Dissipativity-Informed Learning for Chaotic Dynamical Systems with Attractor Characterization

Sunbochen Tang
MIT LIDS & AeroAstro
Cambridge, MA 02139
tangsun@mit.edu

Themistoklis Sapsis
MIT MECHE
Cambridge, MA 02139
sapsis@mit.edu

Navid Azizan
MIT LIDS & MECHE
Cambridge, MA 02139
azizan@mit.edu

Abstract

Accurate prediction for chaotic systems is challenging due to their intrinsic sensitivity to initial condition perturbations. Instead, recent advances have been focused on forecasting models that produce trajectories preserving invariant statistics over a long horizon. However, data-driven methods are prone to generate unbounded trajectories, resulting in invalid statistics evaluation. Despite the sensitive nature of chaos, many chaotic systems show dissipative behaviors, meaning that they will enter a bounded invariant set eventually. In this paper, we propose a novel neural network architecture that preserves dissipativity by leveraging control-theoretic stability notions and constructing a projection layer that ensures trajectory boundedness. Additionally, the trained network also learns a Lyapunov function that governs dissipativity, along with an outer estimate of the attractor. We demonstrate the capability of our model in producing bounded long-horizon forecasts and characterizing the attractor using a truncated Kuramoto–Sivashinsky system.

1 Introduction

Chaos, characterized by exponential divergence after infinitesimal initial perturbations, is ubiquitous in a variety of complex dynamical systems including climate models [1] and turbulence in fluids [2, 3]. The unstable nature of chaotic systems makes it challenging to accurately predict their trajectories using data-driven methods. For specific models such as quadratic regression, theoretical limitations have been explored in [4], where the authors found these models could lead to finite time blowup or unstable statistical solutions, rendering the prediction unreliable. For popular time-series models, it was shown in [5] that training a recurrent neural network (RNN) for chaotic system prediction can lead to unbounded gradients.

Despite intrinsic stability, many chaotic systems are dissipative [6], i.e., their trajectories will converge to a bounded positively invariant set, which is also known as a strange attractor. For dissipative chaotic systems, although accurate long-term prediction is still challenging, recent progress has been made in applying deep learning to produce a dynamics emulator that preserves invariant statistics [7]. Note that in order to obtain meaningful statistics, it is often required to roll out the model prediction for a long horizon, where the model will traverse almost every state possible on the attractor. Without any formal boundedness guarantees, it has been observed that deep learning models are prone to the risk of stepping on a state not seen during training and experiencing finite-time blowup [8, 9], which reduces the length of reliable forecast. To address this issue, noise-inspired regularization has

Machine Learning and the Physical Sciences Workshop, NeurIPS 2024.

shown empirical success in promoting stability [10]. Another approach based on neural operator architectures is proposed in [11], which explicitly alters the flow of the learned dynamics emulator in a pre-specified region of the state space to ensure dissipativity.

In this paper, we propose a novel neural network architecture that ensures the learned dynamics emulator always produces a trajectory that converges to an invariant set characterized by learned parameters. We derive stability conditions that guarantee dissipativity leveraging control-theoretic perspectives, and enforce these conditions through an explicit projection layer in our model. In numerical simulations including Lorenz 63 and a set of truncated ordinary differential equations (ODEs) obtained from the Kuramoto–Sivashinsky equation, our model consistently produces bounded trajectories that converge to the strange attractor over long forecast horizons. Furthermore, the learned invariant set provides an overestimation of the strange attractor, which is notoriously difficult to characterize due to its complex geometry.

2 Problem Formulation

Consider a chaotic dynamical system described as a finite-dimensional ODE,

$$\dot{x}(t) = f(x(t)), \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is a n-dimensional vector which represents the state of the dynamical system at time t. Due to the chaotic nature of such systems, namely exponential divergence after an infinitesimal perturbation in the initial condition, accurate prediction is near impossible [12]. Instead, the objective is to construct a neural network dynamics emulator $\hat{f} : \mathbb{R}^n \to \mathbb{R}^n$, such that by solving the initial value problem

$$\dot{\hat{x}}(t) = \hat{f}(x(t)), \quad \hat{x}(0) = x(0) \tag{2}$$

from a given initial condition $x(0)$, the long-term solution $\hat{x}(t), t \in [0, T]$ approximates the true solution $x(t)$ well in terms of matching statistical properties. However, it has been observed in the literature [4] that emulators learned purely from trajectory data can generate unbounded trajectories from certain initial conditions, which makes statistical property evaluation infeasible.

In this paper, we aim to develop neural network-based models that are guaranteed to generate bounded solutions for the initial value problem (2). In Section 3, we formally introduce the definition of dissipativity and derive control-theoretic stability conditions for it. In Section 4, we propose a novel NN emulator model that incorporates such conditions and guarantees trajectory boundedness.

3 Algebraic Dissipativity Conditions and Attractor Estimation

Definition 1. *We say that the system* (1) *is* ***dissipative*** *if there exists a bounded and positively invariant set $M \subset \mathbb{R}^n$ such that $\lim_{t\to\infty} dist(x(t), M) = 0$, where $dist(x(t), M) = \inf_{y\in M} \|x(t) - y\|$. In other words, every trajectory of the system will converge to M asymptotically, and stays within M once it enters. M is said to be* ***globally asymptotically stable****.*

Note that in our definition, dissipativity requires both the existence of a positively invariant set and the asymptotic convergence of all trajectories to this set. In general, choices of such an invariant set may not be unique, and the smallest one is the strange attractor. However, characterizing the attractor for a known chaotic system is challenging due to its complicated geometry [13]. Instead of recovering the attractor empirically as attempted in [11], we focus on characterizing invariant sets that are level sets of an energy-like function. By leveraging control-theoretic stability notions, we derive conditions for positive invariance and asymptotic stability of a level set in Proposition 1 and 2, respectively.

Proposition 1 (invariant level set)**.** *For a dynamical system in* (1)*, suppose there is a continuously differentiable scalar-valued function $V : \mathbb{R}^n \to \mathbb{R}$ and a constant $c > 0$, such that*

$$\forall x \in \{x \in \mathbb{R}^n : V(x) > c\}, \dot{V}(x) \leq 0.$$

Then the level set $M(c) = \{x : V(x) \leq c\}$ is a positively invariant set for the system (1)*.*

Proposition 2 (asymptotic stability)**.** *For a dynamical system in* (1)*, suppose there is a lower-bounded continuously differentiable scalar-valued function $V : \mathbb{R}^n \to \mathbb{R}$ and a constant $c > 0$, such that*

(1) $\forall x \in \{x \in \mathbb{R}^n : V(x) > c\}, \dot{V}(x) < 0$; (2) V is radially unbounded.

Then the level set $M(c) = \{x : V(x) \leq c\}$ is globally asymptotically stable.

Theorem 1. *Suppose there is a lower-bounded radially unbounded C^1 function $V : \mathbb{R}^n \to \mathbb{R}$ and a constant $c > 0$ such that for the dynamical system in* (1)*,*

$$\forall x \in \mathbb{R}^n, \dot{V}(x(t)) + V(x) - c \leq 0.^1 \tag{3}$$

Then the system (1) *is dissipative and $M(c)$ is globally asymptotically stable. V is also known as a Lyapunov function.*

4 Methodology

In Section 3, we have derived stability conditions that ensure dissipativity for a system in (1). Now we propose our architecture that learns the Lyapunov function V and enforces the conditions in Theorem 1 through the construction of a projection layer. The overall structure is illustrated below:

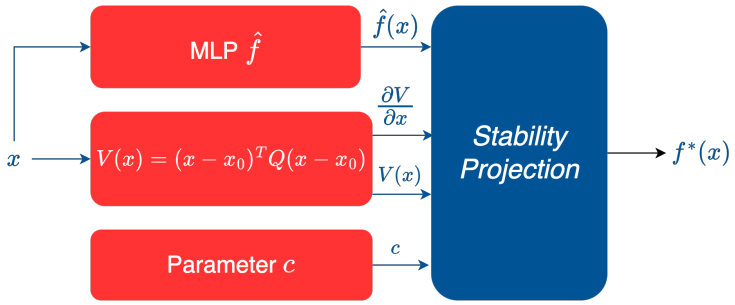


Figure 1: An illustration of the proposed neural network model.

Our proposed model consists of three learnable components illustrated in red: a multilayer perceptron (MLP) $\hat{f}$ that approximates the true nonlinear dynamics f on the right-hand side of (1); a quadratic function $V(x)$ that serves as a Lyapunov function with learnable parameters Q and x_0; and a level set parameter $c > 0$. Using these three components and the gradient information $\partial V/\partial x$, we construct a "stability projection" layer that outputs a predicted dynamics emulator $f^*(x) \in \mathbb{R}^n$ which guarantees dissipativity of its corresponding dynamics $\dot{x} = f^*(x)$.[2]

4.1 The Stability Projection Layer

As discussed in Section 3, given a Lyapunov function satisfying the conditions in Theorem 1, asymptotic convergence to the level set $M(c)$ can be established for a dynamical system if it satisfies the condition (3). Intuitively, this condition informs a subspace for the vector field $f(x)$ in which the forward dynamics will be dissipative. The stability projection layer is designed to project any dynamics emulator, in this case our MLP $\hat{f}(x)$, into such a subspace to ensure dissipativity.

More specifically, given an input $x \in \mathbb{R}^n$, the stability projection layer output $f^*(x)$ is chosen as the vector in the subspace of $\mathbb{R}^n$ confined by (3) that is closest to the emulator $\hat{f}(x)$ in l^2 distance, i.e.,

$$f^*(x) = \text{argmin}_{f(x)} \|f(x) - \hat{f}(x)\|^2 \quad \text{subject to} \quad \frac{\partial V}{\partial x}^T f(x) + V(x) - c \leq 0 \tag{4}$$

Since the above optimization problem has a quadratic loss and a linear constraint, similar to the approach in [14], an explicit solution can be found and computed using ReLU activation,

$$f^*(x) = \hat{f}(x) - \frac{\partial V}{\partial x}^T \frac{\text{ReLU}\left(\frac{\partial V}{\partial x}\hat{f}(x) + V(x) - c\right)}{\|\frac{\partial V}{\partial x}\|^2} \tag{5}$$

Note that the stability projection guarantees dissipativity and asymptotic convergence to any invariant level set $M(c)$ defined by $V(x)$ and $c > 0$. In what follows, we will discuss the training procedure

[1] See Appendix B.1 for remarks on the difference between the conditions in Proposition 1, 2 and Theorem 1.

[2] See Appendix B.2 for discussion on the choices of $V(x)$ parameterization and $\hat{f}$ architecture.

and how to search for a pair of V and c such that the level set $M(c)$ forms a tight outer estimation for the attractor.

4.2 Training Loss with Invariant Set Volume Regularization

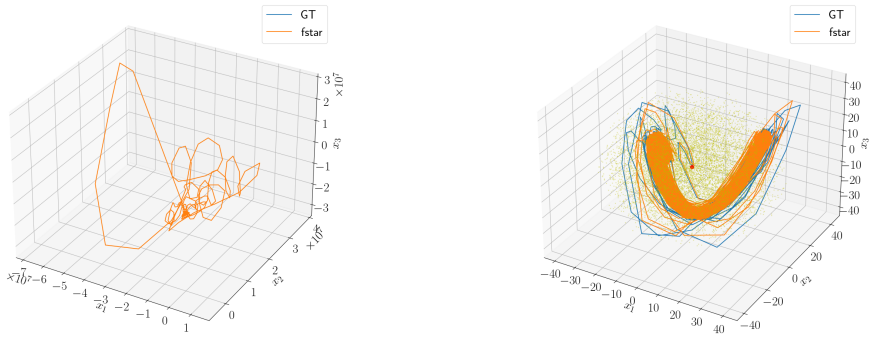
We assume the training dataset consists of trajectory data evenly sampled at h [sec] apart. Unlike [11, 7], we do not assume the trajectories in the training set are already inside the attractor, which allows for more flexibility when learning unknown chaotic systems and the transition period before it reaches an invariant statistics is unknown. During training, we consider a multi-step setting, where we roll out the predicted model for T steps, each step sampled at h [sec] using a numerical integration scheme. More specifically, given an initial condition chosen from the training dataset $x_0^{(i)}$, we forward simulate the ODE $\dot{\hat{x}}^{(i)} = f^*(\hat{x}^{(i)})$ with $\hat{x}^{(i)}(0) = x_0^{(i)}$ and obtain sampled states $\hat{x}_k^{(i)} = \hat{x}^{(i)}(kh)$ at $k = 1, 2, ..., T$. The trajectory prediction loss is defined as the MSE between the predicted sequence $(\hat{x}_k^{(i)})_{k=1}^T$ and the ground truth sequence $(x_k^{(i)})_{k=1}^T$.

As discussed earlier, the stability projection layer guarantees dissipativity along with convergence to an invariant level set $M(c)$, however, it does not inform how to choose an appropriate level set that characterizes the attractor. Since we are using a quadratic Lyapunov function $V(x)$, our goal is to find a pair of $V(x)$ and c such that the resulting ellipsoid $M(c)$ is a tight outer estimation of the actual strange attractor. Towards this objective, we use the volume of the ellipsoid $M(c)$ as the regularization loss to encourage learning as tight of an attractor outer estimation as possible while abiding by the dissipativity of the flow map. Combining the prediction and regularization loss, we have the following training loss with a weight hyperparameter $\lambda > 0$ for balancing the regularization terms:

$$\text{Loss} = \frac{1}{NT} \sum_{i=1}^{N} \sum_{k=1}^{T} \|x_k^{(i)} - \hat{x}_k^{(i)}\|_2^2 + \lambda \text{Vol}(M(c)), \quad \text{Vol}(M(c)) = \frac{\pi^{n/2}}{\Gamma\left(\frac{n}{2}+1\right)} \sqrt{\frac{c^n}{\det(Q)}} \tag{6}$$

5 Numerical Experiments

We demonstrate the effectiveness of our proposed approach through a long-horizon forecast example on an 8th-order truncated Kuramoto–Sivashinsky system of ODEs [15]. The training dataset consists of 4 different trajectories of 500 steps, sampled at 0.01 [sec] using the 4th-order Runge–Kutta (RK4) method. A single-step prediction setting is used during training, and both a vanilla MLP and our proposed method are trained on the same dataset.



(a) Trajectory grows unbounded using a vanilla MLP (no projection).

(b) Bounded trajectory generated by our model along with an ellipsoidal outer estimation for the attractor.

Figure 2: Vanilla MLP model generates unbounded trajectory while our model ensures stability and learns an attractor characterization. ("GT" is ground truth, and "fstar" refers to predicted trajectory)

During testing, we roll out both models using the RK4 method with a sampling rate of 0.01 [sec] over 10,000 steps. It is observed in Figure 2 that the vanilla MLP model experiences finite time blowup while our proposed model not only generates a bounded trajectory that converges to the

strange attractor but also successfully learns an invariant level set that encompasses the attractor. More numerical results can be found in Appendix C. We observe successful numerical results for the Lorenz 63 system as well, which are summarized in Appendix D.

References

[1] Edward N Lorenz. Deterministic nonperiodic flow. *Journal of atmospheric sciences*, 20(2): 130–141, 1963.

[2] Yoshiki Kuramoto. Diffusion-induced chaos in reaction systems. *Progress of Theoretical Physics Supplement*, 64:346–367, 1978.

[3] Gi Siv Ashinsky. Nonlinear analysis of hydrodynamic instability in laminar flames—i. derivation of basic equations. In *Dynamics of Curved Fronts*, pages 459–488. Elsevier, 1988.

[4] Andrew J Majda and Yuan Yuan. Fundamental limitations of ad hoc linear and quadratic multi-level regression models for physical systems. *Discrete & Continuous Dynamical Systems-Series B*, 17(4), 2012.

[5] Jonas Mikhaeil, Zahra Monfared, and Daniel Durstewitz. On the difficulty of learning chaotic dynamics with rnns. *Advances in Neural Information Processing Systems*, 35:11297–11312, 2022.

[6] Andrew Stuart and Anthony R Humphries. *Dynamical systems and numerical analysis*, volume 2. Cambridge University Press, 1998.

[7] Ruoxi Jiang, Peter Y Lu, Elena Orlova, and Rebecca Willett. Training neural operators to preserve invariant measures of chaotic attractors. *Advances in Neural Information Processing Systems*, 36, 2024.

[8] Zhixin Lu, Brian R Hunt, and Edward Ott. Attractor reconstruction by machine learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(6), 2018.

[9] Jaideep Pathak, Zhixin Lu, Brian R Hunt, Michelle Girvan, and Edward Ott. Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(12), 2017.

[10] Alexander Wikner, Brian R Hunt, Joseph Harvey, Michelle Girvan, and Edward Ott. Stabilizing machine learning prediction of dynamics: Noise and noise-inspired regularization. *arXiv preprint arXiv:2211.05262*, 2022.

[11] Zongyi Li, Miguel Liu-Schiaffini, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Learning chaotic dynamics in dissipative systems. *Advances in Neural Information Processing Systems*, 35:16768–16781, 2022.

[12] Steven H Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. CRC press, 2018.

[13] John Milnor. On the concept of attractor. *Communications in Mathematical Physics*, 99: 177–195, 1985.

[14] Youngjae Min, Spencer M Richards, and Navid Azizan. Data-driven control with inherent lyapunov stability. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 6032–6037. IEEE, 2023.

[15] Yiorgos S Smyrlis and Demetrios T Papageorgiou. Computational study of chaotic and ordered solutions of the kuramoto-sivashinsky equation. Technical report, 1996.

[16] Grigoriy Blekherman, Pablo A Parrilo, and Rekha R Thomas. *Semidefinite optimization and convex algebraic geometry*. SIAM, 2012.

A Proof for Theoretical Results

Proof for Proposition 1. Let $x(0) \in M(c)$, suppose there exists $t > 0$ such that $x(t_0) \notin M(c)$, which implies that $V(x(0)) \leq c < V(x(t_0))$. Since $V(x(t))$ is continuously differentiable in t, by the intermediate value theorem, we can find some $t \in [0, t_0)$ such that $V(x(t_0)) = c$, denote $S = \{t : V(x(t)) = c, t \in [0, t_0)\}$. Since $\forall t \in S, t < t_0$, $\sup S \leq t_0$. Suppose $\sup S = t_0$, then we can construct a sequence $(t_k)_{k\in\mathbb{N}}$ such that $t_k \to \sup S$ as $k \to \infty$. By continuity, $V(x(\sup S)) = c$ which contradicts $V(x(t_0)) > c$. Therefore, $\sup S < t_0$. Now by the mean value theorem, there exists $t_1 \in (\sup S, t_0)$ such that $V(x(t_1)) > c$ and $\dot{V}(x(t_1)) = (V(x(t_0)) - V(x(\sup S)))/(t_0 - \sup S) > 0$, which contradicts the assumed condition. Therefore, $\forall t > 0$, we have $V(x(t)) \in M(c)$ as long as $x(0) \in M(c)$, i.e., $M(c)$ is indeed a positively invariant set. □

Proof for Proposition 2. Since the condition here is stronger than the one stated in Proposition 1, $M(c)$ is a positively invariant set. Therefore, it suffices to consider a trajectory that starts outside $M(c)$. Suppose there exists a trajectory $x(t)$ such that $\forall t \in [0, \infty), V(x(t)) > c$, then $\dot{V}(x(t)) < 0$ and $V(x(t))$ is monotonically decreasing over time. Since $V(x(t))$ is lower bounded, $V(x(t)) \to a \geq \inf_x V(x)$ as $t \to \infty$. Suppose $a > c$, i.e., $\text{dist}(x(t), M(c)) = \inf_{y\in M(c)} \|y - x(t)\| \not\to 0$ as $t \to \infty$.

Since V is radially unbounded, for any $\alpha > 0$, we can find r_α such that $V(x) > \alpha$ for all $\|x\| > r_\alpha$. Therefore, any level set of V is bounded as $\{x : V(x) \leq \alpha\} \subset B(r_\alpha)$. Note that $V(x(t)) \in [a, V(x(0))]$ for all $t \in [0, \infty)$ because $V(x(t))$ is monotonically decreasing. Since $V(x)$ is continuous, the pre-image $S = \{x : V(x) \in [a, V(x(0))]$ is a closed set. Additionally, S is bounded because $S \subset \{x : V(x) \leq \alpha\} \subset B(r_{V(x(0))})$, which implies S is compact.

Since $\dot{V}(x)$ is continuous and $\dot{V}(x) < 0$ for all $x \in S$, there is $\gamma > 0$ such that $\max_{x\in S} \dot{V}(x) \leq -\gamma$, which implies $\max_{t\in[0,\infty)} \dot{V}(x(t)) \leq \max_{\|x\|<r_{x(0)}} \dot{V}(x) \leq -\gamma$. This contradicts the fact that $V(x(t)) \geq a > -\infty$ for all $t \in [0, \infty)$ since

$$V(x(t)) = V(x(0)) + \int_0^t \dot{V}(x(\tau))d\tau \leq V(x(0)) - \gamma t.$$

□

Proof for Theorem 1. The condition (3) implies that $\forall x \in \mathbb{R}^n$ such that $V(x) > c$, $\dot{V}(x(t)) \leq -(V(x) - c) < 0$. Therefore, by Proposition (1) and 2, the level set $M(c)$ is both globally asymptotically stable and positively invariant. □

B Remarks in Footnotes

B.1 Computational Tractable Conditions

In Proposition 1 and 2, by using the level set notion, we have greatly simplified the conditions of finding an invariant set that is also globally asymptotically stable. However, verifying the above conditions computationally is not trivial because it's generally difficult to enforce a condition over part of the state space. Inspired by s-procedure in sum-of-squares programming [16], at the cost of using a slightly stronger condition in our main theorem to enforce dissipativity, we obtain a much more tractable stability condition. Since condition (3) only relies on $V(x)$, $\partial V/\partial x$, $f(x)$ computation, it makes the explicit projection layer construction possible, which leads to dissipativity guarantee for our proposed model as discussed in Section 4.

B.2 Choices for Emulator Architecture and Lyapunov function

Note that we choose to parameterize the Lyapunov function V as a quadratic function for simplicity, but the framework can accommodate any function satisfying the requirements stated in Theorem 1. Additionally, the framework can be adapted to function approximators that are more expressive than MLPs. In our experiments with finite-dimensional ODEs, we found MLPs sufficient for producing reliable long-horizon forecast, which aligns with the findings in [11].

C Additional Numerical Results for the truncated KS ODEs

In Figure 3, it is observed that the predicted trajectory shares a similar energy level as the ground truth trajectory. Additionally, the level set value is a reasonably tight upper bound on the ground truth trajectory's energy level after it enters the strange attractor.

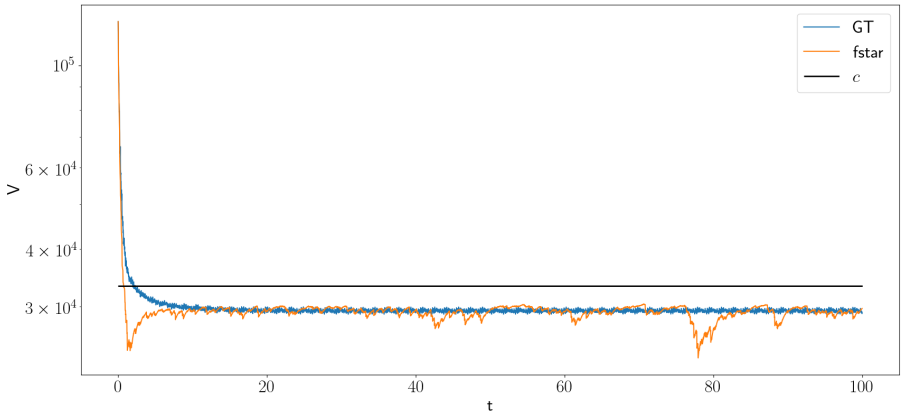


Figure 3: Lyapunov function $V(x(t))$ time history and level set value c.

In Figure 4, it is observed that the average energy spectrum of the predicted trajectory matches with the ground truth trajectory well. Despite the discrepancies between the energy spectra for dimensions 1 and 3, it is important to note that our method is not intended to improve the prediction of statistical properties. Unlike [7], our approach does not incorporate explicit incentives for matching statistical properties during the training process.

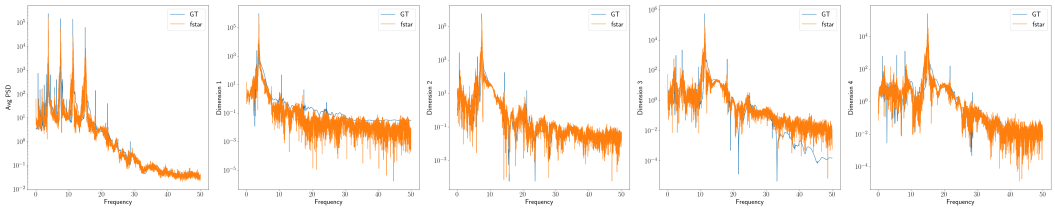


Figure 4: Energy spectrum amongst different dimensions: the first plot on the left is the average spectrum over all 8 dimensions; from the second to the fifth plot (from the left), we present the spectrum for the first four dimensions of the 8th-order system.

D Numerical Example: Learning Lorenz 63

In addition to the truncated KS ODEs example presented in Section 5, we now present numerical results on the Lorenz 63 system. Since it is a third-order ODE system, it is easier to visualize the trajectory in state space and its flow map.

In Figure 5, it is observed that our predicted trajectory matches with the ground-truth trajectory, and our method is able to empirically recover the strange attractor similar to [11]. In Figure 6, it is observed that our model learns a flow map that forces the trajectory to enter the invariant set and stays within that level set. Additionally, the flow further away from the ellipsoid center points more inward compared to the real system flow, which is likely caused by enforcing a stronger hence more conservative condition in (3). However, after the transient phase, we observe very similar flows within the invariant set.

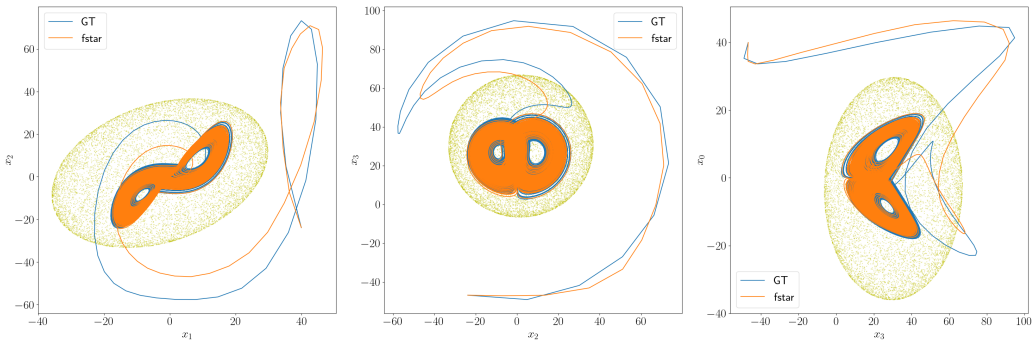


Figure 5: State trajectories from a randomized initial condition and the learned level set $M(c)$ from different views of the 3D space.

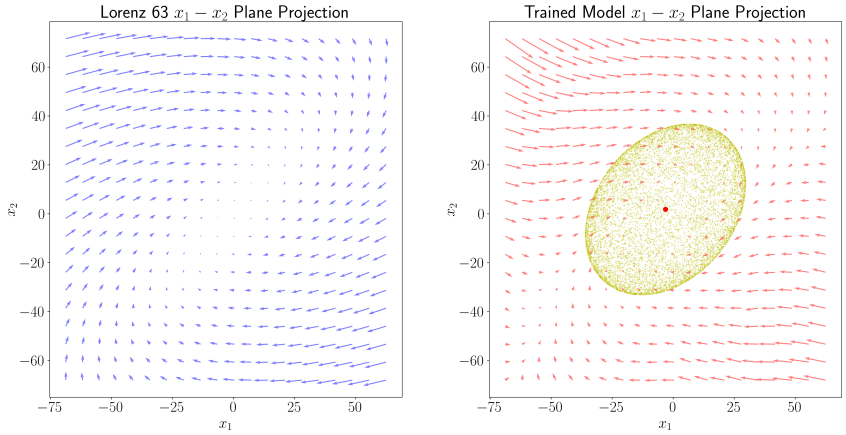


Figure 6: Comparison between the flow maps of the true system f and our emulator model f^*.

Similar to the truncated KS example, we also present the energy level time history and energy spectrum, plotted in Figure 7 and 8, respectively. The energy level of the predicted trajectory matches very well with that of the ground truth trajectory. We observe some discrepancies between the energy spectrums in Figure 8, similar to the discussion in the KS case, we will leave improving statistical invariance for future investigation.

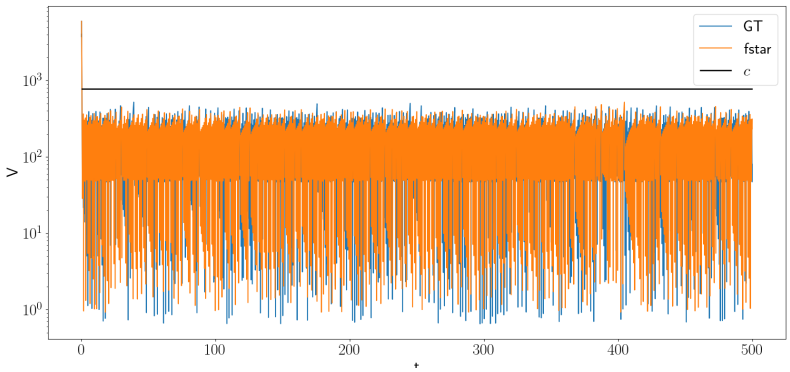


Figure 7: Lyapunov function $V(x(t))$ time history and level set value c

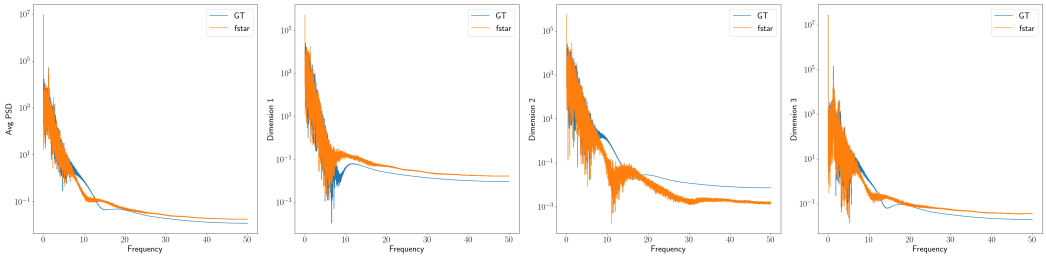


Figure 8: Energy spectrum amongst different dimensions: the first plot on the left is the average spectrum over all 3 dimensions; from the second to the fourth plot (from the left), we present spectrum for each of the three dimensions.