Learning Dissipative Chaotic Dynamics with Boundedness Guarantees

Sunbochen Tang[a,1], **Themistoklis Sapsis**[a], **and Navid Azizan**[a]

This manuscript was compiled on April 30, 2026

Chaotic dynamics, commonly seen in weather systems and fluid turbulence, are characterized by their sensitivity to initial conditions, which makes accurate prediction challenging. Recent approaches have focused on developing data-driven models that attempt to preserve invariant statistics over long horizons since many chaotic systems exhibit dissipative behaviors and ergodicity. Despite the recent progress in such models, they are still often prone to generating unbounded trajectories, leading to invalid statistics evaluation. To address this fundamental challenge, we introduce a modular framework that provides formal guarantees of trajectory boundedness for neural network chaotic dynamics models. Our core contribution is a dissipative projection layer that leverages control-theoretic principles to ensure the learned system is dissipative. Specifically, our framework simultaneously learns a dynamics emulator and an energy-like function, where the latter is used to construct an algebraic dissipative constraint within the projection layer. Furthermore, the learned invariant level set provides an outer estimate for the system's strange attractor, which is known to be difficult to characterize due to its complex geometry. We demonstrate our model's ability to produce bounded long-horizon forecasts that preserve invariant statistics for chaotic dynamical systems, including Lorenz 96 and a reduced-order model of the Kuramoto-Sivashinsky equation.

physics-constrained machine learning | dynamical systems | turbulence

Significance Statement

Dissipative chaos commonly occurs in scientific domains from fluid dynamics to climate systems. Despite the recent progress in data-driven models, they are often prone to generating unbounded trajectories, which prohibits meaningful predictions for long-term statistics. Here, we address this issue by introducing a framework that provides formal guarantees of trajectory boundedness for neural network models. By simultaneously learning a dynamics model and an energy representation from data, we devise a dissipative projection layer that enforces energy dissipation by construction rather than as an empirical heuristic. This work demonstrates how embedding general physics principles can build a new class of trustworthy and physically consistent AI models for scientific discovery.

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 exponential separation makes it challenging to accurately predict trajectories of chaotic systems. However, many chaotic systems of practical interest across various domains, including weather models and fluid dynamics (1, 2), turn out to be *dissipative*, meaning that their trajectories converge to a bounded and positively invariant set, often referred to as a strange attractor (4). Moreover, trajectories of dissipative chaos will visit almost every state on the attractor, resulting in ergodicity and invariant statistics (5). Consequently, rather than seeking pointwise-accurate predictions, the primary goal in modeling dissipative chaotic systems becomes capturing these invariant statistics over long forecast horizons.

Recent data-driven efforts have shown remarkable progress in building surrogate models that accelerate inference while preserving the long-term invariant statistics of dissipative chaos. These methods span a wide spectrum of structural assumptions and model complexity. On one end, structured nonlinear regression introduces physically motivated multi-level models to fit time series data efficiently (6, 7). At the other end, deep learning approaches rely on the representation power of neural networks to directly model complex chaotic behavior from raw data, while incorporating knowledge of shared physical system behaviors as specific architecture choices or regularization schemes (8–13). Hybrid approaches leverage autoencoder architectures to latent representation spaces where the dynamics evolve in simpler forms, with inspirations from Koopman theory (14), Dynamic mode decomposition (15), PCA (16), etc. Beyond one-step prediction, recurrent sequential models have also been explored to promote stability and improve forecast accuracy using more input information (17–19). In addition to standard recurrent models, a specific recurrent network architecture design for time series prediction, known as reservoir computing (RC), has demonstrated improved performance in reconstructing attractors in chaos and preserving invariant statistics (20–22).

To predict statistical properties on the attractor, data-driven models must generate arbitrarily long trajectories during inference to sufficiently sample the invariant measure. In practice, these models adopt an autoregressive paradigm that iteratively predicts the next state from its own prior outputs, making them vulnerable to drifting outside the region of training data. Since model performance

Author affiliations: [a]Massachusetts Institute of Technology, Cambridge, MA 02139

S.T., T.S., and N.A. designed research; S.T. performed research; S.T. analyzed data; and S.T. and N.A. wrote the paper.

The authors declare no competing interests.

[1]To whom correspondence should be addressed. E-mail: tangsun@mit.edu

DRAFT

is generally unpredictable outside of the training dataset, these models are prone to producing unbounded trajectories and invalid statistical forecasts. For models including quadratic regression and recurrent neural networks (RNNs), theoretical analysis has been established to reveal their fundamental difficulty in generating bounded trajectories for chaos forecasting (17, 23). For more complex models, including RCs and FNOs, empirical evidence of the same stability issue has also been reported (20, 24–26). As such, a core challenge remains, which is preventing data-driven models from generating unbounded trajectories in long-term prediction for chaotic dynamical systems.

To address this challenge, we introduce a neural network framework that establishes formal guarantees of trajectory boundedness by enforcing dissipative constraints on the learned model. To the best of our knowledge, this is the first work to achieve such provable guarantees in neural network models for chaotic dynamics. Our core contribution is a novel projection layer that enforces dissipativity by construction, consequently establishing trajectory boundedness guarantees. Rather than relying on system-specific domain knowledge, our approach learns both the underlying dynamics and an energy function that governs the dissipative behavior directly from data. Additionally, our method results in a learned energy function invariant set, which provides a tight outer approximation of the strange attractor. We first illustrate the key components in our framework using the Lorenz 63 system, then demonstrate its effectiveness in preserving invariant statistics and generating bounded trajectories for Lorenz 96 and a reduced-order model of the Kuramoto–Sivashinsky (KS) equation, highlighting its broad applicability across different chaotic benchmarks.

In the recent literature, there have been several solutions proposed to address the stability issue, which generally fall into two categories: the first focuses on empirical stabilization, and the second seeks formal guarantees at the cost of limited model expressivity. Within the first category, some techniques are architecture-specific, such as the use of noise-inspired regularization (27) or network pruning (28) to improve the long-term stability of RC models. Broader neural network-based approaches aim to preserve the system's invariant statistics by constraining models to match key dynamical properties like Lyapunov exponents (29) or by adding statistics-based regularization to the loss function (26, 30), or by combining regularization and post-training modifications to encourage dissipative behaviors (25). While these methods reduce instances of unbounded prediction trajectories, they often require system-specific prior knowledge and, importantly, lack formal guarantees for trajectory boundedness. The second category of methods seeks to establish such guarantees by directly incorporating physical principles; however, so far, it has only been achieved for specific model classes. For instance, for quadratic regression models, energy conservation can be enforced structurally through a specific mathematical construction (7). The explicit mathematical structure in such regression models makes it possible to embed energy conservation, but also limits their expressivity and prevents applications to more complex and realistic chaotic systems.

In this work, we design a general framework that aligns data-driven models with energy dissipation in chaos without reliance on explicit model structures, by formulating constraints based on learned energy representations. We simultaneously learn the flow of the system and an energy function directly from data, similar to prior work on learning stabilizing control inputs to unknown systems (31). By leveraging Lyapunov stability theory, we derive efficient algebraic conditions that characterize energy dissipation purely based on the learned flow and energy function, thus removing the need for system-specific information. The learned energy function and control-theoretic algebraic conditions together allow the design of a closed-form dissipative projection layer, which automatically enforces the learned flow to be dissipative and consequently guarantees trajectory boundedness.

Data-driven Dissipative Chaos Modeling

Consider a chaotic dynamical system described by the finite-dimensional ODE

$$\dot{x}(t) = f(x(t)), \qquad [1]$$

where $x(t) \in \mathbb{R}^n$ represents the state of the dynamical system at time t, and $f : \mathbb{R}^n \to \mathbb{R}^n$ is a nonlinear function that governs the dynamics. Many chaotic systems are *dissipative*, meaning that their trajectories eventually enter a bounded positively invariant set on which they exhibit chaotic behaviors. On this invariant set, also known as the strange attractor, the system becomes ergodic and consequently produces well-defined long-term statistical properties.

Definition 1. *We say that the system in Eq. 1 is* ***dissipative*** *if there exists a bounded and positively invariant set $M \subset \mathbb{R}^n$ such that*

$$\lim_{t\to\infty} dist(x(t), M) = 0, \quad dist(x(t), M) = \inf_{y\in M} \|x(t) - y\|.$$

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***.*

The practical significance of studying dissipative chaotic systems is the possibility to shift the focus from the challenging task of predicting exact chaotic trajectories to the more tractable goal of capturing their invariant statistical properties over long-term trajectory forecasts. In data-driven modeling for dissipative chaos, our objective becomes constructing a neural network dynamics emulator, $\hat{f} : \mathbb{R}^n \to \mathbb{R}^n$, that generates prediction trajectories $\{\hat{x}(t)\}_{t=t_0}^{t_0+T}$ over a future time horizon T by solving the initial value problem

$$\dot{\hat{x}}(t) = \hat{f}\big(\hat{x}(t)\big), \quad \hat{x}(t_0) = x(t_0), \qquad [2]$$

from which the invariant statistics of the true system can be reproduced. The premise of reproducing meaningful statistical properties is that the trajectory remains bounded for a sufficiently long forecast horizon T. Although empirical success has been reported in a variety of recently proposed autoregressive models (8, 20), these models are prone to finite-time blowup in long trajectory rollouts, as illustrated in Fig. 1, which prohibits their use for statistical evaluation. This happens because the training dataset covers only a subset of the strange attractor, and without explicit guarantees, errors accumulate when the model encounters states outside the

*The definition is derived based on the one introduced by (4), where they stated dissipativity as every trajectory will enter M eventually. Here, we quantify this behavior using the notion of asymptotic stability and trajectory distance to M.

DRAFT

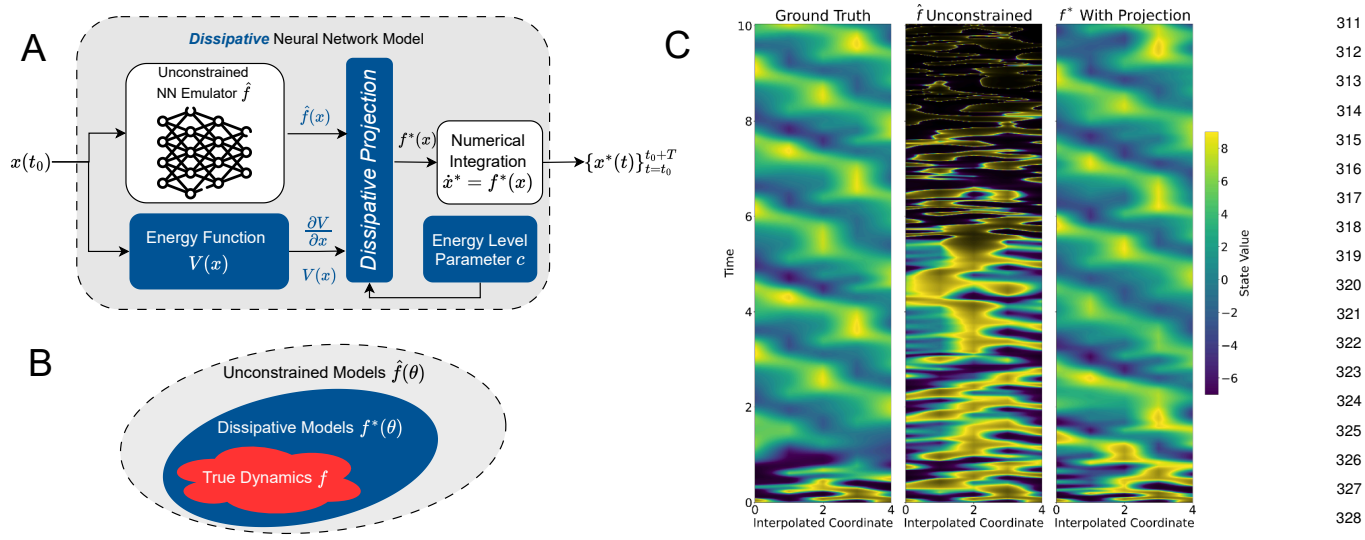


Fig. 1. Overview of our approach and main results. (A) Current autoregressive models without constraints suffer from accumulated error, leading to trajectories growing unbounded over extended rollout. Our approach overcomes this issue by building a dissipative projection layer that ensures the model is dissipative and guarantees bounded trajectories. (B) The dissipative constraint effectively narrows the search space of model parameters because the true dynamics of interest are dissipative. (C) Hovmöller diagrams of the Lorenz 96 system (with linear interpolation in the spatial dimension): Our approach (f^* with projection, on the right) generates a bounded trajectory that reproduces flow characteristics seen in the ground truth trajectory (on the left), while the unconstrained neural network generates an unbounded trajectory failing to capture any meaningful statistics.

training set, leading to divergence. Moreover, the exponential separation characteristic of chaotic systems embedded in the training data further amplifies these errors, contributing to divergence in model rollouts during test time.

To address this challenge, our approach introduces a *dissipative projection* layer to enforce an energy-based constraint into the neural network, which guarantees trajectories always remain bounded, ensuring their validity for long-term statistics evaluation. As illustrated in Fig. 1(A), our *dissipative NN model* jointly learns a dynamics emulator, an energy function, and an energy level parameter directly from data. This approach forces the learned dynamics model $\dot{x}^* = f^*(x)$ to be dissipative and also provides an outer estimate for the strange attractor as an energy level set $\{x : V(x) \leq c\}$. In numerical experiments for the Lorenz 96 system (32), while we observe that the unconstrained model in (8) generated unbounded trajectories during test time, our constrained model produces bounded trajectories that match the statistics patterns seen in the ground truth Hovmöller (33) diagram. This result exemplifies the importance of enforcing dissipativity for reliable long-term predictions in dissipative chaos modeling. Beyond ensuring trajectory boundedness, as illustrated in Fig. 1B, since the true dynamics are dissipative, by only constructing models that are certified to be dissipative, this constraint effectively narrows the search space for neural network parameters. This is especially helpful in the limited data regime, where it becomes difficult for the model to learn the dissipative behaviors by simply fitting labeled data under trajectory prediction settings.

Dissipative Dynamics: A Control-theory Perspective

To develop models with inherent dissipativity, we first need to understand theoretical conditions that make a dynamical system dissipative. Note that dissipativity describes a system's energy behavior over time, independent of whether the system is chaotic or not. In this section, we focus on deriving algebraic conditions that are computationally efficient through the connection between dissipativity and energy, which are crucial for our proposed architecture that guarantees dissipativity.

Energy-based Conditions for Dissipativity. Dissipativity, as the name suggests, has a close relationship with energy in a dynamical system. Intuitively, a dissipative system will lose energy over time, which corresponds to the trajectory converging to a bounded set. Although the trajectory convergence behavior is quantitatively stated in Definition 1, given a system $\dot{x} = f(x)$ known to be dissipative with access to the true dynamics f, it is still challenging to quantify the set M that the trajectory converges to. In the specific context of dissipative chaos, there has been a body of literature that tries to address this issue by studying invariant manifolds, volume contraction, and attempting to characterize the strange attractor (4). Despite the rigorous treatment and the progress over the years that help us understand the strange attractor, these characterizations are often stated in abstract mathematical concepts and a descriptive manner that is intractable to computationally verify, e.g., (34). Given our goal of enforcing dissipativity in neural network models, it is crucial to first derive computationally efficient conditions that ensure a system is dissipative.

In control theory, the concept of Lyapunov functions has been used extensively to formalize asymptotic stability of dynamical systems, which are also known as "energy-like" functions due to strong connections with the mechanical energy of the system. More importantly, by leveraging the level set of such functions, numerous computationally tractable conditions have been derived and extensively used in designing practical controllers to ensure a system's asymp-

DRAFT

totic stability to equilibrium points (35). By generalizing asymptotic stability with respect to an equilibrium point to a level set of a Lyapunov function, we derive computationally efficient conditions that ensure dissipativity in a dynamical system.

Recall in Definition 1, a dissipative system requires the existence of a bounded set M, which satisfies (1) M is an invariant set (2) the system is globally asymptotically stable towards M. By reducing the definition to the existence of a Lyapunov function V and choosing M to be a level set of V, i.e., $M(c) = \{x : V(x) \leq c\}$ where $c > 0$ corresponds to the energy level, we derive the following conditions for invariance and asymptotic stability of $M(c)$ in Proposition 1 and 2, respectively.

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

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

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

Proposition 2 (asymptotic stability)**.** *For a dynamical system in Eq. 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.

The proofs for both propositions are included in the SI Appendix. In Fig. 2, we sketch the intuitions behind the conditions in Proposition 1 and 2.

Algebraic Conditions for Dissipativity and Attractor Outer Estimation. Compared to Definition 1, the conditions derived in the propositions above are much more quantitative. However, verifying these conditions for a given system in Eq. 1 computationally is not trivial; for example, the negative semi-definite condition for $\dot{V}(x)$ is defined only on a certain part of the state space, outside the level set $M(c)$ in Proposition 1. Inspired by the S-procedure in sum-of-squares programming (36), we can replace these conditions with a single algebraic condition that is independent of the state, which is more computationally tractable.

Theorem 1. *Suppose there exists 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 Eq. 1,*

$$\forall x \in \mathbb{R}^n, \dot{V}(x) + V(x) - c \leq 0. \qquad [3]$$

Then the system Eq. 1 is dissipative and $M(c)$ is globally asymptotically stable.

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

[†] Here $\frac{\partial V}{\partial x}$ refers to the row vector $[\frac{\partial V}{\partial x_1}, \ldots, \frac{\partial V}{\partial x_n}]$.

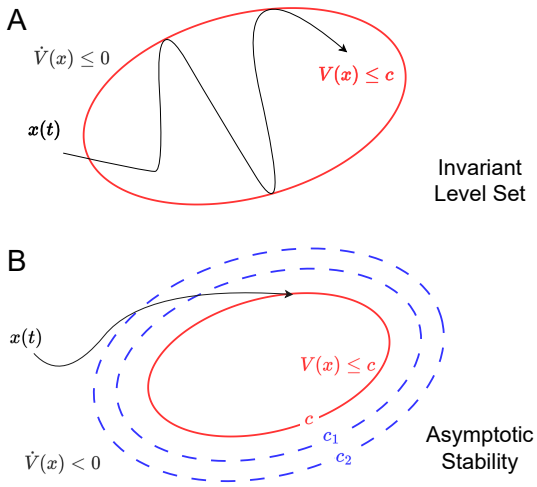


Fig. 2. Illustrations of theoretical results in Proposition 1 and 2: (A) The level set boundary serves as a barrier since the trajectory cannot gain energy outside. Once entering $M(c)$, the trajectory will be confined within. (B) The trajectory loses energy over time outside $M(c)$ because $\dot{V}(x) \leq 0$, resulting in convergence to the level set $M(c)$ eventually ($c_2 > c_1 > c > 0$).

Strange Attractor Outer Estimation. Recall the characteristics of the strange attractor discussed in the background section; if only the first two properties are satisfied by a set S, then this set is known as "the attractor" (37). The difference from the strange attractor is that the attractor does not require the exponential separation of neighboring trajectories everywhere, hence it is also a superset of the strange attractor. Note that if a set $S \subset M(c)$ is an attractor, then $M(c)$ is both invariant and globally asymptotically stable. Therefore, in addition to certifying a system to be dissipative, the level set $M(c)$ also provides an outer approximation of the original attractor, which is consequently an outer estimate of the strange attractor as well.

In practice, characterizing the strange attractor is challenging due to its complex geometry (34). In addition to empirically reproducing the strange attractor using the learned model similar to (25, 30), our method also provides a level set outer estimate for the strange attractor by learning the Lyapunov function V and level set parameter c.

Importance of the Algebraic Condition. The condition in Eq. 3 is strictly stronger than the conditions in Proposition 1 and 2. Although the condition is slightly more conservative, the computational tractability of obtaining a state-independent algebraic condition is well worth the trade-off. More importantly, this algebraic condition is crucial for constructing our proposed model architecture that ensures the learned system is dissipative, which will be discussed in detail in the next section.

Inherently Dissipative Neural Network Dynamics Model

Now we propose our architecture that simultaneously learns a dynamics emulator and a Lyapunov function V, with the former guaranteed to be dissipative by enforcing the condition

DRAFT

in Eq. 3 through the construction of a projection layer. The overall structure is illustrated in Fig. 1(A).

Our proposed model consists of three learnable components: a neural network dynamics emulator $\hat{f}$ that approximates the true nonlinear dynamics f on the right-hand side of Eq. 1; a quadratic energy-like function $V(x) = (x - x_0)^T Q(x - x_0)$ serving as the Lyapunov function with learnable parameters Q (Q is parameterized to be positive definite) and x_0 that define the shape and center of the level sets, respectively; and a level set parameter $c > 0$. Using these three components and the gradient information $\partial V/\partial x$, we construct a "dissipative projection" layer that outputs a predicted dynamics emulator $f^*(x) \in \mathbb{R}^n$ which guarantees dissipativity of its corresponding dynamics $\dot{x} = f^*(x)$.

A key feature of our method is its modularity, which provides the flexibility to enforce stability as an "add-on" scheme. The proposed "dissipative projection" layer is agnostic to the architectural choice for the dynamics emulator, which allows the integration of any suitable function approximator as the backbone model $\hat{f}$. Similarly, while we use a quadratic form for the Lyapunov function V for simplicity, our framework accommodates any function that satisfies the conditions outlined in Theorem 1. This flexibility establishes our approach as a general-purpose pipeline for enforcing stability, rather than a solution dependent on a specific class of models.

Dissipative Projection Ensures Boundedness. Following Theorem 1, if $\dot{x} = f^*(x)$ satisfies the condition in Eq. 3, then the system is guaranteed to be dissipative and converge to the level set $M(c)$ defined by the learned Lyapunov function $V(x)$ and the learned constant $c > 0$. The dissipative projection layer is designed to modify the backbone emulator $\hat{f}$ such that the output f^* produces a learned dynamical model $\dot{x} = f^*(x)$ that satisfies the condition in Eq. 3, therefore certified to be dissipative.

Intuitively, this condition informs a subspace for the vector field $f(x)$ in which the forward dynamics will be dissipative. The dissipative projection layer is designed to project any unconstrained dynamics approximator, $\hat{f}(x)$, into such a subspace to ensure dissipativity.

More specifically, given an input $x \in \mathbb{R}^n$, the dissipative projection layer output $f^*(x)$ is chosen as the vector closest to the approximator $\hat{f}(x)$ under l^2 distance in the subspace of $\mathbb{R}^n$ defined by Eq. 3, i.e., $f^*(x)$ is the solution to the following optimization problem:

$$f^*(x) = \text{argmin}_{f(x)} \|f(x) - \hat{f}(x)\|_2^2 \quad [4a]$$

$$\text{subject to} \quad \frac{\partial V}{\partial x} f(x) + V(x) - c \leq 0 \quad [4b]$$

As discussed extensively earlier, the constraint of the optimization problem, which is adopted from the condition in Eq. 3, ensures the learned dynamics emulator $f^*(x)$ to be dissipative while being computationally efficient. In addition to the fact that the condition is easily verifiable through basic arithmetic operations, the constraint is also linear in the optimization variable $f(x)$. Since the above optimization problem has a quadratic loss and a linear constraint, an explicit solution can be found and computed using ReLU activation, similar to the approach in (31, 38):

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

With the dissipative projection layer implemented as in Eq. 5, we state the following corollary that formalizes the dissipativity of our proposed model architecture, which is a direct result of Theorem 1. A computational proof verifying $f^*(x)$ satisfies Eq. 3 $\forall x \in \mathbb{R}^n$ is included as well.

Corollary 1. *The learned dynamics model $\dot{x} = f^*(x)$ is a dissipative system with a bounded and positively invariant level set $M(c) = \{x : V(x) \leq c\}$. The set $M(c)$ is globally asymptotically stable, which implies every trajectory of the system is bounded and converges to $M(c)$ asymptotically.*

Proof. If $\frac{\partial V}{\partial x}\hat{f}(x) + V(x) - c \leq 0$, it follows from Eq. 5 that $f^*(x) = \hat{f}(x)$. Therefore, in this case, $\frac{\partial V}{\partial x} f^*(x) + V(x) - c \leq 0$, meaning that $f^*(x)$ satisfies the condition in Eq. 3.

If $\frac{\partial V}{\partial x}\hat{f}(x) + V(x) - c > 0$, then we have

$$\begin{aligned}
&\frac{\partial V}{\partial x} f^*(x) + V(x) - c \\
&= \frac{\partial V}{\partial x}\left[\hat{f}(x) - \frac{\partial V}{\partial x}^T \frac{\text{ReLU}\left(\frac{\partial V}{\partial x}\hat{f}(x) + V(x) - c\right)}{\|\frac{\partial V}{\partial x}\|^2}\right] + V(x) - c \\
&= \frac{\partial V}{\partial x}\hat{f}(x) - \left(\frac{\partial V}{\partial x}\hat{f}(x) + V(x) - c\right) + V(x) - c = 0,
\end{aligned}$$

which ensures that $f^*(x)$ satisfies the stability condition Eq. 3. Therefore, by Theorem 1, the system $\dot{x} = f^*(x)$ is dissipative and $M(c)$ is globally asymptotic stable.

Additionally, since $\forall x \notin M(c)$, $\dot{V}(x) < -(V(x) - c) < 0$, the trajectory always loses energy outside $M(c)$. Also note that if the trajectory starts within $M(c)$, it can never leave, i.e., $V(x(t)) \leq c$ for all $t \geq 0$. Therefore, for any $t \in [0, \infty)$, $V(x(t)) = V(x(0)) + \int_0^t \dot{V}(x(\tau))d\tau \leq \max\{V(x(0)), c\}$. Since every level set of $V(x(t))$ is bounded (see proof for Proposition 2 in SI Appendix), $x(t)$ is always bounded. □

Training with Invariant Set Volume Regularization. We consider a training dataset consisting of trajectory points which are evenly sampled at h [sec] from a few ground truth trajectories initialized at randomly sampled initial conditions. Unlike (25, 30), 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, where the transition period before it reaches an invariant statistics is unknown. This is beneficial for our proposed model to learn where to place the invariant set $M(c)$ and apply dissipative projection.

During training, we consider a multi-step setting, where we roll out the learned 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 learned system $\dot{\hat{x}}^{(i)} = f^*(\hat{x}^{(i)})$ with $\hat{x}^{(i)}(0) = x_0^{(i)}$ and obtain sampled states at the same sampling period h [sec], $\hat{x}_k^{(i)} = \hat{x}^{(i)}(kh)$ at $k = 1, 2, ..., T$. By sampling N such trajectory snapshots of length T from the training dataset, we define the prediction loss as the MSE between the predicted rollout sequence $(\hat{x}_k^{(i)})_{k=1}^T$ and

DRAFT

the ground truth sequence $(x_k^{(i)})_{k=1}^T$: Prediction Loss $= \frac{1}{NT}\sum_{i=1}^N \sum_{k=1}^T \|x_k^{(i)} - \hat{x}_k^{(i)}\|_2^2$.

In the dissipative projection layer, the quadratic Lyapunov function $V(x)$ and the level set parameter c both need to be optimized during training. Although the prediction loss depends on V and c through the projection operator that produces $f^*(x)$, optimizing only the prediction loss may not be a well-defined optimization problem. More specifically, if we have found a level set $M(c_1)$ that is globally asymptotically stable and invariant, then any superset $M(c_2)$ for $c_2 > c_1 > 0$ is also globally asymptotically stable and invariant. Therefore, there could potentially be infinitely many solutions that lead to the small prediction loss.

To address this issue, we introduce a regularization loss that encourages the learned level set $M(c)$ to be as small as possible, which aligns with our goal of characterizing a tight outer estimate of the strange attractor. Toward this objective, we use the volume of the ellipsoid $M(c)$ as the regularization loss. Combining the prediction and regularization loss, we have the following training loss with a weight hyperparameter $\lambda > 0$ for balancing the regularization terms:

$$\text{Loss} = \frac{1}{NT}\sum_{i=1}^{N}\sum_{k=1}^{T} \|x_k^{(i)} - \hat{x}_k^{(i)}\|_2^2 + \lambda \text{Vol}(M(c)), \qquad [6a]$$

$$\text{Vol}(M(c)) = \frac{\pi^{n/2}}{\Gamma\left(\frac{n}{2}+1\right)}\sqrt{\frac{c^n}{\det(Q)}} \qquad [6b]$$

Results

Unlike most data-driven methods that prioritize data over structures, our approach aligns the ML model with physics principles by learning both the dynamics and an energy representation and enforcing an energy dissipation constraint. We illustrate our method and demonstrate the effectiveness of providing formal trajectory boundedness guarantees in reliably reproducing invariant statistics through a set of numerical experiments based on Lorenz 63, Lorenz 96, and Kuramoto–Sivashinsky (KS) equation, with an increasing level of complexity. For more quantitative evaluations regarding invariant statistics and more visualizations, please refer to the SI Appendix.

Learned flow is dissipative towards the invariant energy level set. We first consider the classic Lorenz 63 system, originally proposed in (1) as a simplified model for atmospheric convection. The low-dimensionality of the system makes it convenient for visualizing both the strange attractor and the system's dissipative behavior. As shown in Fig. 3(A), a 50,000-step long-term trajectory rollout generated by our model ("fstar") accurately reproduces the characteristic "butterfly-shaped" strange attractor of the ground truth ("GT"). The learned invariant energy level set, visualized as a yellow ellipsoid with a red dot indicating the learned center x_0, provides a tight outer-estimate for the strange attractor, and the trajectory quickly enters the level set and stays in afterwards.

Figure 3(B) further illustrates the learned model's dissipativity by comparing its projected flow on the $x_1 - x_2$ plane with that of the ground truth. The learned vector field matches the true flow near the attractor and, crucially, consistently points inward outside the learned invariant set, as expected by our model construction. In Fig. 3(C), we evaluate the learned energy function V along trajectories of both the true system and our proposed model f^*. In both cases, $V(x(t))$ quickly enters and remains within the learned invariant set $\{x : V(x) \le c\}$ on a comparable time scale, and the range of energy values under f^* closely matches that of the true system while staying tightly bounded by the level set. This confirms that the learned V acts as an effective Lyapunov-like function guiding dissipative behavior. Overall, the Lorenz–63 example demonstrates that our method can learn geometrically interpretable and stable models that generate bounded trajectories while faithfully reproducing invariant statistics.

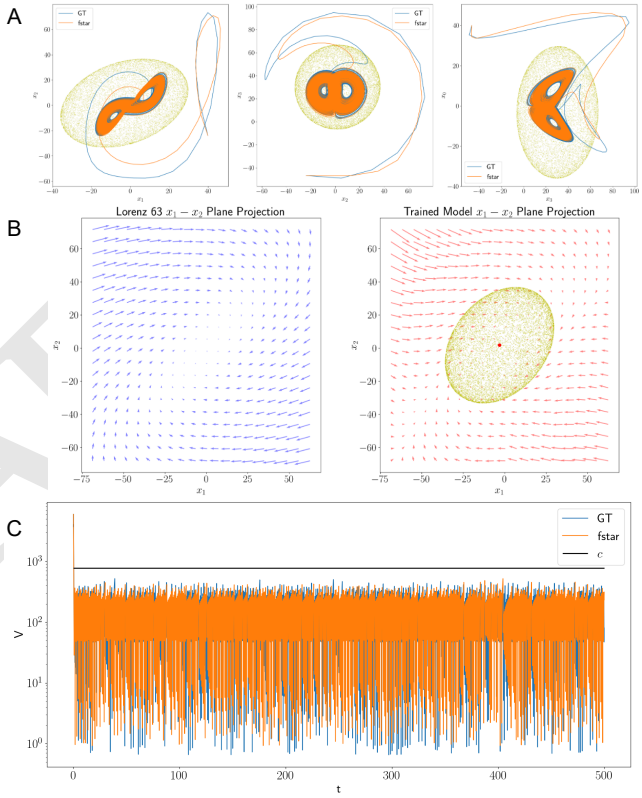


Fig. 3. Lorenz 63 (A) Trajectories generated by the learned model ("fstar") and true dynamics ("GT") are visualized by their 2D projections, along with the learned invariant set (sampled with yellow points). (B) Comparison of the flow map projected onto $x_1 - x_2$ plane. (C) Comparison of the learned energy function time history evaluated on the ground truth trajectory and the trajectory generated by our learned model.

Boundedness guarantee prevents finite-time blowup in high-dimensional systems. To demonstrate the necessity of guaranteed boundedness in data-driven methods, we consider two higher-dimensional chaotic systems: the 5-dimensional Lorenz 96 system and a 32-dimensional reduced-order model of the KS equation (KS-ROM). For these more complex systems, unconstrained machine learning models experience finite-time blow-ups, resulting in prediction trajectories invalid for statistics evaluations.

As illustrated in Fig. 1(A), we benchmark against an unconstrained machine learning baseline which uses a standard multilayer perceptron (MLP) architecture to approximate the dynamics f by $\hat{f}$. In the finite-dimensional case, the

DRAFT

MLP baseline is equivalent to the Neural Operator (39) and the DeepONet (11), both are state-of-the-art data-driven methods for general PDE modeling that do not incorporate any stability or energy constraints. For fair comparisons, we use the same MLP backbone for our method, augmenting it with an energy representation and the dissipative projection. For both Lorenz 96 and the reduced-order model for KS (KS-ROM), we train the unconstrained MLP and our proposed model on the exact same datasets, such that any difference in performance would be a direct consequence of the stability constraints. During testing, we sample a random initial condition, and forward simulate both the unconstrained model, $\dot{x} = \hat{f}(x)$, and our model with dissipative projection $\dot{x} = f^*(x)$ in an autoregressive manner (illustrated in Fig. 1(A)), and compare the trajectory rollouts generated by these two models. Since these system dimensions exceed three, we use Principal Component Analysis (PCA) to project the trajectory rollouts onto their first two principal components, such that a straightforward 2D visualization of the higher-dimensional trajectories can be presented.

The testing trajectories generated by the unconstrained model $\hat{f}$ completely deviate from the attractor and then exhibit finite–time blow-up for both Lorenz 96 and KS-ROM systems, as illustrated in Fig 4(A, C) respectively, by scatter plots comparing the first two PCA components of the generated trajectories with their corresponding ground truth trajectories. In contrast, our proposed model with dissipative projection guarantees trajectory boundedness, producing prediction trajectories that recover the shape of the strange attractor for both systems, as shown in Fig 4(A) and (C). Moreover, by applying the same PCA to sampled points on the boundary of the invariant level set $V(x) \leq c$, Fig. 4(A, C) not only verify the trajectory quickly converges to our learned level set (illustrated as the red point cloud), but also show that the level set form a tight outer-estimate of the strange attractor.

Enforcing boundedness helps preserve invariant statistics. A key goal in modeling dissipative chaos is to reproduce the invariant statistics of the system's strange attractor, a task that requires generating bounded trajectories. When training data is limited, unconstrained models often fail by producing divergent trajectories, rendering statistical evaluation meaningless. Our method overcomes this by ensuring trajectory boundedness without prior knowledge of the system's statistics, unlike other approaches that incorporate known measures into the loss function (26, 30). We achieve this by constraining trajectories to a learned level set that acts as an outer estimate of the attractor, guiding the dynamics to the relevant region of the state space. To demonstrate the efficacy of this constraint, we compare rollouts from our model against an unconstrained baseline using: (1) spatiotemporal plots to illustrate flow patterns and (2) histograms of the leading principal components to show improved statistical distributions. Further visualizations, including Fourier spectra, Fourier modes, and learned energy characteristics, are provided in the SI Appendix.

The spatiotemporal plots in Fig. 1(C) and Fig. 5 validate that our model generates trajectories that recover system characteristics while the unconstrained baseline fails and diverges unboundedly. For the Lorenz 96 system, the Hovmöller plot (33) shows that the unconstrained model's solution diverges to unbounded values. Our model, however, maintains a stable rollout that correctly reproduces the westward-propagating wave patterns seen in the ground truth. Similarly, for the reduced-order model of the Kuramoto-Sivashinsky equation, the solution reconstructed from 32 Fourier modes shows a finite-time blowup for the unconstrained model. In contrast, our model's trajectory remains bounded and successfully captures the complex emergence of cellular chaos and coarsening events characteristic of the true dynamics.

To quantitatively evaluate the preservation of invariant statistics, we project the long-horizon testing trajectory rollouts onto their first two principal components (PCs) and compute the probability density function (PDF) of the projected points. These leading PCs capture dominant modes of variance and provide a low-dimensional visualization of the geometric structure of the attractor. The 2D histograms in Fig. 4(B, D) compare the PDFs of the ground truth (left), unconstrained model ($\hat{f}$, middle), and our proposed model (f^*, right). In the region near the attractor, the unconstrained model's trajectory distribution is scattered and unstructured, failing to capture the distinct characteristics of the ground truth attractor. In contrast, our model not only confines the trajectory to the correct region of the subspace but also reproduces the intricate shape of the invariant measure.

We provide a quantitative comparison of the statistical property prediction errors between the unconstrained model $\hat{f}$ and our constrained model f^* for both Lorenz 96 and KS-ROM systems. First, the Kullback-Leibler (KL) divergence between the predicted and the true PCA distributions further validates the visual assessment in Fig. 4(B, D): our model's divergence (Lorenz 96: 0.4078; KS-ROM: 0.8355) is significantly lower than that of the unconstrained model (Lorenz 96: 9.7856, KS-ROM: 3.0557) which fails to capture the attractor's geometric characteristics. Second, we evaluated the relative error of the Fourier energy spectra acquired from the predicted trajectory rollout compared to ground truth. Here again, the unconstrained model exhibits orders of magnitude higher error due to finite-time blowup (Lorenz 96: 164.10; KS-ROM: 244.47). In stark contrast, our proposed model's spectrum matches the ground truth closely (Lorenz 96: 0.0213; KS-ROM: 0.0001). These metrics demonstrate that our method effectively preserves the invariant statistics of chaotic systems by ensuring trajectory boundedness through dissipative constraints, even when trained on limited data and without prior knowledge of the system's statistical properties.

Discussion

In this paper, we have proposed a general framework that provides formal guarantees of trajectory boundedness in data-driven chaotic dynamics models, which directly addresses a crucial stability issue in reliable applications of these models to long-horizon forecasts. To achieve such guarantees without heavy reliance on system-specific domain knowledge, we derived efficient algebraic conditions characterizing dissipative dynamics based on analyzing an energy function, by leveraging stability results in control theory. We designed a dissipative projection layer to enforce such dissipative algebraic conditions by construction in our model, which learns the dynamics and the energy function concurrently from data. Additionally, we incorporated an invariant set

DRAFT

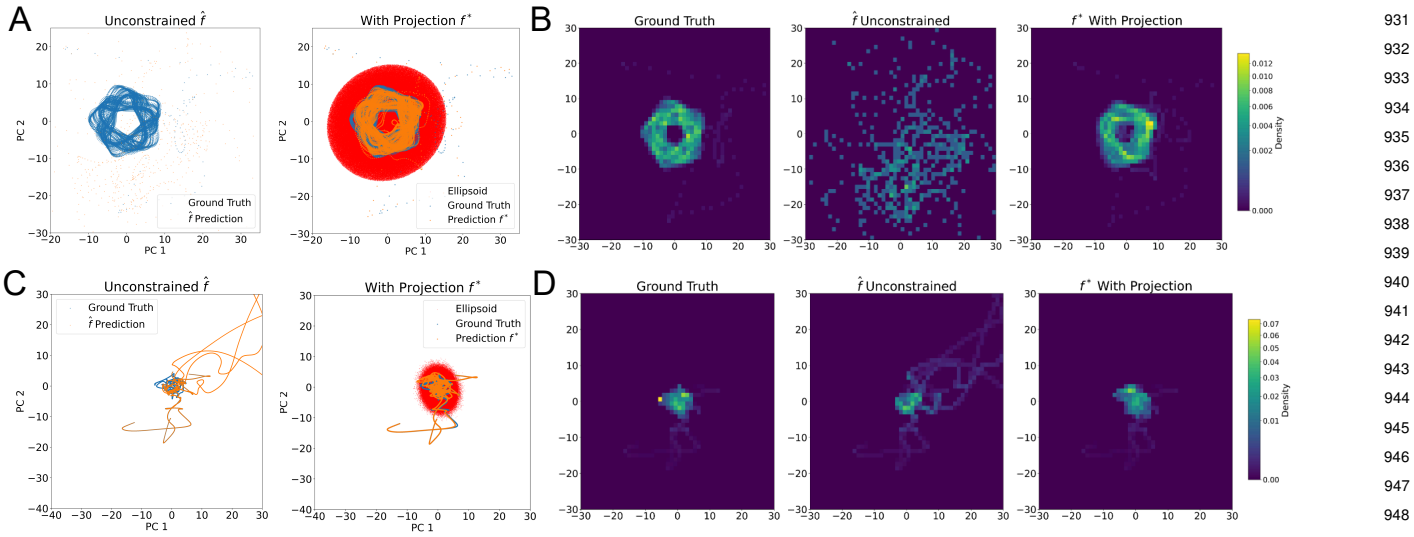


Fig. 4. Principal component analysis (PCA) comparison between trajectories generated by unconstrained model ($\hat{f}$) and our proposed model with dissipative projection (f^*) for Lorenz 96 (first row, figures A and B) and a reduced-order KS model (second row, figures C and D). Unconstrained model (left figure in (A) and (C)) generates a trajectory that quickly deviates from the attractor and then grows unbounded. Our proposed model (right figure in (A) and (C)) provides boundedness guarantees, which guides the generated trajectory to enter and traverse the strange attractor. In addition, the learned invariant set (red ellipsoid point cloud) provides a tight outer-estimate of the attractor. The 2D histograms represent the probability density of trajectories in the PCA leading components for Lorenz 96 (B) and the reduced-order KS model (D). In both cases, the unconstrained model $\hat{f}$ (middle figure in (B) and (D)) produces an unstructured distribution that scatters across the PCA space, while our proposed model f^* (right figure in (B) and (D)) is able to reproduce the shape and density of the ground truth distribution (left figure in (B) and (D)), which validates its capability to better preserve the system's invariant statistics.

volume-based regularization, which enables the model to self-discover the energy constraint and an invariant set that serves as a tight outer approximation for the strange attractor. As such, our framework ensures that the learned neural network model is provably dissipative, thus providing trajectory boundedness guarantees. Numerical experiments, including on Lorenz 96 and a reduced-order KS model, demonstrated that our model generates bounded trajectories and, in turn, reconstructs invariant statistics that closely match those of the true dynamics.

Notably, we also observed that an unconstrained model without the projection layer, trained on the exact same dataset, generated unbounded trajectories. This ablation revealed both the effectiveness and the necessity of establishing formal guarantees in machine learning models. For physical systems where collecting data can be expensive, a low-data regime, where an unconstrained model cannot learn stable behaviors purely from the given dataset, can become common. Our proposed projection layer suggests that the model quality can be significantly improved by adding the correct inductive bias devised from physical principles, without the need for any additional training data.

In summary, we advance data-driven modeling of chaotic dynamical systems by establishing formal guarantees of trajectory boundedness. By embedding hard constraints as an inductive bias, our models are reliable by construction, consistent with known physical principles, without requiring additional data or compute. This capability is especially valuable in data-scarce or safety-critical scientific domains, such as climate modeling, optimal design of fluid systems, and autonomous systems. More broadly, our framework for constraining neural-network outputs mitigates the black-box nature of ML models and their unpredictable behavior at deployment. Looking ahead, we plan to extend these guarantees to modeling more complex real-world systems and their downstream control tasks. We expect this line of work to inform model design and enable more trustworthy applications in complex cyber-physical systems.

Materials and Methods

Numerical Models of Chaotic Systems. We used three chaotic systems of increasing complexity for our numerical experiments. Ground truth data for all systems was generated by integrating the governing equations using the 4th-order Runge-Kutta (RK4) method with a time step of $h = 0.01$ seconds.

Lorenz 63 System. The Lorenz 63 system (1) is a 3-dimensional ordinary differential equation (ODE) given by:

$$\begin{aligned}\dot{x}_1 &= \sigma(x_2 - x_1),\\ \dot{x}_2 &= x_1(\rho - x_3) - x_2,\\ \dot{x}_3 &= x_1 x_2 - \beta x_3,\end{aligned}$$

where the state is $x \in \mathbb{R}^3$. We used the standard chaotic parameters $\sigma = 10.0$, $\beta = 8/3$, and $\rho = 28.0$.

Lorenz 96 System. The Lorenz 96 system (32) is a higher-dimensional ODE. We used a 5-dimensional version described as follows, with the forcing constant set to $F = 8.0$ to generate chaotic behavior.

$$\dot{x}_i = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F, \quad i = 1, \ldots, 5,$$
where $x_{-1} = x_4, x_0 = x_5, x_6 = x_1$.

Kuramoto–Sivashinsky (KS) Reduced-Order Model (ROM). We used a 32-dimensional ODE model derived from the Kuramoto–Sivashinsky (KS) partial differential equation (2). The reduction was performed using the Galerkin projection method (40, 41). Full details on the derivation are provided in the SI Appendix.

Model Architecture and Training.

DRAFT

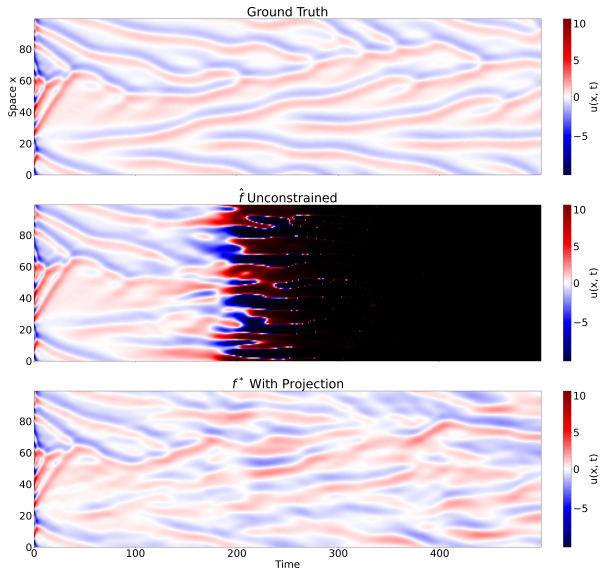


Fig. 5. Reproduction of complex emergent dynamics in the Kuramoto-Sivashinsky system. The figure compares the spatiotemporal evolution of the ground truth (top) with predictions from the unconstrained ($\hat{f}$, middle) and our constrained (f^*, bottom) models. The unconstrained model exhibits finite-time blowup and grows unbounded. In contrast, our model's trajectory not only remains bounded but also faithfully captures the essential physical phenomena of the system, including the intricate patterns of coarsening events seen in the ground truth.

Model Architecture. Our proposed model, f^*, and the unconstrained baseline model, $\hat{f}$, share an identical multilayer perceptron (MLP) backbone to ensure a fair comparison.

The MLP backbone $\hat{f}$ consists of two hidden layers and uses a GeLU (42) activation function for all three systems. For the Lorenz 63 and 96 systems, each hidden layer contains 64 neurons. For the reduced-order KS (KS-ROM) system, the hidden layer width increases to 512 neurons as the dynamics the model tries to capture become much more complex.

Our model, f^*, augments this backbone with an additional network to represent the energy function $V(x)$ and a final projection layer that enforces the energy dissipation constraint. The energy function $V(x)$ is parameterized as a quadratic function $V(x) = (x - x_0)^T Q(x - x_0)$, where the positive-definite matrix Q and center x_0 are learned. Note that Q is constructed to be positive definite, as the learnable parameters are log-diagonal terms and off-diagonal terms of a lower triangular matrix L forming the Cholesky decomposition $Q = LL^T$.

Training Datasets and Procedure. All models were trained using the Adam optimizer with a learning rate of 1×10^{-4} for 30,000 epochs. For our proposed model, the level set parameter c is initialized at a sufficiently large number, and the regularization weight parameter λ can be chosen over a hyperparameter grid search over $10^{-1}, 10^{-2}, ..., 10^{-8}$.

The baseline MLP and our constrained model were trained on the exact same datasets for each system:

- **Lorenz 63:** The dataset consisted of 10 trajectories, each with 200 steps. Training was performed using a multi-step prediction loss with a horizon of $T = 5$.
- **Lorenz 96:** The dataset for each system consisted of 4 trajectories, each with 500 steps. Training was performed using a single-step prediction loss.
- **Reduced-order KS:** The dataset for each system consisted of 4 trajectories, each with 50,000 steps. Training was performed using a multi-step prediction loss with a horizon of $T = 5$.

Evaluation and Visualization.

Trajectory Rollouts and Simulation. For testing, long-horizon trajectories were generated by simulating the learned models autoregressively from a random initial condition not seen during training. Trajectory integration was performed using the RK4 method with a time step of $h = 0.01$ seconds. For the Lorenz 63 and the KS-ROM system, 10 trajectories of 50,000 steps were generated for creating visualizations and evaluating the statistical properties of the learned systems. For Lorenz 96, 20 trajectories of 20,000 steps were generated for evaluation.

Statistical Metrics. We used two quantitative metrics to evaluate the models' ability to reproduce the invariant statistics of the true system from the long-horizon rollouts.

- **KL Divergence:** The Kullback-Leibler (KL) divergence was calculated between the 2D probability density function (PDF) of the model's trajectory projected onto its first two principal components and the PDF of the ground truth trajectory. The PDFs were estimated using 2D histograms with 50×50 bins.
- **Spectrum Error:** The Fourier energy spectrum was first computed for the long-horizon trajectory rollouts. To quantify the discrepancy, we calculated the relative L2 norm error between the predicted spectrum (S_{pred}) and the ground truth spectrum (S_{true}). This error is given by the formula:

$$\text{Error} = \frac{\|S_{\text{pred}} - S_{\text{true}}\|_2}{\|S_{\text{true}}\|_2},$$

where $\|\cdot\|_2$ denotes the standard Euclidean norm. This metric measures the overall deviation of the predicted spectrum's shape and magnitude relative to that of the ground truth.

ACKNOWLEDGMENTS. The authors acknowledge the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing computing resources that have contributed to the results reported within this paper. This work was supported in part by MathWorks, the MIT-Amazon Science Hub, the MIT-Google Program for Computing Innovation, and the MIT-IBM Watson AI Lab.

1. EN Lorenz, Deterministic nonperiodic flow. *J. atmospheric sciences* **20**, 130–141 (1963).
2. Y Kuramoto, Diffusion-induced chaos in reaction systems. *Prog. Theor. Phys. Suppl.* **64**, 346–367 (1978).
3. GS Ashinsky, Nonlinear analysis of hydrodynamic instability in laminar flames—i. derivation of basic equations in *Dynamics of Curved Fronts*. (Elsevier), pp. 459–488 (1988).
4. A Stuart, AR Humphries, *Dynamical systems and numerical analysis*. (Cambridge University Press) Vol. 2, (1998).
5. J Guckenheimer, P Holmes, *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*. (Springer Science & Business Media) Vol. 42, (2013).
6. AJ Majda, I Timofeyev, E Vanden Eijnden, A mathematical framework for stochastic climate models. *Commun. on Pure Appl. Math. A J. Issued by Courant Inst. Math. Sci.* **54**, 891–974 (2001).
7. AJ Majda, J Harlim, Physics constrained nonlinear regression models for time series. *Nonlinearity* **26**, 201 (2012).
8. Z Li, et al., Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895* (2020).
9. M Raissi, P Perdikaris, GE Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. physics* **378**, 686–707 (2019).
10. SL Brunton, JN Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control*. (Cambridge University Press), (2022).
11. L Lu, P Jin, G Pang, Z Zhang, GE Karniadakis, Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nat. machine intelligence* **3**, 218–229 (2021).
12. D Kochkov, et al., Machine learning–accelerated computational fluid dynamics. *Proc. Natl. Acad. Sci.* **118**, e2101784118 (2021).
13. J Page, P Norgaard, MP Brenner, RR Kerswell, Recurrent flow patterns as a basis for two-dimensional turbulence: Predicting statistics from structures. *Proc. Natl. Acad. Sci.* **121**, e2320007121 (2024).
14. BO Koopman, Hamiltonian systems and transformation in hilbert space. *Proc. Natl. Acad. Sci.* **17**, 315–318 (1931).

DRAFT

15. JN Kutz, SL Brunton, BW Brunton, JL Proctor, *Dynamic mode decomposition: data-driven modeling of complex systems*. (SIAM), (2016).
16. K Pearson, Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, Dublin philosophical magazine journal science* **2**, 559–572 (1901).
17. J Mikhaeil, Z Monfared, D Durstewitz, On the difficulty of learning chaotic dynamics with rnns. *Adv. Neural Inf. Process. Syst.* **35**, 11297–11312 (2022).
18. PR Vlachas, W Byeon, ZY Wan, TP Sapsis, P Koumoutsakos, Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proc. Royal Soc. A: Math. Phys. Eng. Sci.* **474**, 20170844 (2018).
19. M Sangiorgio, F Dercole, Robustness of lstm neural networks for multi-step forecasting of chaotic time series. *Chaos, Solitons & Fractals* **139**, 110045 (2020).
20. Z Lu, BR Hunt, E Ott, Attractor reconstruction by machine learning. *Chaos: An Interdiscip. J. Nonlinear Sci.* **28** (2018).
21. PR Vlachas, et al., Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks* **126**, 191–217 (2020).
22. E Bollt, On explaining the surprising success of reservoir computing forecaster of chaos? the universal machine learning dynamical system with contrast to var and dmd. *Chaos: An Interdiscip. J. Nonlinear Sci.* **31** (2021).
23. AJ Majda, Y Yuan, Fundamental limitations of ad hoc linear and quadratic multi-level regression models for physical systems. *Discret. & Continuous Dyn. Syst. B* **17** (2012).
24. J Pathak, Z Lu, BR Hunt, M Girvan, E Ott, Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data. *Chaos: An Interdiscip. J. Nonlinear Sci.* **27** (2017).
25. Z Li, et al., Learning chaotic dynamics in dissipative systems. *Adv. Neural Inf. Process. Syst.* **35**, 16768–16781 (2022).
26. Y Schiff, et al., Dyslim: Dynamics stable learning by invariant measure for chaotic systems. *arXiv preprint arXiv:2402.04467* (2024).
27. A Wikner, BR Hunt, J Harvey, M Girvan, E Ott, Stabilizing machine learning prediction of dynamics: Noise and noise-inspired regularization. *arXiv preprint arXiv:2211.05262* (2022).
28. A Haluszczynski, J Aumeier, J Herteux, C Räth, Reducing network size and improving prediction stability of reservoir computing. *Chaos: An Interdiscip. J. Nonlinear Sci.* **30** (2020).
29. JA Platt, SG Penny, TA Smith, TC Chen, HD Abarbanel, Constraining chaos: Enforcing dynamical invariants in the training of reservoir computers. *Chaos: An Interdiscip. J. Nonlinear Sci.* **33** (2023).
30. R Jiang, PY Lu, E Orlova, R Willett, Training neural operators to preserve invariant measures of chaotic attractors. *Adv. Neural Inf. Process. Syst.* **36** (2024).
31. Y Min, SM Richards, N Azizan, Data-driven control with inherent lyapunov stability in *2023 62nd IEEE Conference on Decision and Control (CDC)*. (IEEE), pp. 6032–6037 (2023).
32. EN Lorenz, Predictability: A problem partly solved in *Proc. Seminar on predictability*. (Reading), Vol. 1, pp. 1–18 (1996).
33. E Hovmöller, The trough-and-ridge diagram. *Tellus* **1**, 62–66 (1949).
34. J Milnor, On the concept of attractor. *Commun. Math. Phys.* **99**, 177–195 (1985).
35. H Khalil, *Nonlinear Systems*, Pearson Education. (Prentice Hall), (2002).
36. G Blekherman, PA Parrilo, RR Thomas, *Semidefinite optimization and convex algebraic geometry*. (SIAM), (2012).
37. SH Strogatz, *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. (CRC press), (2018).
38. Y Min, N Azizan, HardNet: Hard-constrained neural networks with universal approximation guarantees. *arXiv preprint arXiv:2410.10807* (2024).
39. N Kovachki, et al., Neural operator: Learning maps between function spaces with applications to pdes. *J. Mach. Learn. Res.* **24**, 1–97 (2023).
40. YS Smyrlis, DT Papageorgiou, Computational study of chaotic and ordered solutions of the kuramoto-sivashinsky equation, Technical report (1996).
41. P Holmes, *Turbulence, coherent structures, dynamical systems and symmetry*. (Cambridge university press), (2012).
42. D Hendrycks, K Gimpel, Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* (2016).

DRAFT