation (16) gives $\mathcal{L}_f W(x) \leq -\epsilon < 0$. If $I(x) \neq \emptyset$, then the first alternative in equation (17) fails because $W(x) \leq c_{\text{out}}$; hence $\mathcal{L}_f h_i(x) \leq -\delta_i < 0$ for every $i \in I(x)$. At a point where several constraints are active, or where a constraint and the level $W = c_{\text{out}}$ are active together, all these inequalities hold simultaneously. Lemma 3 therefore applies.

Now consider a trajectory starting at $x_0 \in S$. By positive invariance, it remains in $\mathcal{D} \cap \{W \leq c_{\text{out}}\}$. As long as $W(\phi(t, x_0)) \geq c_{\text{in}}$, equation (16) yields

$$W(\phi(t, x_0)) \leq W(x_0) - \epsilon t.$$

The trajectory therefore enters $\mathcal{D} \cap \{W \leq c_{\text{in}}\}$ no later than $t = (c_{\text{out}} - c_{\text{in}})/\epsilon$. By equation (15), it then lies in C_0, which is positively invariant and on which convergence to the origin is certified. Since the trajectory never leaves $\mathcal{D}$, we conclude that $x_0 \in \mathcal{A}_\mathcal{D}$. □

A.6 Proof of Proposition 3

The map $\tanh : [0, \infty) \to [0, 1)$ is strictly increasing with inverse atanh, and $H_\theta \geq 0$. This proves equation (18). By equation (8), $\mathcal{L}_f W_\theta = (1 - W_\theta^2) e^{G_\theta} R_\theta$, and both factors multiplying R_θ are strictly positive. This proves equation (19). Next, let $x \in \mathcal{D}$ satisfy $V(x) \geq c_V$. If equation (20) holds, then

$$H_\theta(x) = V(x) e^{G_\theta(x)} > c_V \cdot \frac{\gamma(c_{\text{in}})}{c_V} = \gamma(c_{\text{in}}),$$

and therefore $W_\theta(x) > c_{\text{in}}$ by equation (18). The contrapositive is the claimed inner containment. Finally, the annulus Q is a closed subset of the compact set $\mathcal{D}$, hence compact, and the continuous factor $(1 - W_\theta^2) e^{G_\theta}$ is positive on it, so the minimum in equation (21) is positive; multiplying $R_\theta \leq -\epsilon_R$ by this factor gives $\mathcal{L}_f W_\theta \leq -\epsilon$ on Q by equation (8). □

B Homogeneous anchors

This appendix complements the homogeneous-anchor construction of Section 3: it states the verification obligations of that branch and gives the cubic-leading system of Section 6.3.

B.1 VERIFICATION OBLIGATIONS

The homogeneous branch uses a radial split in place of the two-stage proof of Proposition 2. A unit-sphere inequality and an analytic perturbation bound prove decrease on a punctured ball $0 < \|x\| \le \rho$; the nonlinear verifier proves decrease on the remainder of the candidate sublevel set and excludes that sublevel set from the domain boundary. The formal obligations are therefore the complete unit-sphere decrease inequality, the analytic local perturbation bound, the outer-shell decrease inequality, and one exclusion query for each box face. The interpolated running cost constructs the training target, whereas the inner–outer split certifies the trained candidate; the two devices are complementary but logically independent.

B.2 THE CUBIC-LEADING SYSTEM

The system of Section 6.3 is, with $r^2 = x_1^2 + x_2^2$,

$$\dot{x} = f_3(x) + \sigma(x)\, x, \qquad f_3(x) = \begin{bmatrix} 4x_1^3 - x_1^2 x_2 - 6x_1 x_2^2 - x_2^3 \\ x_1^3 + 4x_1^2 x_2 + x_1 x_2^2 - 6x_2^3 \end{bmatrix},$$
$$\sigma(x) = \tfrac{r^4}{5}\,(3 + \sin x_1 + \cos x_2) + \tfrac{r^6}{20}\,(3 + \cos(x_1 + x_2) + \sin(x_1 - x_2)). \tag{33}$$

Here $Df(0) = 0$, and the leading field f_3 is homogeneous of degree three and asymptotically stable, so Assumption 1 is met by a homogeneous anchor of degree two with $d = 3$. The candidate is a 48-feature anchor of the form in equation (6) with a two-layer width-32 correction network, trained with an interpolated running cost that equals $-\mathcal{L}_f V_p$ on the certified local core. The proof establishes the homogeneous core inequality on the complete unit circle, closes the local perturbation bound for $0 < \|x\| \le 4 \times 10^{-5}$, proves decrease on the remaining outer shell, and excludes the sublevel set from all four box faces.

C BENCHMARK DETAILS

Table 5: Proof and evaluation domains, in physical coordinates, for the planar (left) and three-dimensional (right) common-recipe benchmarks.

case	domain
P1	$[-2.5, 2.5] \times [-3.5, 3.5]$
P2	$[-2, 3] \times [-3, 1.5]$
P3	$[-3, 3]^2$
P4	$[-2, 3] \times [-3, 2]$; SOS on $[-2, 2]^2$
P5	$[-8, 8]^2$
P6	$[-3, 3]^2$
P7	$[-10, 10] \times [-6, 8]$
P8	$[0, 1]^2$
P9	$[-6, 6]^2$
P10	$[-3, 3] \times [-6, 6]$
P11	$[-4, 4]^2$
P12	$[-8, 8] \times [-12, 12]$

case	domain
T1	$[-4, 4] \times [-8, 8] \times [-16, 16]$
T2	$[-3, 3]^3$
T3	$[-10, 10]^3$
T4	$[-8, 8]^3$

Table 5 lists the domains of the common-recipe benchmarks. The structured labels map as follows: S1 is the exact 4D full-swing system, S2 the exact 4D Pendubot, S3 the 5D aircraft, S4 the 6D three-machine swing system, S5 the 10D five-bus microgrid, and S6 the 20D networked Van der Pol system. The release snapshot uses only these labels; the general benchmark library uses descriptive system names.

Sources. P1 and P10 are the reversed Van der Pol oscillator with $\mu = 1$ and $\mu = 3$. P2, P3, and P4 are the two-machine power system, the system with the exact semialgebraic basin $x_1 x_2 < 1$, and the shifted Lotka–Volterra model of Vannelli & Vidyasagar (1985). P5, P6, and T1 are the systems S1, S4, and S5 of Chesi et al. (2005), also studied as examples E1–E3 of Topcu et al. (2008), and P12 is the system S2 of Chesi et al. (2005). P7 is from Khodadadi et al. (2014). P8 is the uncontrolled

Stepanova tumour–immune model of Sun et al. (2026). P9 is the polynomial oscillator $\dot{x}_1 = x_2$, $\dot{x}_2 = -2x_1 + x_1^3/3 - x_2$ with two saddle equilibria. P11 is the triangular system of Ahmadi et al. (2011). T2 is the Davison–Kurak system (Davison & Kurak, 1971), and T3, T4, and S3 are examples E5 and E6 and the aircraft model of Topcu et al. (2008). S1 and S4 are the four- and six-state full swing equations of two and three machines coupled to an infinite bus, with sine coupling. S2 is the Pendubot of Wiebe et al. (2024) under LQR feedback. S5 is the five-bus microgrid of Zhang et al. (2023), and S6 is the networked Van der Pol system of Liu et al. (2024a). The repository contains every model.

Higher-dimensional architectures. The exact four-dimensional two-machine swing model uses a 2-by-40 anchored network and verifies $W \leq 0.581192047$. Its quartic SOS baseline, which we verified ourselves against the same dynamics, has level one. The exact Pendubot frontier repair uses a 2-by-30 anchored network and verifies $W \leq 0.04$; the polynomial comparator is tied to a cubic Taylor model of the dynamics. The five-dimensional aircraft uses a shallow 1-by-64 anchored network and a shell proof through $W = 0.2755$. The six-dimensional three-machine model uses three two-state networks. The ten-dimensional microgrid uses five two-state networks and verifies the compositional level 0.3. The twenty-dimensional Van der Pol network uses ten two-state components.

Formal-proof scope. The 16 formal polynomial comparisons are the twelve planar cases, T1, S1, S3, and S5. The 6D polynomial volume in Table 4 passed dense screening but has no complete formal proof, so it is not counted; neither are the three source-reported 3D baselines, the model-mismatched Pendubot polynomial, or the 20D neural comparator. The microgrid ratio of more than 2×10^4 corresponds to a factor of about 2.7 per coordinate in ten dimensions, and the sparse SOS program maximizes a common decay margin rather than the volume of its sublevel set. Every Zubov-Net volume in the tables belongs to a set with a complete formal proof. For P4, the neural and polynomial certificates use the same dynamics but different domains, as stated in the table above; a polynomial sublevel set that is verified strictly inside the smaller box remains a valid certificate on the larger domain.

Plots. Figures 2–5 show the certified sets of the structured comparisons summarized in Table 4. For dimensions above two, the plots only show slices or components of the certified sets; the sublevel set in the full state space is the object verified.

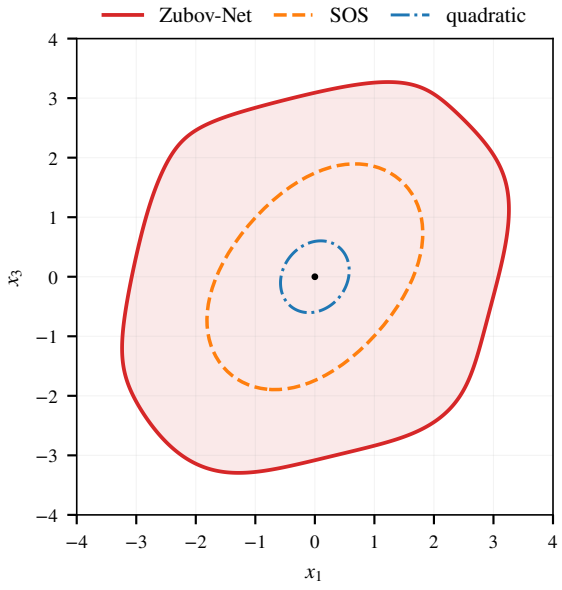

Figure 2: Exact 4D two-machine swing system (S1) on the slice $x_2 = x_4 = 0$: the formally verified Zubov-Net set (red, shaded), the formally verified quartic SOS set (orange, dashed), and the verified quadratic core (blue, dash-dotted).

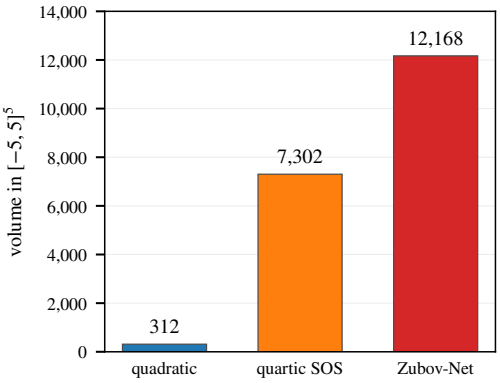

Figure 3: Volumes of the formally verified sets for the constrained 5D aircraft (S3) on the domain $[-5,5]^5$: the quadratic core, the quartic SOS certificate at the level $V \leq 0.5$ that we verified with CROWN, and the Zubov-Net certificate. All volumes are numerical estimates.

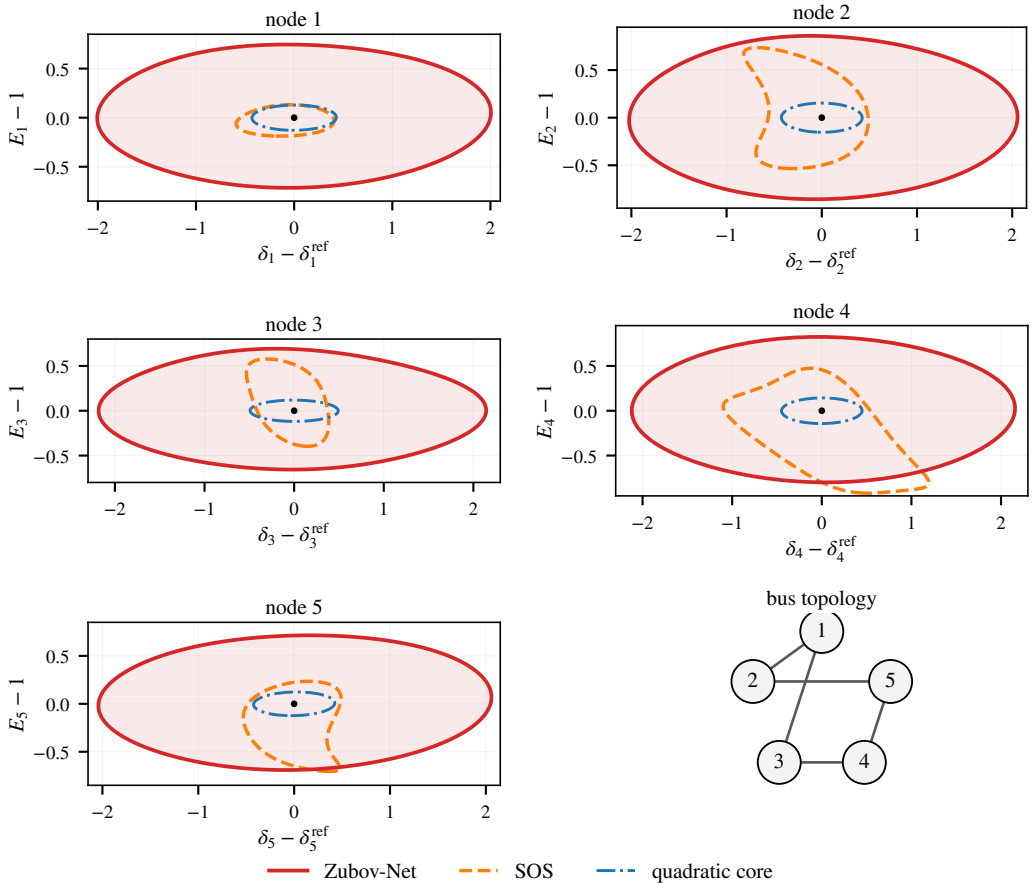

Figure 4: Sparse 10D microgrid (S5). Each node panel shows the coordinates of one bus (angle deviation and voltage deviation) with the other buses at equilibrium: the formally verified Zubov-Net set (red, shaded), the sparse SOS set (orange, dashed), and the quadratic core (blue, dash-dotted). The last panel shows the line topology.

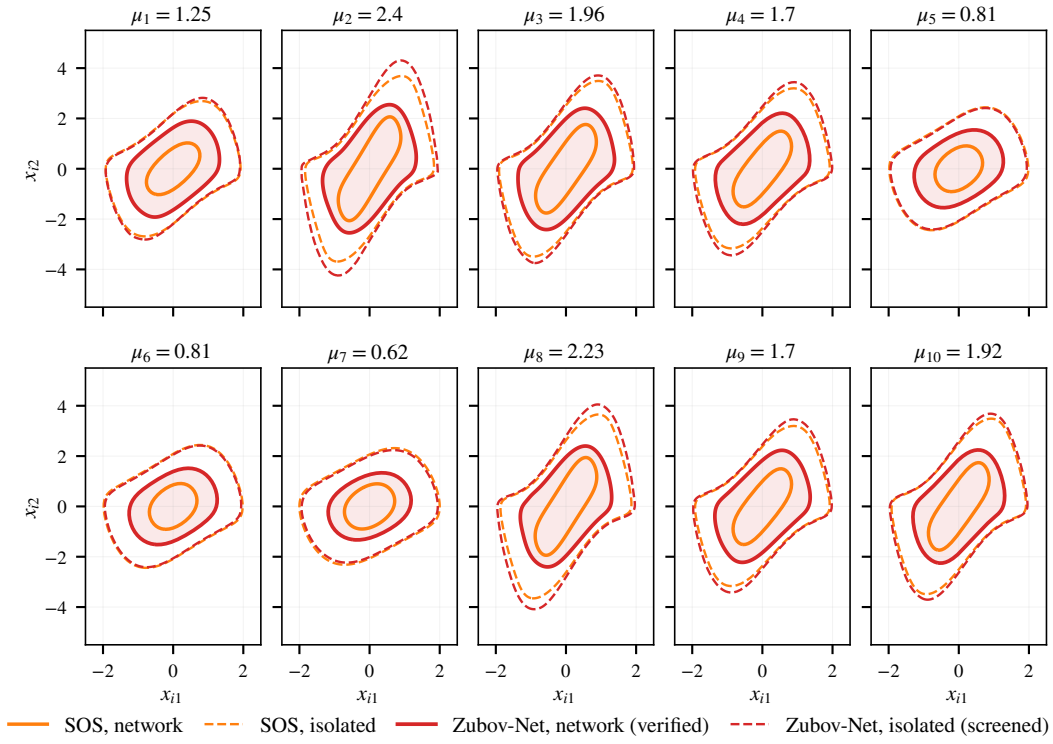

Figure 5: Networked 20D Van der Pol system (S6). Panel i shows subsystem i with parameter μ_i. Solid curves bound the component sets whose product is certified for the coupled network: the sextic SOS certificates published with the example (Liu et al., 2024a) (orange) and the Zubov-Net components verified with CROWN (red, shaded). Dashed curves show the larger levels that each component reaches for the isolated subsystem; the isolated Zubov-Net levels are screened numerically, not verified.

D Reproducibility Details

Software. The implementation builds on the open-source LyZNet toolbox (Liu et al., 2024b), extended with the anchored architecture, the boundary-aware verification conditions, and a CROWN backend. Experiments use Python 3.11, PyTorch 2.11 with CUDA 12.6, α-β-CROWN (commit `e5c7e17`) with auto_LiRPA 0.7.2, and dReal 4.21.6.2. Training is seeded, but GPU kernels are not bit-exact across hardware, so the repository ships the trained networks and their proof records rather than relying on retraining. The full branch-and-bound records total about 68 GB and are released separately from the code.

What is verified. For a quadratic anchor, a certificate consists of four statements: (1) the anchor sublevel set $C_0 = \{V \le c_V\}$ is positively invariant, lies in $\operatorname{int} \mathcal{D}$, and consists of convergent initial conditions; (2) the neural inner sublevel set lies in C_0, by equation (20); (3) $R_\theta \le -\epsilon_R$ on the annulus $\{x \in \mathcal{D} : c_{\text{in}} \le W_\theta(x) \le c_{\text{out}}\}$; and (4) every point with an active constraint satisfies equation (17). The negation of each statement is submitted to a complete branch-and-bound verifier: CROWN (Xu et al., 2021; Wang et al., 2021) for the neural statements, with an explicit absolute margin, and Z3 or dReal (Gao et al., 2013) for the polynomial or transcendental statement (1). A set is called verified only when every statement is proved on every box. Sampling is used only to propose levels and rank candidates; it never changes the formal status. The networked systems (S4–S6) are verified blockwise: local network gradients and coupling terms are bounded on low-dimensional graphs before the small-gain or comparison inequalities are assembled.

Volume estimates. Volumes measure already verified sets via sampling. For the common recipes, each volume is the mean over eight independent scrambled-Sobol replicates of 2^{17} points, and the reported uncertainty is the standard error across replicates; the same estimator ranks candidates during training, without any comparator information. The structured cases use case-specific sample sizes; S4 and S5 report the binomial standard error of a single Sobol sample.

Ablation. The plain network of Table 3 was verified at fixed levels in ascending order, each within the time used by the anchored proof; $W \leq 0.297$ is the largest level that verified. Since its empirical screen has volume $9{,}932$, the anchored gain of $1.42\times$ is at least $1.23\times$ for any plain level. The right panel of Figure 1 is evaluated from the frozen homogeneous-anchor network; its certified set covers 19.97% of the box, estimated via sampling.