Two-Stage Learning of **Stabilizing Neural Controllers** via Zubov Sampling and Iterative Domain Expansion

Haoyu Li, Xiangru Zhong, Bin Hu, Huan Zhang
(UIUC)

Paper: [https://arxiv.org/pdf/**2506.01356**](https://arxiv.org/pdf/2506.01356)
Code: [https://PaperCode.cc/**ContinousTimeLyapunov**](https://PaperCode.cc/ContinousTimeLyapunov)

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

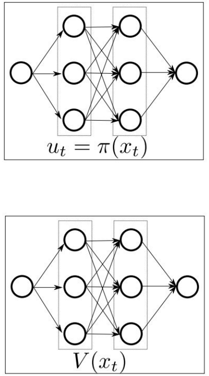
Synthesizing Stable Neural Network Controllers

Dynamical System (known)

$$\dot{x} = f(x, u(x))$$

Training

Synthesizing Neural Network Controller with Lyapunov function



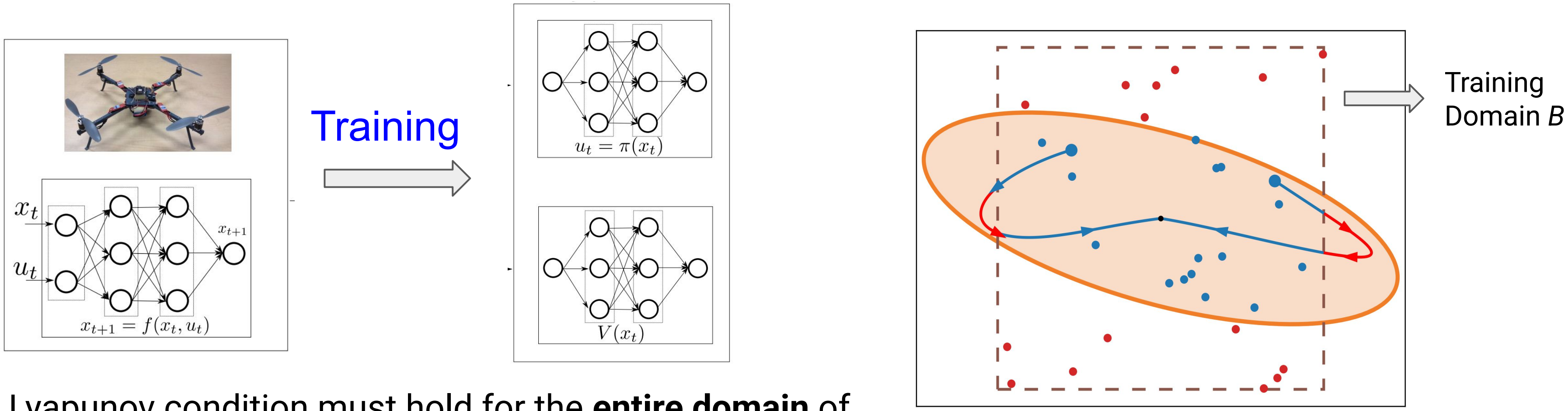
Verification

Formally Verify the Validity of the Lyapunov function

$$\nabla V(x) \cdot f(x, u(x)) < 0$$

Neural Lyapunov Control (Chang et al. 2019)
Lyapunov-stable neural-network control (Dai et al. 2021)
Neural lyapunov control for discrete-time systems (Wu et al. 2023)
Lyapunov-stable Neural Control for State and Output Feedback: A Novel Formulation (Yang et al. 2024)

Challenges in Learning a Stable NN Controller



1. Lyapunov condition must hold for the **entire domain** of interest
    - What training samples to use here? Different from typical ML with **finite** training samples

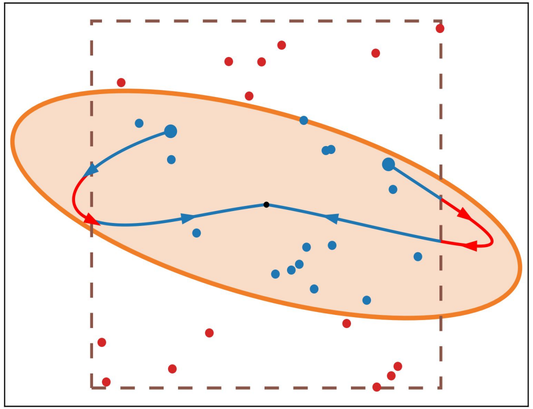
2. Hard to learn both Lyapunov function and controller for a **large** domain of interest
    - What domain of interest to use? We don't know the ROA for nonlinear system beforehand

Train by minimizing empirical violation of Lyapunov condition

$$L(\theta) = \mathbb{E}_{x \in \mathcal{B}}[\nabla V(x) \cdot f(x)]$$
$$\approx \frac{1}{n} \sum_{i=1}^{n} \nabla V(\boldsymbol{x}_i) \cdot f(\boldsymbol{x}_i)$$

Limitations in existing work

[Chang et al. 2019, Dai et al. 2021, Wu et al. 2023, Yang et al. 2024]



Limitations

1. Often requires **good initializations** (LQR/RL)
2. **Restricted** region of attraction (requires a fixed training domain of interests)
3. Hyperparameter tuning is very **difficult**; success rate not high on slightly complicated system

Root Cause

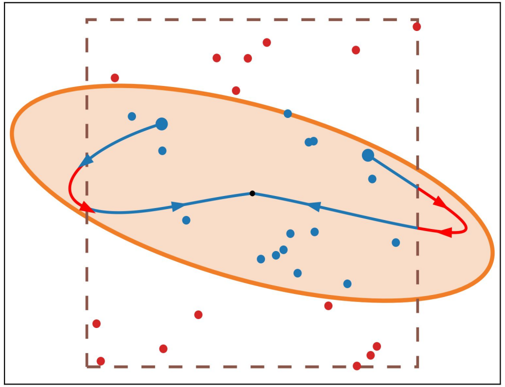
1. **Training data selection** What training samples to use to minimize Lyapunov violation?
2. **Training domain selection** What domain of interest to use to ensure easier and more stable learning?

Contributions (Training)

- Introduce a **two stage** training framework that is **non-conservative**, **stable** with **no need for special initializations**

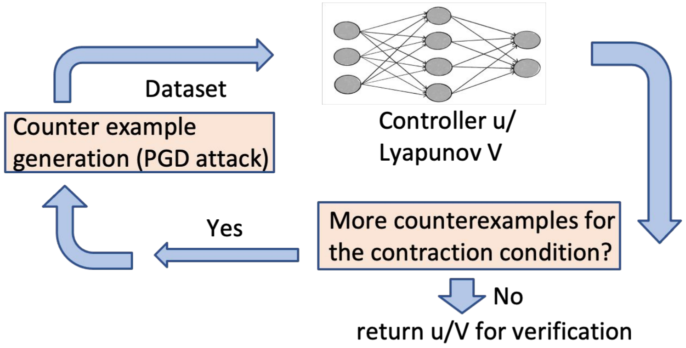
Contributions (Training)

<u>Stage 1</u>: ROA Estimation Stage



Once the controller can stabilize most trajectories

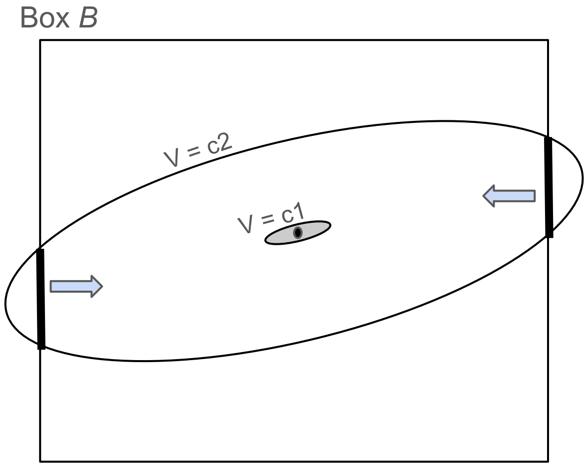
<u>Stage 2</u>: CEGIS Stage



Key Techniques

- **Zubov Guided Sampling**: Ensures balanced samples inside/outside of ROA
- **Dynamic Domain Expansion**: Expand training domain dynamically to include all convergent trajectories

Contributions (Verification)



Let 0 < c1 < c2, we verify

1. **<u>Lyapunov Condition</u>**

$$x \in \{x : c_1 \le V(x) \le c_2\} \cap B \implies \nabla V(x) \cdot f(x, u(x)) < 0$$

2. **<u>Boundary Condition</u>**

$$x \in \partial B \cap \{x : V(x) \le c_2\} \implies f(x, u(x)) \cdot \vec{n}(x) < 0$$

1. We extend α,β-CROWN with the capability of **automatically performing bound propagation through Jacobian**
2. **New linear relaxations for derivatives of many commonly used operators** (tanh(x), sigmoid(x)) to further tighten bounds
3. **Adjust c1, c2 during verification** using found counterexamples to avoid expensive bisesction

Verification comparisons:

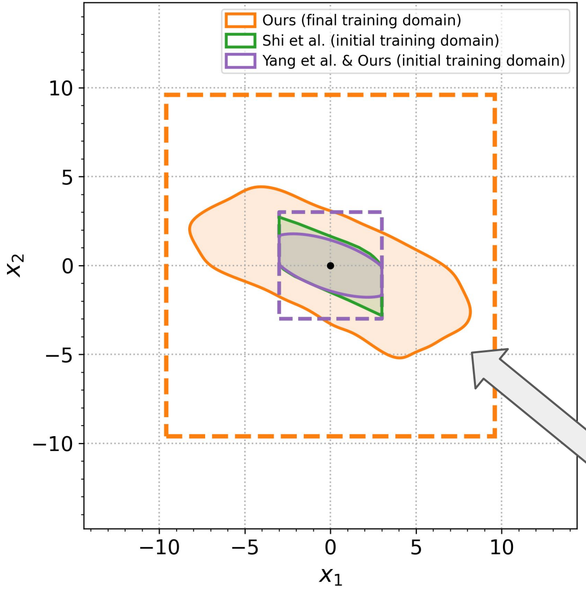
- α,β-CROWN is able to formally perform **formal stability verification** much more **efficiently** than the commonly used dReal verifier
- α,β-CROWN achieves a **40x to 10000x** acceleration compared to dReal

System	dReal	α,β-CROWN
Van der Pol	39265.21s	3.94s
Double Integrator	359.27s	3.00s
Pendulum (large torque)	1479.55s	3.64s
Pendulum (small torque)	1709.19s	3.94s
Path Tracking (large torque)	113.72s	3.77s
Path Tracking (small torque)	142.20s	3.67s

dReal: An SMT solver for nonlinear theories over the reals (CADE 2013)

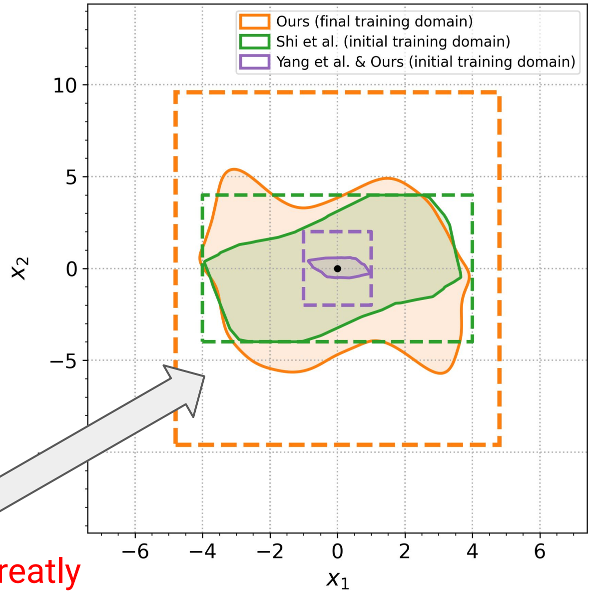
Numerical Results (2D):

Path Tracking



Success rates: ours 100%
Yang et al. 100%, Shi et al. 100%

Van Der Pol

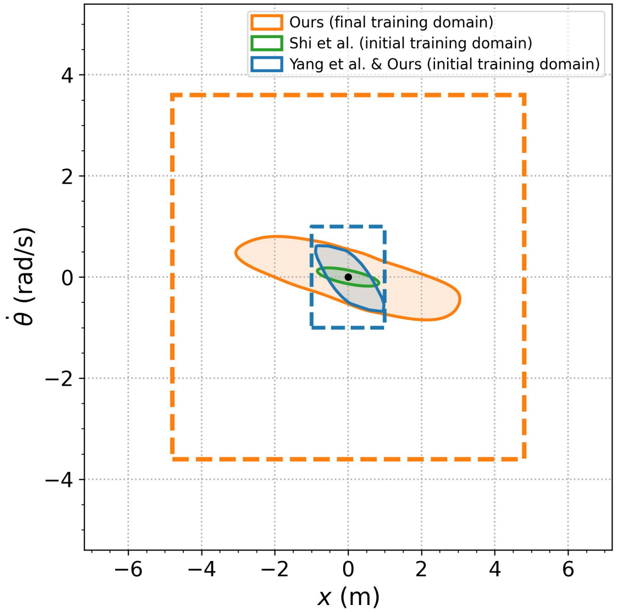


Our method greatly enlarged ROA, even with a small initial training domain

Success rates: ours 100%
Yang et al. 80%, Shi et al. 80%

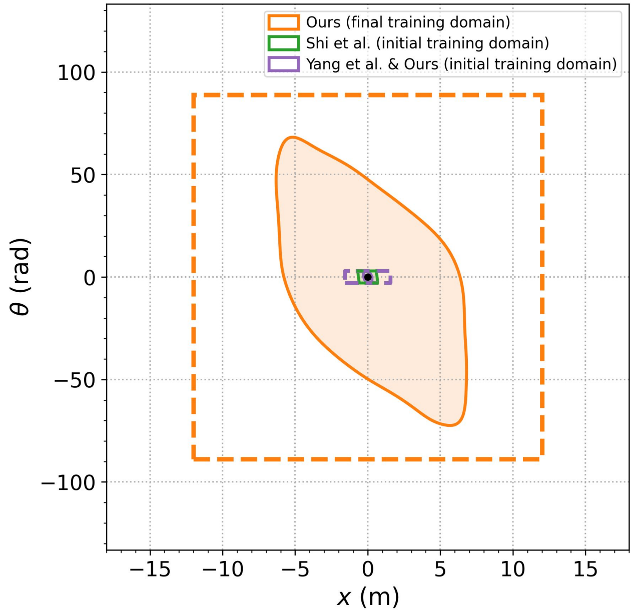
Numerical Results (Higher Dimension, slice):

Cartpole



Success rates: ours 100%
DITL 100%, Yang et al. 0%, Shi et al. 20%

2D Quadrotor



Success rates: ours 100%, DITL 0%,
Yang et al. 100%, Shi et al. 80%

ROA volume comparisons:

- Our method **robustly** obtains controller with **much larger certifiable ROA** compare to baselines (B means bigger torque limit, S means small torque limit)

System	DITL [36]		Yang et al. [40]		Shi et al. [26]		Ours	
	ROA	Succ	ROA	Succ	ROA	Succ	ROA	Succ
Van-der-Pol	–	–	1.36 ± 1.57	80%	20.2 ± 18.3	80%	$\mathbf{57.6 \pm 3.4}$	100%
Double Int	–	–	18.18 ± 16.19	60%	130.3 ± 3.9	60%	$\mathbf{302.5 \pm 10.7}$	100%
Pendulum B	61 ± 31	100%	70.6 ± 12.2	100%	487.5 ± 58.5	80%	$\mathbf{2946.5 \pm 149.1}$	100%
Pendulum S	–	–	217.34 ± 6.07	60%	306.3 ± 48.7	40%	$\mathbf{1169.2 \pm 124.5}$	100%
Path Tracking B	9 ± 3.5	100%	24.06 ± 0.29	100%	15.3 ± 8.9	60%	$\mathbf{122.0 \pm 3.7}$	100%
Path Tracking S	–	–	14.86 ± 0.18	100%	12.5 ± 6.5	100%	$\mathbf{73.8 \pm 12.5}$	100%
Cartpole	0.021 ± 0.012	100%	–	–	0.9266	20%	$\mathbf{306.1 \pm 54.2}$	100%
PVTOL*	–	–	–	–	49.87 ± 3.91	80%	$(\mathbf{1.91 \pm 0.23}) \cdot \mathbf{10^4}$	100%*
2D Quadrotor*	–	–	2.33 ± 0.47	100%	44.53 ± 18.38	80%	$(\mathbf{6.64 \pm 4.67}) \cdot \mathbf{10^6}$	100%*
Ducted Fan*	–	–	–	–	–	–	$(\mathbf{4.31 \pm 1.86}) \cdot \mathbf{10^4}$	100%*
3D Quadrotor**	–	–	–	–	–	–	$(\mathbf{1.17 \pm 0.64}) \cdot \mathbf{10^9}$	100%**

Neural lyapunov control for discrete-time systems (Neurips 2023)
Lyapunov-stable Neural Control for State and Output Feedback: A Novel Formulation (ICML 2024)
Certified Training with Branch-and-Bound: A Case Study on Lyapunov-stable Neural Control (arXiv 2025)