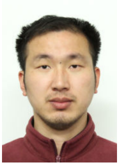
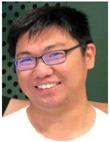
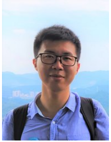
A Robust Learning Framework built on Contraction and Monotonicity

Ian R. Manchester,
with Ruigang Wang, Nic Barbara Johnny Cheng (ACFR)
Bowen Yi (Polytech Montreal)
Krishnamurthy (Dj) Dvijotham (ServiceNow/Google Deepmind)




THE UNIVERSITY OF SYDNEY
Australian Centre for Robotics

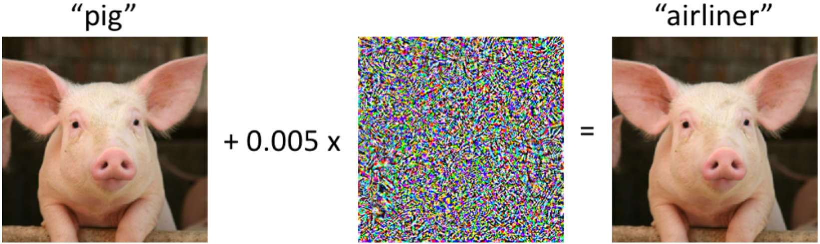
CDC Contraction Workshop, Dec 2024

Robust Neural Networks

Monotone, Bi-Lipschitz, and Polyak-Łojasiewicz Networks

Neural Lyapunov Functions, Stable Dynamics, and Contraction

Robust Neural Networks



Small input perturbation $x + \Delta x$

$\Downarrow$

Large output change $y + \Delta y$

Image: Aleksander Madry, MIT.

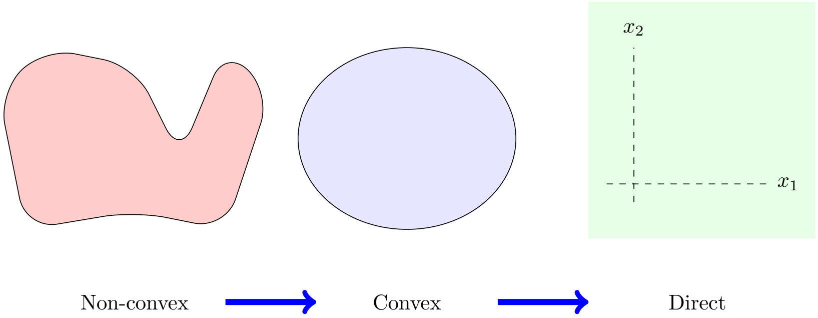
Adversarial Inputs and Lipschitz Bounds

- We want to avoid small input perturbations leading to large input perturbations
- If a model $f : x \mapsto y$ satisfies a **Lipschitz bound**:

$$\underbrace{\|f(x^a) - f(x^b)\|}_{\|\Delta y\|} \leq \gamma \underbrace{\|x^a - x^b\|}_{\|\Delta x\|} \quad \forall\ x^a, x^b$$

then the effect of adversarial perturbations is bounded.

Direct Parameterizations



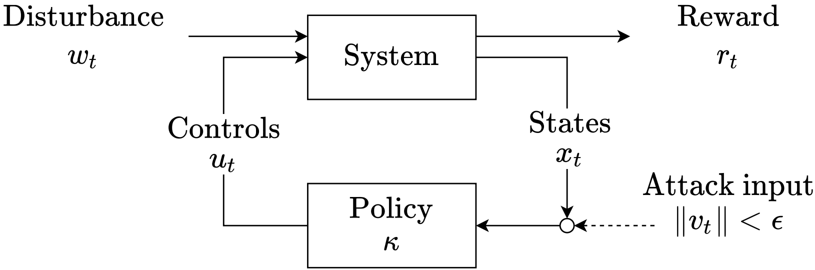
- How to impose Lipschitz bounds during training of large models?
- Our approach: construct **direct** parameterization of models satisfying this bound.
  - a.k.a. an intrinsic parameterization of the constraint manifold.
- Learn via unconstrained optimization: SGD, ADAM, etc.

Direct Parameterization

$$H = \begin{bmatrix} \gamma I & -W_0^\top \Lambda_0^\top & & & \\ -\Lambda_0 W_0 & 2\Lambda_0 & -W_1^\top \Lambda_1^\top & & \\ & \ddots & \ddots & \ddots & \\ & & -\Lambda_1 W_1 & 2\Lambda_{L-1} & -W_L^\top \Lambda_L^\top \\ & & & -\Lambda_L W_L & \gamma I \end{bmatrix} \succeq 0.$$

- Basic idea: square root representation: $H \succeq 0 \Leftrightarrow H = PP^\top$
- **Problem**: construct P s.t. H has the right sparsity structure:
- The blocks on the main diagonal $\gamma I, 2\Lambda_0, 2\Lambda_1, ...$ are **diagonal matrices**.

Robust Reinforcement Learning



- Lipschitz-bounded control policy limits affect of attacks/errors in state measurement.
- Parameterization of policies that guarantee closed-loop contraction (Youla-REN)

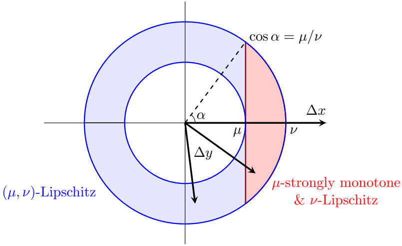
Pong: Uniform Random Noise

Unconstrained Lipschitz-Bounded

Code and videos can be found at
`https://github.com/nic-barbara/Lipschitz-RL-Atari`

Monotone, Bi-Lipschitz, and Polyak-Łojasiewicz Networks

Monotone and Bi-Lipschitz networks



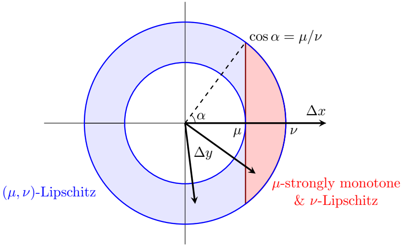
- A function $y = f(x), f : \mathbb{R}^n \to \mathbb{R}^n$ is **bi-Lipschitz** if

$$\mu\|\Delta x\| \leq \|\Delta y\| \leq \nu\|\Delta x\|, \quad \forall x_1, x_2 \in \mathbb{R}^n.$$

- We construct a direct parameterization (via incremental quadratic constraints) of strongly monotone and Lipschitz residual networks:

$$y_i = x_i + \mathcal{F}(x_i), \quad \langle \Delta y_i, \Delta x_i \rangle \geq \mu\|\Delta x_i\|^2,$$

Composition Properties



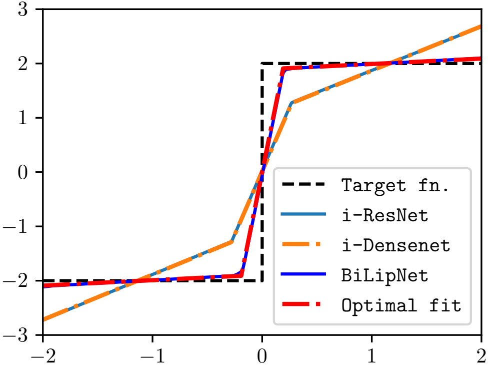
- Composition $f_1 \circ f_2(x)$ of monotone functions is *not* necessarily monotone.
- But composition two functions μ_1, μ_2-strongly monotone *is* $(\mu_1 \mu_2)$-inverse Lipschitz.
- Orthogonal layers $O(x) = Qx + b$, $Q^\top Q = I$: norm-preserving.
- Composition of orthogonal and strongly-monotone and Lipschitz layers:

$$f(x) = O_{K+1} \circ F_K \circ O_K \circ F_{K-1} \circ \ldots \circ O_2 \circ F_1 \circ O_1(x)$$

are **Bi-Lipschitz** with constants $(\prod_k \mu_k, \prod_k \nu_k)$

Toy Example

Fitting a step with (0.1, 10) - Bi-Lipschitz model



Model	inv. Lip.	Lip.	loss
i-ResNet	0.80	4.69	0.2090
i-DenseNet	0.82	4.66	0.2091
BiLipNet	**0.11**	**9.97**	**0.0685**
Optimal	0.10	10.0	0.0677

Learning Surrogate Cost Functions

- Given data $\{x_i, y_i\}, i = 1, ..., N$, with x_i vector and y_i scalar
- learn a model f

  $$y_i \approx f(x_i)$$

  i.e. standard supervised learning
- **But** with constraint that $f(x)$ is "easy to optimize", i.e.

  $$x^\star = \arg\min f(x)$$

  is can be efficiently and reliably computed.
- Why:
  - Data-driven optimization of black-box functions (MDO, experiment design, etc)
  - Learning terminal costs in MPC
  - Q learning with continuous action spaces.
  - Inverse reinforcement learning

Polyak-Łojasiewicz Networks

- A function $f : \mathbb{R}^n \to \mathbb{R}$ satisfies the *Polyak-Łojasiewicz (PL) condition*[1] if

$$\frac{1}{2}\|\nabla_x f(x)\|^2 \geq m(f(x) - \min_x f(x)), \forall x \in \mathbb{R}^n, \quad (1)$$

- **Guarantees** linear convergence of gradient descent to global minimum, less restrictively than convexity.
- If a function $g(x)$ is (μ, ν) bi-Lipschitz, then

$$f(x) = \frac{1}{2}\|g(x)\|^2 + c, \quad c \in \mathbb{R} \quad (2)$$

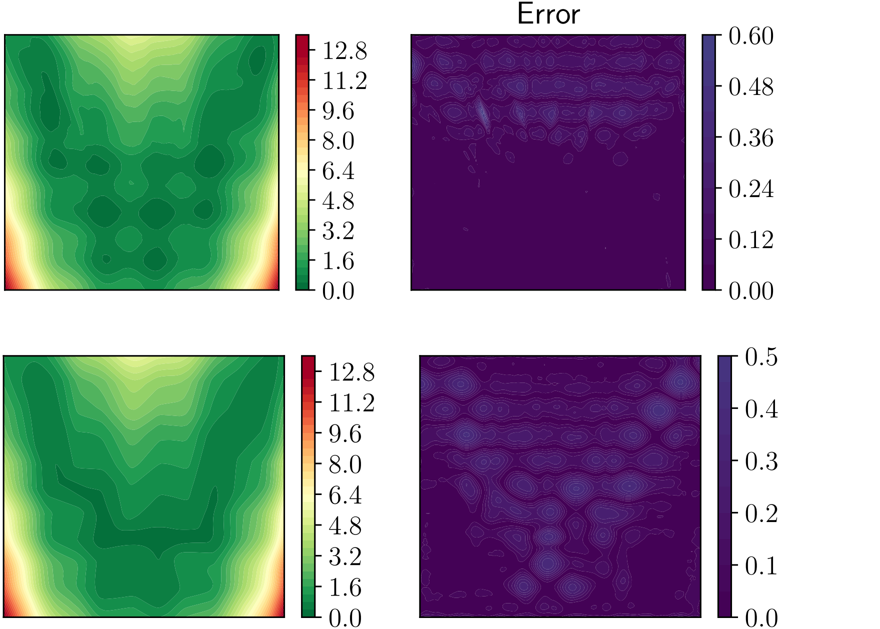
is a satisfies PL $m = \mu^2$. We call it a **PLNet**

- Unique minimum at the solution of $g(x) = 0$, i.e. $x^\star = g^{-1}(0)$, and a minimum value of c.

---

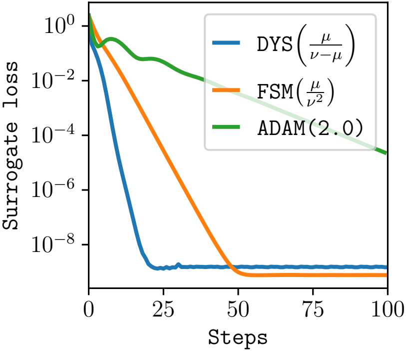
[1] Polyak, 1967; Łojasiewicz, 1967

Polyak-Łojasiewicz Networks: Rosenbrock function + Sine



Fast Solution of Minimum

- The minimum of $f(x) = |g(x)|^2 + c$ is at the point $x^\star : g(x^\star) = 0$, i.e. $x^\star = g^{-1}(0)$
- For deep networks g we can "backtrack" through layers.
- Solution via Davis-Yin 3-operator splitting
- Illustration on 20-dimensional Rosenbrock function:



Neural Lyapunov Functions, Stable Dynamics, and Contraction

Learning Lyapunov Functions

JOURNAL OF DIFFERENTIAL EQUATIONS 3, 323-329 (1967)

The Structure of the Level Surfaces
of a Lyapunov Function

F. WESLEY WILSON, JR.*

Department of Mathematics, University of Michigan, Ann Arbor, Michigan 48104

Received February 24, 1966

- Lyapunov functions can be written in the form:

$$V(x) = \frac{1}{2}\|g(x)\|^2$$

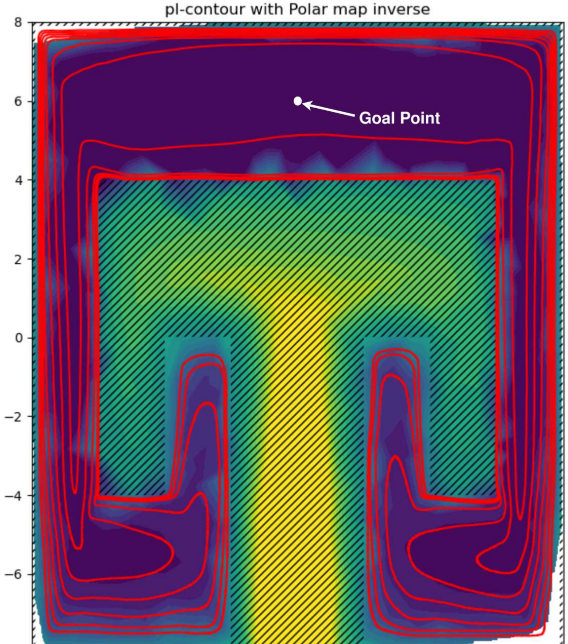
where g is a homeomorphism. Unique $x^\star : V(x^\star) = 0$.

- If g is *Bi-Lipschitz* we automatically get

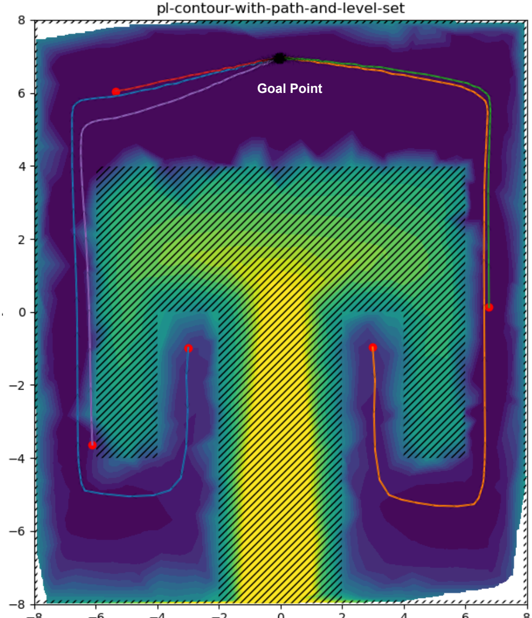
$$\mu|x - x^\star|^2 \leq V(x) \leq \nu|x - x^\star|^2$$

- If $x^\star$ is known, use $V(x) = \frac{1}{2}\|g(x) - g(x^\star)\|^2$

Flexible Lyapunov Functions

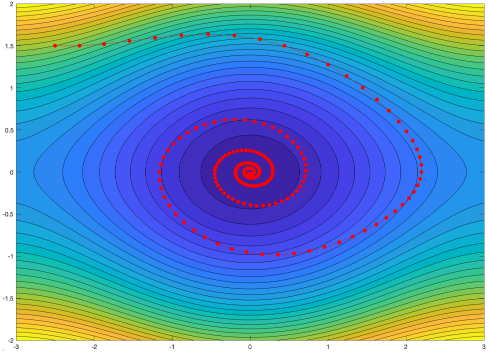


Flexible Lyapunov Functions



From Gradient Flow to Hamiltonians

Parameterize **descent directions**: $\dot{x} = (\underbrace{J(x)}_{J=-J^\top} - \underbrace{D(x)}_{\succ 0})\nabla V(x)$

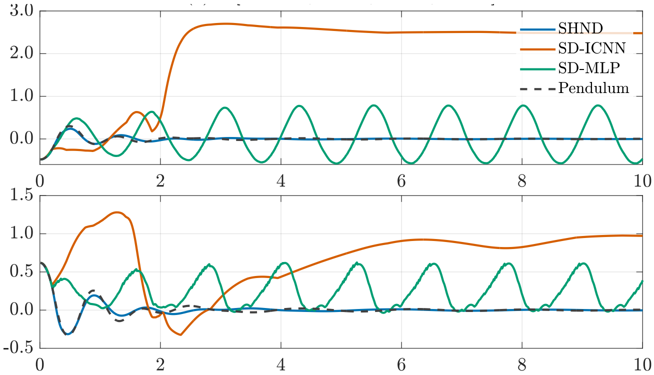


Extends to **passive and stable** port-Hamiltonian system:

$$\begin{aligned} \dot{x} &= (J(x) - R(x))\nabla V(x) + B(x)u \\ y &= B(x)^\top \nabla V(x) \end{aligned} \tag{3}$$

Double Pendulum

Learning dynamics from data $x(k), \dot{x}(k), k = 1, 2, ..., K$.



Comparison of ours to unconstrained method MLP and previous stability-preserving method ICNN.

Equivalence of Contraction and Koopman

Consider a system

$$\dot{x} = f(x) \tag{4}$$

and also consider changes of variables (Koopman embeddings) $\phi : \mathbb{R}^n \to \mathbb{R}^N$:

$$z = \phi(x) \quad \Longrightarrow \quad \dot{z} = Az \tag{5}$$

such that $\Phi(x) := \frac{\partial \phi}{\partial x}$ is full column-rank.

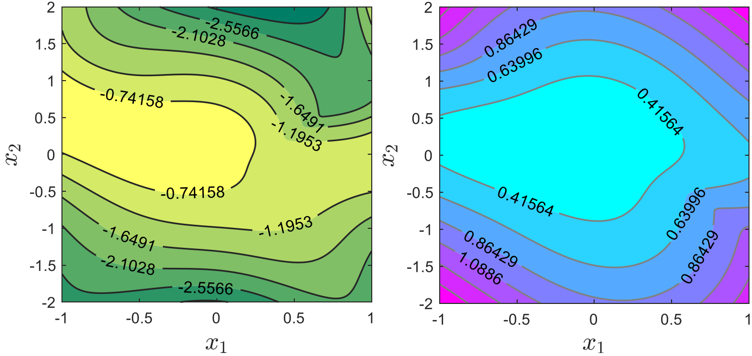
Theorem (informal):

- Suppose $\exists$ embedding (5) such that A is stable, then (4) is contracting with metric $M(x) = \Phi(x)^\top P \Phi(x)$ where $A^\top P + PA < 0$
- Conversely, suppose (4) is contracting, then there exists a full-rank embedding (5) such that A is stable.

In fact, ϕ can be strongly monotone $\mathbb{R}^n \to \mathbb{R}^n$

Learning a Contraction Metric from Trajectory Data

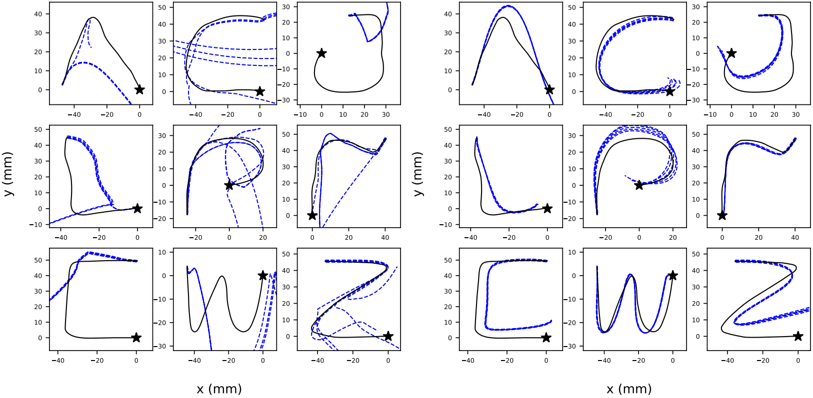
- Given trajectory data $x(k), \dot{x}(k), k = 1, 2, ..., N$
- Learn a mapping $z = \phi(x)$ such that $\dot{z} = Az$,
- Evaluate Lyapunov equation $A^\top P + PA = -I$
- Metric: $\Phi(x)^\top P \Phi(x)$.



Left: Max real parts of eigenvalue of $(\partial_f M + MF + F^\top M)$
Right: The smallest real parts of eigenvalue of M

Learning Robot Motion from Demonstration

- **Idea:** parameterize ϕ via biLipNet, parameterize A as stable.
- E.g. $A = -R^\top R + S - S^\top$.
- Learning motion of a robot arm from demonstration:



Summary

Main message:
We provide a rich parameterization of robustly invertible (bi-Lipschitz) neural networks.

Useful for:

- Learning "easily optimizable" surrogate losses (PLNet)
- Learning Lyapunov functions satisfying natural conditions
- Learning stable dynamics via Lyapunov descent directions
- Learning Koopman embeddings and contraction metrics

Thank you!

ThB12.4: Learning Stable and Passive Neural Differential Equations

- R. Wang & I. R. Manchester, **Direct Parameterization of Lipschitz Bounded Deep Networks**, ICML23
- N. Barbara, R. Wang, & I. R. Manchester, **On Robust Reinforcement Learning with Lipschitz-Bounded Policy Networks**, SysDO 2024.
- R. Wang, K. Dvijotham, & I. R. Manchester, **Monotone, Bi-Lipschitz and Poylak Lojasiewicz Networks**, ICML24
- J. Cheng, R. Wang, & I. R. Manchester, **Learning Stable and Passive Neural Differential Equations**, CDC24 .
- F. Fan, B. Yi, D. Rye, G. Shi, & I. R. Manchester, **Learning Stable Koopman Embeddings**, ACC2022, extended version (submitted): arXiv:2401.08153
- B. Yi & I. R. Manchester, **On the Equivalence of Contraction and Koopman Approaches for Nonlinear Stability and Control**, TAC 2023