Learning Chaotic PDEs with Boundedness Guarantees

Anonymous Author(s)
Affiliation
Address
email

Abstract

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 been focused on developing data-driven models that preserve invariant statistics over long horizons since many chaotic systems observe dissipative behaviors and ergodicity. Although these methods have shown empirical success, many of the models are still prone to generating unbounded trajectories, leading to invalid statistics evaluation. In this paper, we propose a novel neural network architecture that simultaneously learns a dissipative dynamics emulator that guarantees to generate bounded trajectories and an energy-like function that governs the dissipative behavior. More specifically, by leveraging control-theoretic ideas, we derive algebraic conditions based on the learned energy-like function that ensure asymptotic convergence to an invariant level set. Using these algebraic conditions, our proposed model enforces dissipativity through an explicit convex quadratic projection layer, which provides formal trajectory boundedness guarantees. Furthermore, the invariant level set provides an outer estimate for the strange attractor, which is known to be very difficult to characterize due to its complex geometry. We demonstrate the capability of our model in producing bounded long-horizon trajectory forecasts that preserve invariant statistics and characterizing the attractor, for chaotic dynamical systems including Lorenz 63 and the Kuramoto-Sivashinsky equation.

1 Introduction

Chaos, characterized by exponential divergence after infinitesimal initial perturbations, is ubiquitous in a variety of complex dynamical systems, including climate models [Lorenz, 1963] and turbulence in fluids [Kuramoto, 1978, Ashinsky, 1988]. 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 [Lorenz, 1963, Kuramoto, 1978], turn out to be *dissipative*, meaning that their trajectories converge to a bounded and positively invariant set, often referred to as a strange attractor [Stuart and Humphries, 1998]. Moreover, trajectories of dissipative chaos will visit almost every state on the attractor, resulting in ergodicity and invariant statistics [Guckenheimer and Holmes, 2013]. 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 empirical success 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 [Majda et al., 2001, Majda and Harlim, 2012]. 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

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

specific architecture choices or regularization schemes [Li et al., 2020, Raissi et al., 2019, Brunton and Kutz, 2022, Lu et al., 2021, Kochkov et al., 2021, Page et al., 2024]. Hybrid approaches leverage autoencoder architectures to latent representation spaces where the dynamics evolve in simpler forms, with inspirations from Koopman theory [Koopman, 1931], Dynamic mode decomposition [Kutz et al., 2016], PCA [Pearson, 1901], etc. Beyond one-step prediction, recurrent sequential models have also been explored to promote stability and improve forecast accuracy using more input information [Mikhaeil et al., 2022, Vlachas et al., 2018, Sangiorgio and Dercole, 2020]. 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 [Lu et al., 2018, Vlachas et al., 2020, Bollt, 2021].

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. Consequently, even though these models often demonstrate strong empirical performance in preserving invariant statistics, they are still prone to producing unbounded trajectories and invalid statistical forecasts. For more structured models such as multi-level quadratic regression models, theoretical analysis in [Majda and Yuan, 2012] establishes pathological instability in their statistical solutions. A fundamental difficulty in using RNNs to model chaotic systems is revealed in [Mikhaeil et al., 2022], where it was mathematically proved that the gradients of RNNs diverge during the training process. Although it is generally difficult to theoretically analyze the behaviors of data-driven machine learning models, they are also found to generate diverging trajectories resulting in invalid statistical solutions. Recent advanced time-series modeling approaches such as RCs and Fourier Neural Operators have been reported to experience the same fundamental issue in practice [Lu et al., 2018, Pathak et al., 2017, Li et al., 2022]. Thus, whether models rely heavily on explicit physical constraints or are purely data-driven, the common challenge is ensuring stable long-range rollouts.

A variety of methods have been proposed to mitigate unbounded trajectories in data-driven forecasting of chaotic systems. Within reservoir computing (RC), noise-inspired regularization [Wikner et al., 2022] and architectural pruning [Haluszczynski et al., 2020] have shown empirical improvement in stability over long horizons. Beyond RC-specific techniques, some approaches prioritize learning invariant statistics over short-time trajectory prediction by matching Lyapunov exponents and fractal dimensions [Platt et al., 2023], or by adding statistics-based regularization [Jiang et al., 2024, Schiff et al., 2024]. These methods often require prior knowledge of the systems that might be difficult to acquire; while they reduce occurrences of finite-time blowups in practice, they still lack formal guarantees of boundedness. More rigorous approaches have been proposed to directly enforce energy conservation or dissipation, such as adding energy conservation terms in multi-level quadratic regression models [Majda and Harlim, 2012] or by carefully modifying the learned flow of neural operators in a pre-specified region of state space [Li et al., 2022]. However, these methods can require restrictive model structures, substantial domain knowledge, or both. As a result, a gap remains in establishing general-purpose boundedness guarantees without heavily relying on system-specific hyperparameter tuning or prior knowledge.

In this paper, we present a neural network architecture that is inherently dissipative, ensuring bounded trajectories over arbitrarily long horizons. Rather than relying on system-specific domain knowledge, our approach learns both the underlying dynamics and the governing energy function directly from data, and naturally aligns the model's behavior with fundamental energy dissipation principles through projection. In particular, we leverage Lyapunov stability theory [Khalil, 2002] to characterize dissipativity and derive efficient algebraic conditions, embed these conditions into the model through a dissipativity projection layer to guarantee stability during training and inference, and regularize the volume of the learned level set to obtain an outer approximation of the strange attractor. We illustrate this framework using the Lorenz 63 system, and then demonstrate its effectiveness in preserving invariant statistics and generating bounded trajectories for the Kuramoto-Sivashinsky equation, highlighting its scalability and applicability in chaotic benchmarks.

2 Problem Formulation

Consider a chaotic dynamical system described with a PDE of the form,

$$\begin{aligned} \partial_t w = F\left(x, w, \partial_x w, \partial_{xx} w, ...\right), \qquad & (t,x) \in [0,T] \times \mathbb{X} \\ w(0,x) = w^0(x), \qquad & x \in \mathbb{X} \\ B\left[w\right](t,x) = 0, \qquad & (t,x) \in [0,T] \times \partial\mathbb{X} \end{aligned} \tag{1}$$

Here, $w(t,x)$ represents the n-dimensional state of the dynamical system at any time $t \in [0,T]$ and position $x \in \mathbb{X} \subseteq \mathbb{R}^d$, $w_0(x)$ is the initial condition defined on the full spatial domain $x \in \mathbb{X}$, and $B\left[w\right](t,x)$ is the boundary condition defined on the spatial boundary $\partial\mathbb{X}$. We adopt a discrete-time formulation of this problem,

$$w_{t+1}(x) = G(w_t(x), x), \qquad (t,x) \in \{0,1,2,...N\} \times \mathbb{X} \tag{2}$$

where $w_t = w(t,\cdot) : \mathbb{X} \to \mathbb{R}$ represents the state of the dynamical system at any position $x \in \mathbb{X}$ at time step t. We focus on PDEs that govern dissipative chaotic systems, as many physically relevant chaotic systems inherently exhibit dissipative behavior. The goal is to learn a neural operator $G^*(\theta)$ that emulates the true dynamics G. As discussed in Section 1, the chaotic nature of the dynamics makes long-term pointwise predictions inherently unreliable. Additionally, the trajectories of dissipative dynamics converge to a statistically invariant strange attractor, where the dynamics exhibit ergodicity. This motivates our focus on ensuring $G^*(\theta)$ captures the statistical properties of the true dynamics, rather than attempting to replicate pointwise predictions over long time horizons.

A predicted trajectory is generated by iteratively applying $G^*(\theta)$ to predict states from prior output states, causing the model to encounter regions of state space not necessarily observed during training. Approximation errors on these unseen states can accumulate upon iterative composition of $G^*(\theta)$. This, compounded with the system's intrinsic exponentially divergent behavior, can lead to trajectory blow-up in finite time, compromising the model's ability to reproduce meaningful statistical behavior. Because dissipativity is an inherent property of our systems of interest, it is natural to incorporate this property directly into the model. That is, we aim to learn the operator $G^*(\theta)$ that enforces dissipative convergence to a strange attractor, thereby emulating system dynamics and enforcing long-term boundedness.

3 Dissipative Dynamics: A Control-theoretic Perspective

To develop models with inherent dissipativity, we first need to understand theoretical conditions that make a dynamical system dissipative. 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.

For a discrete-time infinite-dimensional dynamical system in (2) whose solution at any time t is in the L^2 space, $w_t \in L^2$, we first formalize the concept of dissipativity in the following definition.

Definition 1. *We say that the system in* (2) *is* ***dissipative*** *if there exists a bounded (with respect to L^2 norm) and positively invariant set $M \subset L^2$ such that*

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

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****.*

Intuitively, a dissipative system is one that continuously loses energy, until its trajectories eventually enter and remain within a bounded region of M in its state space. For chaotic PDEs, this bounded region is its strange attractor. While many attempts have been made to characterize the strange attractor mathematically [Stuart and Humphries, 1998], these characterizations are typically abstract and computationally intractable [Milnor, 1985]. Given our goal of enforcing dissipativity in neural network models, it is crucial to first derive computationally efficient descriptions for the bounded set M, which then defines dissipative behaviors of the predicted dynamics.

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 asymptotic stability to equilibrium points [Khalil, 2002]. We apply the same strategy here by ensuring dissipativity towards a level set of the energy function, instead of analyzing the complex strange attractor. In the following proposition, we generalize the notion of asymptotic stability with respect to an equilibrium point to a level set of a Lyapunov function, which yields computationally efficient conditions that ensure dissipativity in a dynamical system.

Proposition 1 (set asymptotic stability)**.** *For an infinite-dimensional dynamical system in* (2)*, suppose there exists a non-negative-valued continuously differentiable function* $V : L^2 \to \mathbb{R}_+$ *and a constant* $c > 0$*, such that*

$$\begin{aligned} (i) \quad & \forall w_t \notin M(c) = \{w \in L^2 : V(w) \leq c\}, V(w_{t+1}) \leq \alpha V(w_t), \quad 0 < \alpha < 1 \\ (ii) \quad & \forall w_t \in M(c) = \{w \in L^2 : V(w) \leq c\}, V(w_{t+1}) \leq c \\ (iii) \quad & V \text{ is radially unbounded, i.e., } V(w) \to \infty \text{ as } \|w\| \to \infty \end{aligned}$$

Then the system (2) *is dissipative, where the level set* $M(c)$ *is globally asymptotically stable.*

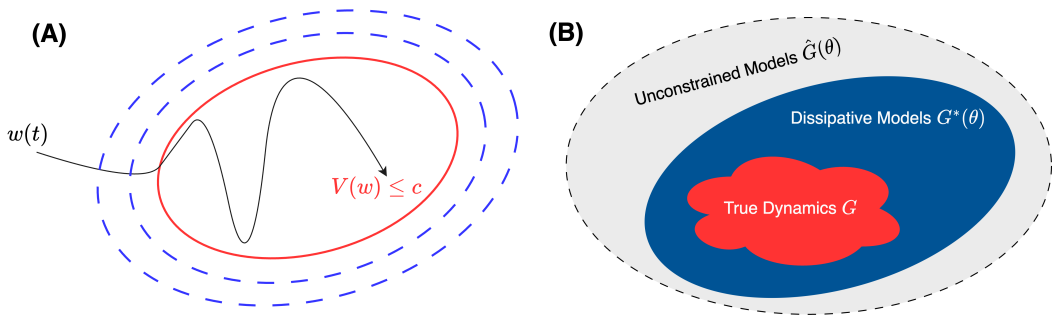


Figure 1: (A) An illustration of the conditions in Proposition 1, where the trajectory loses energy over time and enters an invariant level set. (B) Illustration of the fact that an inherently dissipative model would have an effectively smaller search space for parameters due to its alignment with true dynamics in dissipativity.

As illustrated in Figure 1(A), the conditions in Proposition 1 guide any solution starting outside the level set $M(c)$ to lose energy exponentially due to the α factor, therefore entering the level set in finite time and remaining inside thereafter. A detailed proof for Proposition 1 is included in Appendix A.

Despite the simplicity of the algebraic conditions derived in Proposition 1, overall the conditions still obtain a form of "if-else" condition, which might not be straightforward to enforce in a neural network that requires differentiability for backpropagation. To resolve this issue, we unify conditions (i) and (ii) in the above proposition into the following single inequality constraint which involves a ReLU activation and the $\alpha \in (0, 1)$ used in condition (3):

$$V(w_{t+1}) - \alpha \left[V(w_t) + \text{ReLU}\left(c - V(w_t)\right)\right] \leq 0 \tag{3}$$

Note that the reformulation here is equivalent, up to scaling the level set constant by α. More specifically, the inequality (3) reduces to $V(w_{t+1}) \leq \alpha V(w_t)$ when $V(w_t) \notin M(c)$, and reduces to $V(w_{t+1}) \leq \alpha c$ when $V(w_t) \in M(c)$. As a result, under the assumption for V made in Proposition 1, the reformulated constraint ensures that the system is dissipative and the level set $M(c)$ is globally asymptotically stable.

4 Methodology

To achieve the goal of building a neural network prediction model that ensures the trajectory it generates always stays bounded, we now introduce a framework that learns dissipative dynamics by

design, based on the control-theoretic conditions derived in Section 3. As illustrated in Figure 1(B), learning an inherently dissipative prediction model conceptually limits the parameter search to a smaller space that is always aligned with physical properties of the true dynamics. Compared to unconstrained models which might search over parameters that lead to unstable behaviors, our approach makes the training process more efficient.

Our methodology is built on two key components: 1) a learnable Lyapunov functional $V(w)$ that represents the system's energy, and 2) a custom dissipative projection layer that strictly enforces the constraint in (3). In addition to the learned model being dissipative, our framework is also able to produce an outer-estimate (the level set $M(c)$) for the complex strange attractor without any prior knowledge of the system's invariant statistics, which is known difficult to be characterized. In what follows, we discuss the details of our architecture design and training procedure.

4.1 Architecture Design with Boundedness Guarantees

We propose a neural network architecture that simultaneously learns the dynamics operator in Equation (2) and an energy-like Lyapunov functional V, which together guarantee the dissipativity conditions in Proposition 1 through the construction of a dissipative projection layer. Following common practices in learning operators in function spaces [Lu et al., 2021, Kovachki et al., 2023, Li et al., 2022], we consider a discretized spatial domain where the queried spatial location $x \in \mathbb{X}$ is sampled from a finite set $\mathbb{X}_d$ consisting of n grid points, i.e., $x \in \mathbb{X}_d \subset \mathbb{X}$ and the cardinality of $\mathbb{X}_d$ is n. As an example, if the spatial domain $\mathbb{X} = [0, 2\pi]$, a fixed grid on $\mathbb{X}$ can be n evenly sampled points, $\mathbb{X}_d = \{k\frac{2\pi}{n-1} : k = 0, 1, ..., n-1\}$. Under the grid setting, the function $w_t \in L^2$ can be effectively represented as a n–dimensional vector, which is a collection of solution values at every grid points $w_t := \{w(t, x) : x \in \mathbb{X}_d\} \in \mathbb{R}^n$. Consequently, the L^2 norm of w_t is reduced to a standard 2–norm in $\mathbb{R}^n$.

As illustrated in Figure 2(A), our model is composed of two learnable components:

1. An unconstrained dynamics emulator $\hat{G}$, which approximates the true dynamics operator G. The backbone model for the emulator $\hat{G}$ can be any neural operator that maps between function spaces. Here we choose to use DeepONet proposed in [Lu et al., 2021].
2. A quadratic Lyapunov functional $V(w) = (w - w_c)^T Q(w - w_c)$, which serves as the energy function. The learnable parameters include a positive definite matrix Q of size n–by–n and a center vector $w_c \in \mathbb{R}^n$.

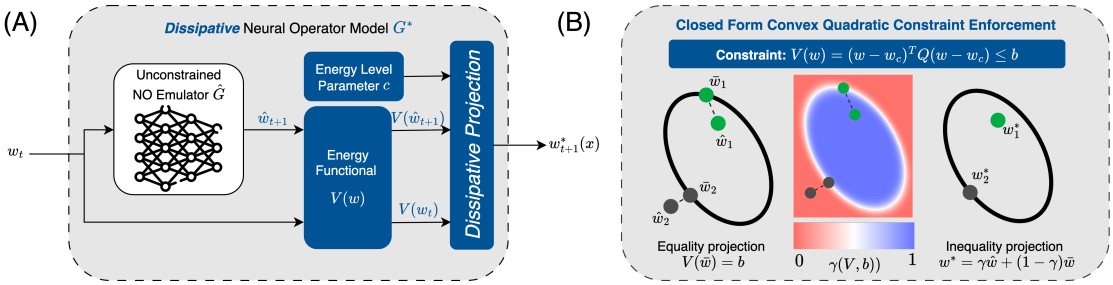


Figure 2: (A) An overview of the proposed model architecture. The input, current time solution w_t, is fed into an unconstrained neural operator (NO) emulator $\hat{G}$ to produce a preliminary prediction $\hat{w}_{t+1}$ and a learned energy functional V to compute its energy $V(w_t)$. The dissipative projection layer modifies $\hat{w}_{t+1}$ to produce a final output w^*_{t+1} that satisfies the dissipative energy constraint in (3). (B) Illustration of the convex quadratic projection for a constraint in the form of $V(w) \leq b$. The equality projection maps any point not on the ellipsoid boundary w to a boundary point $\bar{w}$ in closed form (for both w_1, w_2). The quadratic projection is only active for when the constraint is violated, so w_2 is projected while w_1 is left unchanged.

These components are integrated into a *dissipative projection* layer, which modifies the output of the unconstrained emulator $\hat{G}$ to produce an operator G^* that maps the current solution w_t to the predicted solution at the next time step w^*_{t+1}. By construction, the *dissipative projection* layer ensures

the condition in (3) is satisfied, which guarantees the predicted dynamical system $w^*_{t+1} = G^*(w_t)$ is dissipative. As a direct consequence of Definition 1, all trajectories generated by the predicted dynamical systems in an autoregressive manner are guaranteed to be bounded.

We now summarize the main theoretical guarantees our framework provides in the following theorem:

Theorem 1. *Let the learned dynamics be defined by the operator $w^*_{t+1} = G^*(w^*_t)$, which is composed of an unconstrained neural operator emulator $\hat{G}$ and a dissipative projection layer. Let the learned energy-like function be a quadratic Lyapunov functional $V(w) = (w - w_c)^T Q(w - w_c)$ with learnable center $w_c \in \mathbb{R}^n$ and a symmetric positive definite matrix $Q \in \mathbb{S}^n_{++}$. If for any state w_t, the dissipative projection layer ensures that the final model output w^*_{t+1} satisfies the constraint: $V(w^*_{t+1}) \leq \alpha[V(w_t) + ReLU(c - V(w_t))]$ for $c > 0$ and $0 < \alpha < 1$, then the learned dynamical system is dissipative. Consequently, the level set $M(c)$ is globally asymptotically stable, and all trajectories generated by the learned dynamics are guaranteed to be bounded.*

4.2 Convex Quadratic Projection Layer

The quadratic form of the Lyapunov functional $V(w)$ motivates the development of a convex quadratic projection layer that projects the model predictions onto a feasible set of trajectories where the dynamics are dissipative. We introduce a differentiable convex quadratic projection layer shown in Figure 2(B) that can handle constraints of the form $(w - w_c)^T Q(w - w_c) \leq b$, where b can be a constant or an arbitrary function of the model input. This form is equivalent to (3), with $b = \alpha\,[V(w_t) + ReLU(c - V(w_t))]$.

The convex quadratic constraint projection is illustrated in Figure 2. The general strategy is to define a projection $\bar{w}$ of the model output $\hat{w}$ onto the equality constraint $V(\bar{w}) = b$ and selectively project points that violate the constraint $V(\hat{w}) \leq b$ onto their respective equality projection. For positive definite Q, there exists an explicit form for the projection of $\hat{w}$ onto the equality constraint $V(\bar{w}) = b$. That is, for the quadratic Lyapunov function described in Section 4.1, the projection

$$\bar{w} = w_0 + \sqrt{b}\left(L^T\right)^{-1} \frac{\hat{w}}{||\hat{w}||_2} \tag{4}$$

satisfies the equation $V(\bar{w}) = b$, where L is the Cholesky decomposition of Q such that $LL^T = Q$. To ensure that this equality projection is only active when the constraint is violated, the final output w^* is calculated as an interpolation between the projected ($\bar{w}$) and non-projected ($\hat{w}$) outputs.

$$w^*(x) = \gamma(b, V(\hat{w}))\hat{w}(x) + [1 - \gamma(b, V(\hat{w}))]\,\bar{w}(x) \tag{5}$$

Ideally, $\gamma(b, V(\hat{w}))$ is an indicator function that is 1 when $V(\hat{w}_t) \leq b$ and 0 when $V(\hat{w}_t) > b$. In practice, we use a differentiable function $\gamma(b, V(\hat{w})) = \text{sigmoid}\,[k(b - V(\hat{w})]$. In doing so, we trade off strict adherence to the constraint for model tractability, although a large value for k pushes the projection closer to strict constraint enforcement. We use $k = 100$ for all experiments.

4.3 Training with Invariant Set Volume Regularization

We construct the training dataset as a collection of N input-output pairs, denoted as $\{(w_i, w_{next,i})\}_{i=1}^N$, where the input w_i is the current time step solution and the output $w_{next,i}$ is the next time step solution based on the true dynamics. During training, each input w_i is mapped to a predicted output $w^*_{next,i}$, and the loss is computed relative to the true next state $w_{next,i}$. The dynamic loss is defined as the average mean squared error (MSE) between the predicted outputs $\{w^*_{next,i}\}_{i=1}^N$ and the ground truth future states $\{w_{next,i}\}_{i=1}^N$.

While the convex quadratic projection layer enforces dissipativity and convergence to a level set $M(c)$, it does not inform how to choose an appropriate level set that characterizes the attractor. The goal is to learn the energy functional $V(w)$ such that the resulting ellipsoid is a tight outer estimate of the attractor. To this end, we include a regularization loss in the loss function that penalizes large ellipsoid volumes.

$$\text{Loss} = \frac{1}{N}\sum_{i=1}^{N} \|w^*_{next,i} - w_{next,i}\|_2^2 + \lambda \frac{1}{\sqrt{\det(Q)}}, \tag{6}$$

where Q is defined in Section 4.1 and $\lambda > 0$ is a regularization hyperparameter.

5 Numerical Experiments

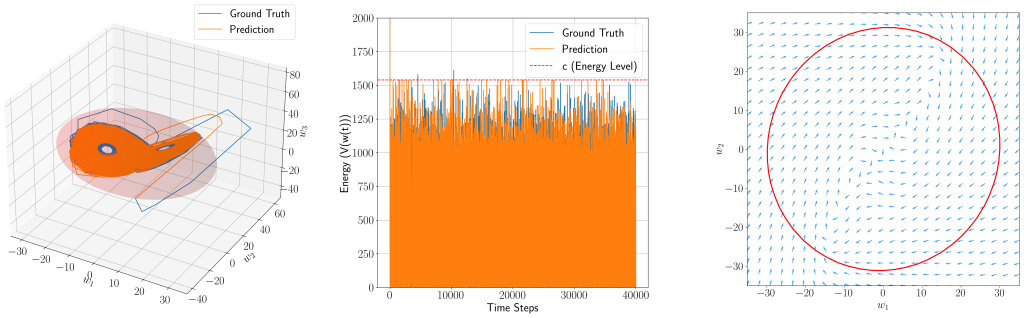
5.1 Lorenz 63

We first apply our methodology to the classic Lorenz 63 system, a three-dimensional model of atmospheric convection originally proposed by Lorenz [Lorenz, 1963]. Its low dimensionality makes it ideal for visualizing the system's strange attractor and dissipative flow. The governing dynamics is described as the following ordinary differential equation (ODE), where parameters $\sigma = 10.0, \rho = 28.0, \beta = 8/3$ generate chaotic behaviors, and $w \in \mathbb{R}^3$ is the system state,

$$\dot{w}_1 = \sigma(w_2 - w_1),\ \dot{w}_2 = w_1(\rho - w_3) - w_2,\ \dot{w}_3 = w_1 w_2 - \beta w_3.$$

This system can be viewed as a special case of an infinite-dimensional dynamical system where the spatial domain consists of just three discrete points. In this simplified context, backbone neural operators like the Fourier Neural Operator [Kovachki et al., 2023] and DeepONet [Lu et al., 2021] reduce to a simple multilayer perceptron (MLP). Accordingly, we constructed our unconstrained neural operator emulator, $\hat{G}$, as a feedforward neural network with 6 hidden layers, each containing 150 neurons.

The training data was generated from a single long trajectory with a sampling interval of 0.05 seconds, initialized at a random state outside the strange attractor. To evaluate the trained MLP, we generated a test trajectory by starting from a new initial condition unseen during training, and iteratively applying the model $w^*_{t+1} = G^*(w_t)$ for 40,000 steps.



(a) Long-term trajectory rollout. (b) Learned energy level $V(w_t)$. (c) Learned flow field dynamics.

Figure 3: **Lorenz 63 prediction results.** (a) A 40,000-step trajectory generated by the model from an unseen initial condition (orange) compared to the ground truth attractor (blue). The learned invariant ellipsoid (red) is a tight outer-estimate of the strange attractor. (b) The energy of the predicted trajectory quickly drops below the energy level c, and remains bounded by c. (c) The learned flow field on the w_1-w_2 plane, showing that vectors correctly point inwards across the ellipsoid boundary.

The results for the Lorenz 63 system illustrate our model's ability to learn stable, dissipative dynamics and characterize the strange attractor. Figure 3a shows a 40,000-step trajectory generated from an initial condition not seen during training. The model's prediction (orange) accurately recovers the geometry of the true attractor formed by the ground truth trajectory (blue). Crucially, the learned invariant ellipsoid (red) serves as a tight outer-estimate of the attractor, validating the effectiveness of our volume regularization. The learned energy function evaluations on both prediction and true trajectories are visualized in Figure 3b, which validates that the energy level of the true system is indeed bounded by the level set parameter c, and that the learned model is dissipative as it quickly loses energy and remains in the invariant set. Figure 3c further visualizes this dissipative behavior by showing the learned flow field on the $w_1 - w_2$ plane. The vector fields along the boundary of the learned ellipsoid all point inwards, ensuring that the set is positively invariant. Together, these results validate our framework's capacity to produce bounded long-horizon forecasts while simultaneously learning a meaningful energy function that characterizes the system's attractor.

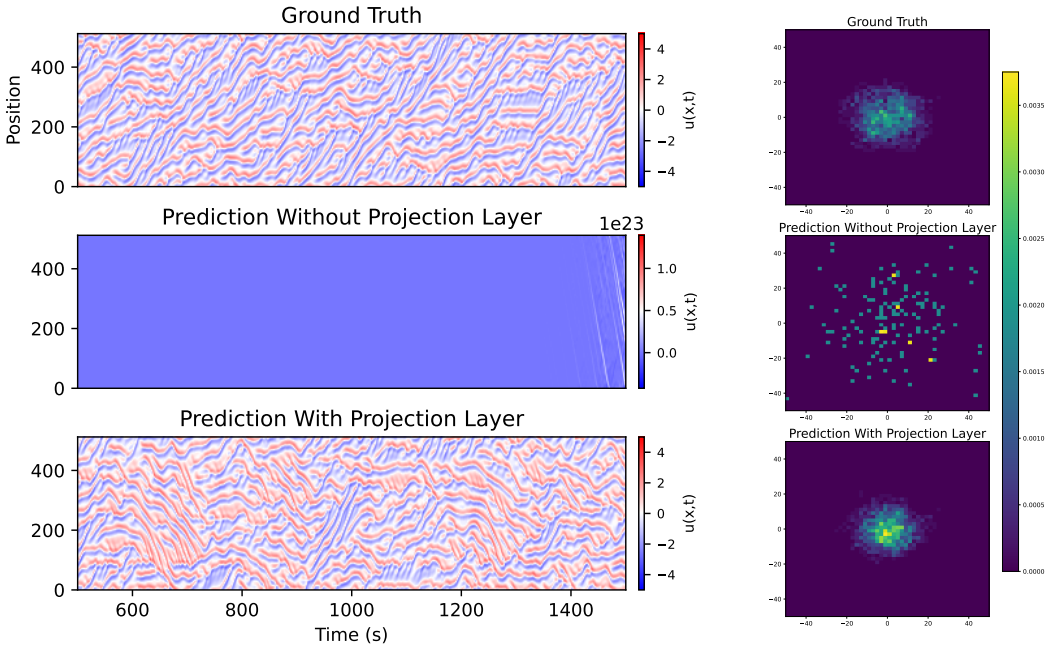
5.2 Kuramoto-Sivashinsky

To demonstrate the effectiveness of our proposed architecture on predicting long-horizon trajectories of chaotic PDEs, we consider the one-dimensional chaotic Kuramoto-Sivashinsky equation.

$$\begin{aligned} w_t + w_{xx} + w_{xxxx} + \frac{1}{2}(w^2)_x = 0, \quad (t,x) \in [0,\infty) \times [0,L] \\ w(0,x) = w^0(x), \quad x \in [0,L] \end{aligned} \tag{7}$$

We train on a dataset consisting of six trajectories starting from random initial conditions, where $L = 32\pi$ is chosen to generate chaotic behaviors. Each trajectory has snapshots every 1 second for 500 seconds, and each snapshot has a spatial resolution of 512. We use a validation set of two trajectories with the same spatial and temporal discretization as the training data set. Two DeepONet architectures are trained on the same dataset, one with the added projection layer discussed in Section 4.2 and one vanilla model without. For both models, the DeepONet branch network consists of three convolutional layers of output dimension 32, 64,128, and two fully connected layers, each with 256 neurons. The trunk network consists of four fully connected layers, each with 256 neurons.

To test the trained models with and without projection, we iteratively compose each of the trained models repeatedly for 2000 time steps to characterize the statistical properties, starting from an unseen random initial condition. Figure 4(a) validates that our model with dissipative projection generates bounded trajectory prediction that recovers the flow patterns seen in the true trajectory, while the predictions from the unconstrained model blow up. Not only does our proposed methodology produce bounded dynamics, it successfully replicates statistical properties of the true dynamics. As observed in Figure 4(b), our projected model samples a similar attractor to the true dynamics, indicating convergence to the strange attractor, while the unconstrained model produces a distribution that is sparse across a larger state space. We include additional energy and statistical property comparisons between the ground truth, our model, and the unconstrained model in Appendix B, to further demonstrate the effectiveness of our approach in preserving invariant statistics.



(a) Long term trajectory rollout.

(b) Projection of trajectories onto PCA modes.

Figure 4: **KS prediction results.** (a) Comparison of ground truth trajectory with predictions from vanilla and projected model. Trajectories are visualized for 1000 seconds after a transient period of 500 seconds. The vanilla model blows up, while the projected model stays bounded. (b) Projection of trajectories onto the first two PCA modes. The projected model and ground truth sample the strange attractor.

References

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

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

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.

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

John Guckenheimer and Philip Holmes. *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, volume 42. Springer Science & Business Media, 2013.

Andrew J Majda, Ilya Timofeyev, and Eric Vanden Eijnden. A mathematical framework for stochastic climate models. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 54(8):891–974, 2001.

Andrew J Majda and John Harlim. Physics constrained nonlinear regression models for time series. *Nonlinearity*, 26(1):201, 2012.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

Dmitrii Kochkov, Jamie A Smith, Ayya Alieva, Qing Wang, Michael P Brenner, and Stephan Hoyer. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21):e2101784118, 2021.

Jacob Page, Peter Norgaard, Michael P Brenner, and Rich R Kerswell. Recurrent flow patterns as a basis for two-dimensional turbulence: Predicting statistics from structures. *Proceedings of the National Academy of Sciences*, 121(23):e2320007121, 2024.

Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.

Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901.

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.

Pantelis R Vlachas, Wonmin Byeon, Zhong Y Wan, Themistoklis P Sapsis, and Petros Koumoutsakos. Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2213):20170844, 2018.

Matteo Sangiorgio and Fabio Dercole. Robustness of lstm neural networks for multi-step forecasting of chaotic time series. *Chaos, Solitons & Fractals*, 139:110045, 2020.

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

Pantelis-Rafail Vlachas, Jaideep Pathak, Brian R Hunt, Themistoklis P Sapsis, Michelle Girvan, Edward Ott, and Petros Koumoutsakos. Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks*, 126:191–217, 2020.

Erik 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 Interdisciplinary Journal of Nonlinear Science*, 31(1), 2021.

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.

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.

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.

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.

Alexander Haluszczynski, Jonas Aumeier, Joschka Herteux, and Christoph Räth. Reducing network size and improving prediction stability of reservoir computing. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(6), 2020.

Jason A Platt, Stephen G Penny, Timothy A Smith, Tse-Chun Chen, and Henry DI Abarbanel. Constraining chaos: Enforcing dynamical invariants in the training of reservoir computers. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 33(10), 2023.

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.

Yair Schiff, Zhong Yi Wan, Jeffrey B Parker, Stephan Hoyer, Volodymyr Kuleshov, Fei Sha, and Leonardo Zepeda-Núñez. Dyslim: Dynamics stable learning by invariant measure for chaotic systems. *arXiv preprint arXiv:2402.04467*, 2024.

H.K. Khalil. *Nonlinear Systems*. Pearson Education. Prentice Hall, 2002. ISBN 9780130673893. URL `https://books.google.com/books?id=t_d1QgAACAAJ`.

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

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

A Proof for Theoretical Results

Proof for Proposition 1. By definition, condition (ii) implies that $M(c)$ is indeed an positively invariant set. Since V is radially unbounded, for any $\alpha > 0$, we can find r_α such that $V(w) > \alpha$ for all $\|w\| > r_\alpha$. Therefore, any level set of V is bounded as $\{x : V(w) \leq \alpha\} \subset B(r_\alpha)$. Thus, $M(c)$ is both positively invariant and bounded.

Based on the positive invariance property, any trajectory starting with $w_0 \in M(c)$ will always stays within $M(c)$, meaning that $\lim_{t\to\infty} \text{dist}(w_t, M(c)) = 0$ is satisfied.

Consider a trajectory $\{w'_t\}_{t\in\mathbb{N}}$ that starts outside of the level set, i.e. $w'_0 \notin M(c)$ and $V(w'_0) > c > 0$. Suppose the trajectory never enters $M(c)$, i.e., $\forall t \in \mathbb{N}, V(w'_t) > c$. Using condition (i), we have $V(w'_{t+1}) \leq \alpha V(w'_t)$, which implies $V(w'_t) \leq \alpha^t V(w'_0)$. For any $t \geq \log_\alpha \left(\frac{c}{V(w'_0)}\right)$, $V(w'_t) \leq c$ which contradicts the prior assumption. In fact, the trajectory $\{w'_t\}_{t\in\mathbb{N}}$ will enter $M(c)$ in finite time at most $\log_\alpha \left(\frac{c}{V(w'_0)}\right)$ steps. □

Proof for Theorem 1. The dissipativity of the learned dynamics and global asymptotic stability of the level set $M(c)$ is a direct consequence of Proposition 1 and the equivalence of the inequality constraint in (3).

Note that for the predicted trajectory under the learned dynamics, at any time $t > 0$, $V(w^*_t)$ is strictly bounded by $max\{V(w_0), c\}$, which implies $V(w^*_t) \leq V(w_0)$. Let $z = w^*_t - w_c$, the constraint can be rewritten as $z^T Q z \leq V(w_0)$. Since Q is positive definite, we have $\lambda_m in(Q)\|z\|^2 \leq z^T Q z \leq V(w_0)$. Thus w^*_t will always be bounded with respect to its L^2 norm. □

B Additional Numerical Results for Kuramoto-Sivashinsky

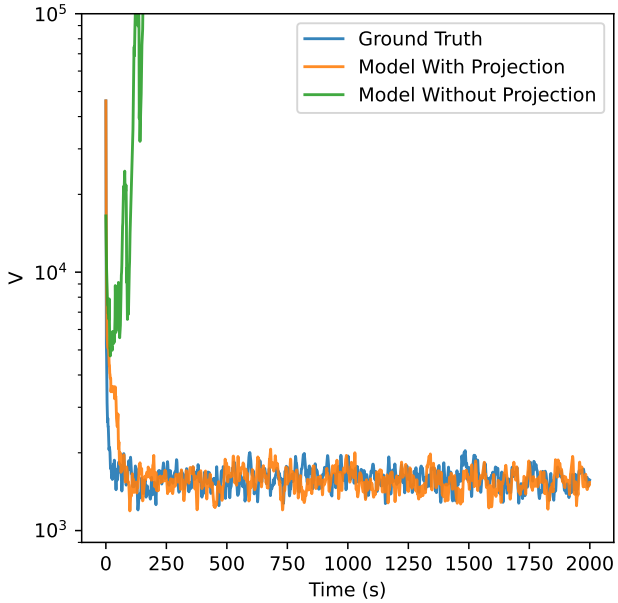


Figure 5: Energy $V(w)$ over time for the vanilla model and our projected model compared with the ground truth. It is observed that the energy blows up for the vanilla model, while the energy of the projected trajectory remains bounded.

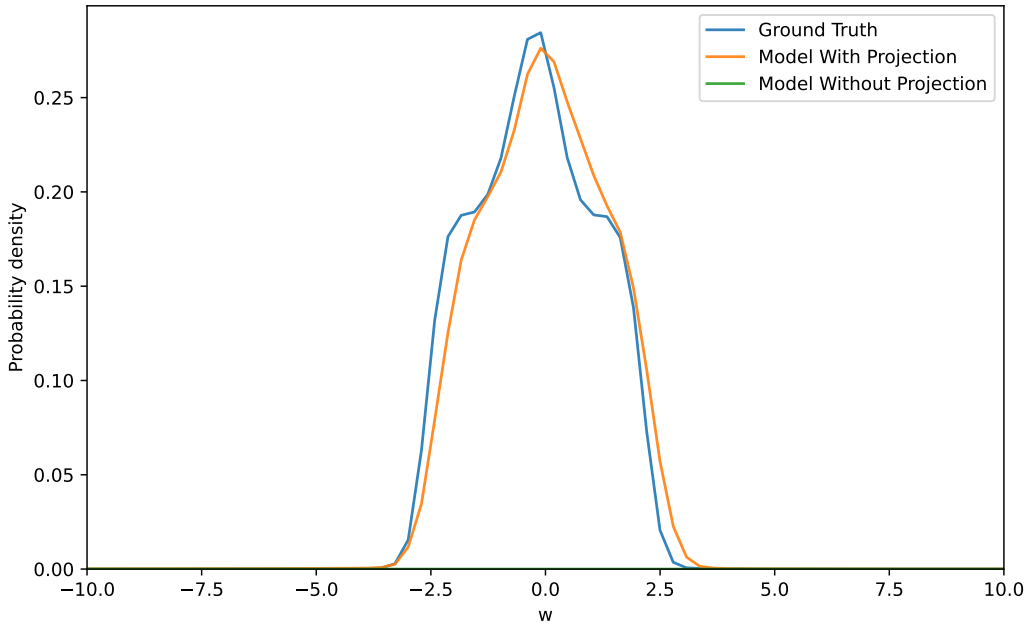


Figure 6: Probability distributions of the state w sampled in time for the true dynamics, vanilla model, and dissipative projection model. The curve for the vanilla model (model without projection) sits at just above 0, as the divergent trajectory samples many points on an unbounded domain. The distribution for the projected model is similar to that of the ground truth.