# SPARSE IDENTIFICATION OF NONLINEAR DYNAMICS WITH SHALLOW RECURRENT DECODER NETWORKS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Spatio-temporal modeling of real-world data is a challenging problem as a result of inherent high-dimensionality, measurement noise, and expensive data collection procedures. In this paper, we present **S**parse **I**dentification of **N**onlinear **Dy**namics with **SH**allow **R**ecurrent **D**ecoder networks (SINDy-SHRED) to jointly solve the sensing and model identification problems with simple implementation, efficient computation, and robust performance. SINDy-SHRED utilizes Gated Recurrent Units (GRUs) to model the temporal sequence of sensor measurements along with a shallow decoder network to reconstruct the full spatio-temporal field from the latent state space using only a few available sensors. Our proposed algorithm introduces a SINDy-based regularization. Beginning with an arbitrary latent state space, the dynamics of the latent space progressively converges to a SINDy-class functional, provided the projection remains within the set. We conduct a systematic experimental study including synthetic PDE data, real-world sensor measurements for sea surface temperature, and direct video data. With no explicit encoder, SINDy-SHRED allows for efficient training with minimal hyperparameter tuning and laptop-level computing. SINDy-SHRED demonstrates robust generalization in a variety of applications with minimal to no hyperparameter adjustments. Additionally, the interpretable SINDy model of latent state dynamics enables accurate long-term video predictions, achieving state-of-the-art performance and outperforming all baseline methods considered, including Convolutional LSTM, PredRNN, ResNet, and SimVP.

## 1 INTRODUCTION

Modeling unknown physics is an exceptionally challenging task that is complicated further by the computational burden of high-dimensional state spaces and expensive data collection. Partial differential equations (PDEs) derived from first principles remain the most ubiquitous class of models to describe physical phenomena. However, we frequently find that the simplifying assumptions necessary to construct a PDE model can render it ineffectual for real data where the physics is multi-scale in nature, only partially known, or where first principles models currently do not exist. In such cases, machine learning (ML) method offers an attractive alternative for learning both the physics and coordinates necessary for quantifying observed spatio-temporal phenomenon. Many recent efforts utilizing ML techniques seek to relax the computational burden for PDE simulation by learning surrogate models to forward-simulate or predict spatiotemporal systems. However, this new machine learning paradigm frequently exhibits instabilities during the training process, unstable roll outs when modeling future state predictions, and often yields minimal computational speedups.

Shallow Recurrent Decoder networks (SHRED) (Williams et al., 2024) are a recently introduced architecture that utilize data from sparse sensors to reconstruct and predict the entire spatiotemporal domain. Similar to Takens' embedding theorem, SHRED models trade spatial information at a single time point for a trajectory of sensor measurements across time. Previous work has shown SHRED can achieve excellent performance in examples ranging from weather forecasting, atmospheric ozone concentration modeling, and turbulent flow reconstructions. In this paper, we introduce **S**parse **I**dentification of **N**onlinear **Dy**namics with **SH**allow **Re**current **D**ecoder networks (SINDy-SHRED). SINDy-SHRED exploits the latent space of recurrent neural networks for sparse sensor modeling, and enforces interprebility via a SINDy-based functional class. In this way, SINDy-SHRED enables a robust and sample-efficient joint discovery of the governing equation and

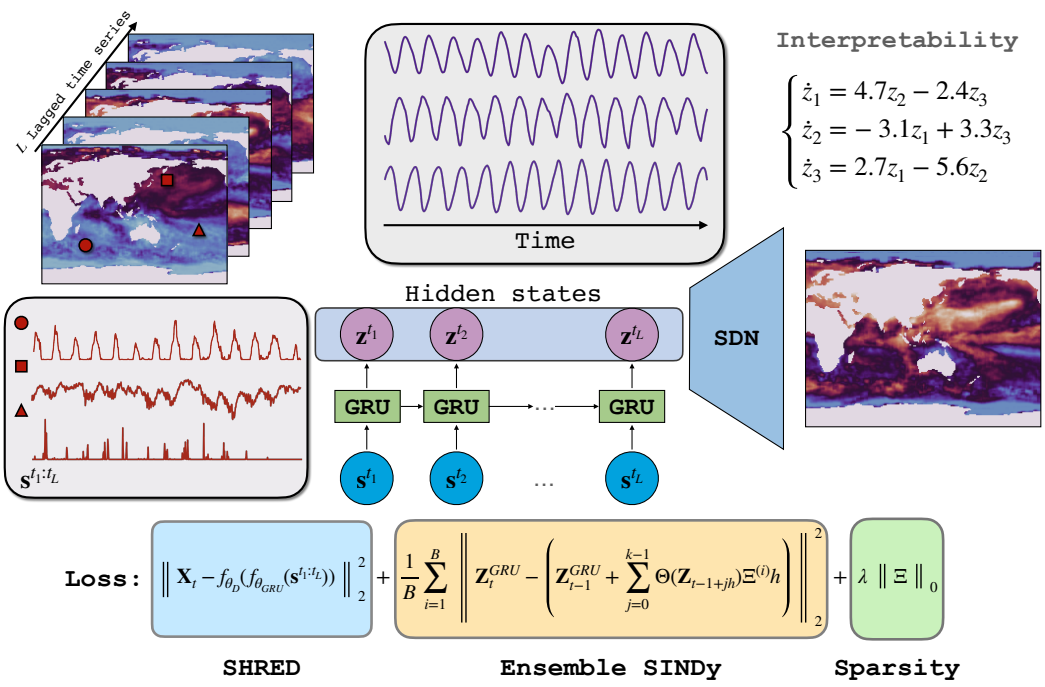



Figure 1: Illustration of the SINDy-SHRED architecture. SINDy-SHRED transfers the original sparse sensor signal (red) to an interpretable latent representation (purple) that falls into the SINDy-class functional. The shallow decoder performs a reconstruction in the pixel space.

coordinate system. With the correct governing equation, SINDy-SHRED can perform an accurate long-term prediction in a learned latent space, and in turn allow for long-term forecasting in the pixel space. For practical applications, SINDy-SHRED is a lightweight model which can perform low-rank recovery with only a few (e.g. three) active sensors, which is critical for large-scale scientific data modeling and real-time control. It does not require large amounts of data during training, thereby avoiding a common pitfall in existing ML techniques for accelerating physics simulations. SINDy-SHRED also exhibits remarkable training speed, even when executed on a single laptop. Furthermore, SINDy-SHRED is highly reproducible with minimal effort in hyperparameter tuning. The recommended network structure, hyperparameter, and training setting can generalize to many different datasets. In short, we demonstrate SINDy-SHRED to be very robust and highly applicable in many modern scientific modeling problems.

Existing work seeking to perform data-driven, long-term forecasts of spatio-temporal phenomena typically suffers from (i) instabilities and (ii) massive computational requirements. We find that SINDy-SHRED ameliorates many of these issues because (i) it is based on a stable equation discovery and (ii) the learned model is an ODE in a learned latent space, rendering simulation more computationally efficient. We further conjecture that the fact SINDy-SHRED does not include a spatial encoder contributes to the architecture's robustness, rendering it more difficult for the model to overfit during training.

We perform a wide range of studies to demonstrate the effectiveness of SINDy-SHRED. We first apply the model on the sea surface temperature data, which is a complex real-world problem. We also consider data from a complex simulation of atmospheric chemistry, video data of flow over a cylinder, and video data of a pendulum. The ability of SINDy-SHRED to perform well on video data is an important result for so-called "GoPro physics." With extremely small sample size and noisy environments, SINDy-SHRED achieves governing equation identification with stable long-term predictions. Finally, we demonstrate the performance of SINDy-SHRED on a chaotic 2D Kolmogorov flow (in Appendix D), finding a reasonable model even in the presenece of chaos. The contribution of our paper is three-fold.

- We propose SINDy-SHRED to incorporate symbolic understanding of the latent space of recurrent models of spatio-temporal dynamics.
- We further analyze the latent space of case studies and propose scientific models for these systems.
- We systematically study SINDy-SHRED and compare to popular deep learning algorithms in spatio-temporal prediction.

## 2 Related works

Traditionally, spatio-temporal physical phenomena are modeled by Partial Differential Equations (PDEs). To accelerate PDE simulations, recent efforts have leveraged neural networks to model physics. By explicitly assuming the underlying PDE, physics-informed neural networks (Raissi et al., 2019) utilize the PDE structure as a constraint for small sample learning. However, assuming the exact form of governing PDE for real data can be a strong limitation. There have been many recent works on learning and predicting PDEs directly using neural networks (Khoo et al., 2021; Li et al., 2020b; Holl et al., 2020; Lu et al., 2021; Lin et al., 2021). Meanwhile, PDE-find (Rudy et al., 2017; Messenger & Bortz, 2021; Fasel et al., 2022) offers a data-driven approach to identify PDEs from the spatial-temporal domain. Still, the high-dimensionality and required high data quality can be prohibitive for practical applications.

In parallel, previous efforts in the discovery of physical law through dimensionality reduction techniques (Champion et al., 2019; Lusch et al., 2018; Mars Gao & Nathan Kutz, 2024) provide yet another perspective on the modeling of scientific data. The discovery of physics from a learned latent space has previously been explored by (Fukami et al., 2021; Cheng et al., 2024; Farenga et al., 2024; Conti et al., 2023; Wu et al., 2022; Li et al., 2020a), yet none of these methods consider a regularization on the latent space with no explicit encoder. Yu et al. proposed the idea of physics-guided learning (Yu & Wang, 2024) which combines physics simulations and neural network approximations. Directly modeling physics from video is also the subject of much research in the field of robotics (Finn et al., 2016; Todorov et al., 2012; Sanchez-Gonzalez et al., 2018), computer vision (Xie et al., 2024; Wu et al., 2017) and computer graphics (Kandukuri et al., 2020; Liu et al.; Wu et al., 2015; Mrowca et al., 2018), since many fields of research require better physics models for simulation and control. From the deep learning side, combining the structure of differential equations into neural networks (He et al., 2016; Chen et al., 2018) has been remarkably successful in a wide range of tasks. When spatial-temporal modeling is framed as a video prediction problem, He et al. found (He et al., 2022) that random masking can be an efficient spatio-temporal learner, and deep neural networks can provide very good predictions for the next 10 to 20 frames (Shi et al., 2015; Wang et al., 2017; Gao et al., 2022; Guen & Thome, 2020). Generative models have also been found to be useful for scientific data modeling (Mirza, 2014; Song et al., 2021; Cachay et al., 2024).

## 3 Methods

The shallow recurrent decoder network (**SHRED**) is a computational technique that utilizes recurrent neural networks to predict the spatial domain. (Williams et al., 2024). The method functions by trading high-fidelity spatial information for trajectories of sparse sensor measurements at given spatial locations. Mathematically, consider a high-dimensional data series $\{\boldsymbol{X}_i\}_{i=1}^T \in \mathbb{R}^{(W\times H)\otimes T}$ that represents the evolution of a spatio-temporal dynamical system, $W$, $H$, and $T$ denote the width, height, and total time steps of the system, respectively. In SHRED, each sensor collects data from a fixed spatial position in a discretized time domain. Denote the subset of sensors as $\mathcal{S}$, the input data of SHRED is $\{\boldsymbol{X}^{\mathcal{S}}\}_{i=1}^T \in \mathbb{R}^{\text{card}(S)\otimes T}$. Provided the underlying PDE allows spatial information to propagate, these spatial effects will appear in the time history of the sensor measurements, enabling the sensing of the entire field using only a few sensors. In vanilla SHRED, a Long Short-Term Memory (LSTM) module is used to map the sparse sensor trajectory data into a latent space, followed by a shallow decoder to reconstruct the entire spatio-temporal domain at the current time step.

SHRED enables efficient sparse sensing that is widely applicable to many scientific problems (Ebers et al., 2024; Kutz et al., 2024; Riva et al., 2024). The advantage of SHRED comes from three aspects. First, SHRED only requires minimal sensor measurements. Under practical constraints, collecting full-state measurements for data prediction and control can be prohibitively expensive.

Second, SHRED does not require grid-like data collection, which allows for generalization to more complex data structures. For example, it is easy to apply SHRED to graph data with an unknown underlying structure, such as human motion data on joints, robotic sensor data, and financial market data. Moreover, SHRED is theoretically rooted in PDE modeling methods from the perspective of separation of variables, which has the potential to offer strong theoretical guarantees such as convergence and stability.

### 3.1 Empowering SHRED with representation learning and physics discovery

To achieve a parsimonious representation of physics, it is important to find a representation that effectively captures the underlying dynamics and structure of the system. In SINDy-SHRED (shown in Fig. 1), we extend the advantages of SHRED, and perform a joint discovery of coordinate systems and governing equations. This is accomplished by enforcing that the latent state of the recurrent network follows an ODE in the SINDy class of functions.

**Finding better representations** SHRED has a natural advantage in modeling the latent governing physics due to its small model size. SHRED is based on a shallow decoder with a relatively small recurrent network structure. The relative simplicity of the model allows the latent representation to maintain many advantageous properties such as smoothness and Lipschitzness. Experimentally, we observe that the hidden state space of a SHRED model is generally very smooth. Second, SHRED does not have an explicit encoder, which avoids the potential problem of spectral bias (Rahaman et al., 2019). Many reduced-order modeling methods that rely on an encoder architecture struggle to learn physics and instead focus only on modeling the low-frequency information (background) (Refinetti & Goldt, 2022; Champion et al., 2019; Mars Gao & Nathan Kutz, 2024). Building upon SHRED, we further incorporate SINDy to regularize the learned recurrence with a well-characterized and simple form of governing equation. In other words, we perform a joint discovery of a coordinate system (which transfers the high-dimensional observation into a low-dimensional representation) and the governing law (which describes how the summarized latent representation progresses forward with respect to time) of the latent space of a SHRED model. This approach is inspired by the principle in physics that, under an ideal coordinate system, physical phenomena can be described by a parsimonious dynamical model (Champion et al., 2019; Mars Gao & Nathan Kutz, 2024). When the latent representation and the governing law are well-aligned, this configuration is likely to capture the true underlying physics. This joint discovery results in a latent space that is both interpretable and physically meaningful, enabling robust and stable future prediction based on the learned dynamics.

### 3.2 SINDy-SHRED: latent space regularization via SINDy

As a compressive sensing procedure, there exist infinitely many equally valid solutions for the latent representation. Therefore, it is not necessary for the latent representation induced by SHRED to follow a well-structured differential equation. For instance, even if the exhibited dynamics are fundamentally linear, the latent representation may exhibit completely unexplainable dynamics, making the model challenging to interpret and extrapolate. Therefore, in SINDy-SHRED, our goal is to further constrain the latent representations to lie within the SINDy-class functional. This regularization promotes models that are fundamentally explainable by a SINDy-based ODE, allowing us to identify a parsimonious governing equation. The SINDy class of functions typically consists of a library of commonly used functions, which includes polynomials, and Fourier series. Although they may seem simple, these functions possess surprisingly strong expressive power, enabling the model to capture very complex dynamical systems.

**SINDy as a Recurrent Neural network** We first reformulate SINDy using a neural network form, simplifying its incorporation into a SHRED model. ResNet (He et al., 2016) and Neural ODE (Chen et al., 2018) utilize skip connections to model residual and temporal derivatives. Similarly, this could also be done via a Recurrent Neural Network (RNN) which has a general form of

$$z_{t+1} = z_t + f(x_t), \tag{1}$$

where $f(\cdot)$ is some function of the input. From the Euler method, the ODE forward simulation via SINDy effectively falls into the category of Recurrent Neural Networks (RNNs) which has the form

$$z_{t+1} = z_t + f_\Theta(x_t, \Xi, \Delta t), \tag{2}$$

where $f_\Theta(x_t, \Xi, \Delta t) = \Theta(x_t)\Xi\Delta t$ is a nonlinear function. Notice that this $f_\Theta(\cdot)$ has exactly the same formulation as in SINDy (Brunton et al., 2016). The application of function libraries with sparsity constraints is a manner of automatic neural architecture search (NAS) (Zoph & Le, 2016). Compared to all prior works (Champion et al., 2019; Fukami et al., 2021; Conti et al., 2023), this implementation of the SINDy unit fits better in the framework of neural network training and gradient descent. We utilize trajectory data $\{\mathbf{z}_i\}_{i=1}^T$ and forward simulate the SINDy-based ODE using a trainable parameter $\Xi$. To achieve better stability and accuracy for forward integration, we use Euler integration with $k$ mini-steps (with time step $\frac{\Delta t}{k}$) to obtain $\mathbf{z}_{t+1}$. In summary, defining $h = \frac{\Delta t}{k}$, we optimize $\Xi$ with the following:

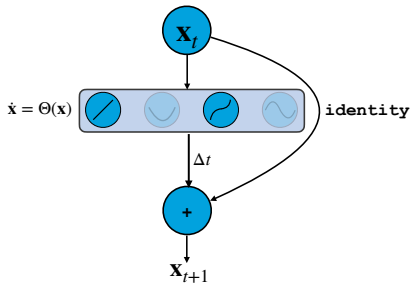



Figure 2: Diagram of the RNN form of SINDy.

$$\Xi = \arg\min \left\| \mathbf{z}_{t+1} - \left( \mathbf{z}_t + \sum_{i=0}^{k-1} \Theta(\mathbf{z}_{t+ih})\Xi h \right) \right\|_2^2, \quad \mathbf{z}_{t+ih} = \mathbf{z}_t + \Theta(\mathbf{z}_{t+(i-1)h})\Xi h, \quad \min \|\Xi\|_0 . \tag{3}$$

To achieve $\ell_0$ optimization, we perform pruning with $\ell_2$ which approximates $\ell_0$ regularization under regularity conditions (Zheng et al., 2014; Gao et al., 2023; Blalock et al., 2020). Applying SINDy unit has the following benefits: (a) The SINDy-function library contains frequently used functions in physics modeling (e.g. polynomials and Fourier series). (b) With sparse system identification, the neural network is more likely to identify governing physics, which is fundamentally important for extrapolation and long-term stability.

**Latent space regularization via ensemble SINDy** We first note that we deviate from the original SHRED architecture by using a GRU as opposed to an LSTM. This choice was made because we generally found that GRU provides a smoother latent space. Now, recall that our goal is to find a SHRED model with a latent state that is within the SINDy-class functional. However, the initial latent representation from SHRED does not follow the SINDy-based ODE structure at all. On the one hand, if we naively apply SINDy to the initial latent representation, the discovery is unlikely to fit the latent representation trajectory. On the other hand, if we directly replace the GRU unit to SINDy and force the latent space to follow the discovered SINDy model, it might lose information that is important to reconstruction the entire spatial domain. Therefore, it is important to let the two latent spaces align progressively.

In Algorithm 1, we describe our training procedure that allows the two trajectories to progressively align with each other. To further ensure a gradual adaptation and avoid over-regularization, we introduce ensemble SINDy units with varying levels of sparsity constraints, which ranges the effect from promoting a full model (all terms in the library are active) to a null model (where no dynamics are represented). From the initial latent representation $\mathbf{z}_{1:t}^{\text{iter }0}$ from SHRED, the SINDy model first provides an initial estimate of ensemble SINDy coefficients $\{\hat{\Xi}_0^i\}_{i=b}^B$. Then, the parameters of SHRED will be updated towards the dynamics simulated by $\{\hat{\Xi}_0\}_{i=b}^B$, which generates a new latent representation trajectory $\mathbf{z}_{1:t}^{\text{iter }1}$. We iterate this procedure and jointly optimize the following loss function to let the SHRED latent representation trajectory approximate the SINDy generated trajectory:

$$\mathcal{L} = \left\| \boldsymbol{X}_t - f_{\theta_D}(f_{\theta_{\text{GRU}}}(\boldsymbol{X}_{t-L:t}^{\mathcal{S}})) \right\|_2^2 + \sum_{i=1}^{B} \left\| \boldsymbol{Z}_t^{\text{GRU}} - \left( \boldsymbol{Z}_{t-1}^{\text{GRU}} + \sum_{j=0}^{k-1} \Theta(\boldsymbol{Z}_{t-1+jh})\Xi^{(i)} h \right) \right\|_2^2 + \lambda \|\Xi\|_0 , \tag{4}$$

where $\boldsymbol{Z}_{t-1+ih} = \boldsymbol{Z}_t + \Theta(\boldsymbol{Z}_{t-1+(i-1)h})\Xi h$, $\boldsymbol{Z}_{t-1} = \boldsymbol{Z}_{t-1}^{\text{GRU}}$, and $h = \frac{\Delta t}{k}$.

## 4 Experiment

In the following, we perform case studies across a range of scientific and engineering problems.

**Algorithm 1** Latent state space regularization via SINDy

**Input:** input $\boldsymbol{X}^{\mathcal{S}}_{t-L:t+1}$, $\boldsymbol{X}_t$, SINDy library $\Theta(\cdot)$, timestep $\Delta t$.

**function** LATENTSPACESINDY($\boldsymbol{X}^{\mathcal{S}}_{t-L:t+1}, \boldsymbol{X}_{t+1}, \Delta t$)
  **for** i in $0, 1, \cdots, n-1$: **do**
    $\boldsymbol{Z}_t,\ \boldsymbol{Z}_{t+1} = f_{\theta_{\text{GRU}}}(\boldsymbol{X}^{\mathcal{S}}_{t-L:t}),\ f_{\theta_{\text{GRU}}}(\boldsymbol{X}^{\mathcal{S}}_{t-L+1:t+1})$;
    **for** j in $(0, 1, \frac{\Delta t}{k})$: **do** ▷ SINDy forward simulation
      $\boldsymbol{Z}^{\text{SINDy}}_{t+\frac{j+1}{k}\Delta t} = \boldsymbol{Z}^{\text{SINDy}}_{t+\frac{j}{k}\Delta t} + \Theta(\boldsymbol{Z}^{\text{SINDy}}_{t+\frac{j}{k}\Delta t})\Xi\Delta t$
    **end for**
    $\hat{\boldsymbol{X}}_{t+1} = f_{\theta_D}(\boldsymbol{Z}_{t+1})$ ▷ SHRED reconstruction
    $\theta_{\text{GRU}}, \Xi, \theta_D = \arg\min_{\theta_{\text{GRU}},\Xi,\theta_D} \left\|\mathbf{X}_{t+1} - \hat{\boldsymbol{X}}_{t+1}\right\|_2^2 + \left\|\mathbf{Z}^{\text{GRU}}_{t+1} - \boldsymbol{Z}^{\text{SINDy}}_{t+1}\right\|_2^2 + \lambda \left\|\Xi\right\|_0$
    **if** $i \mod 100 = 0$ **then**
      $\Xi[|\Xi| < \text{threshold}] = 0$
    **end if**
  **end for** ▷ Train until converges
**end function**

**Sea-surface temperature** The first example we consider is that of global sea-surface temperature. The SST data contains 1,400 weekly snapshots of the weekly mean sea surface temperature from 1992 to 2019 reported by NOAA (Reynolds et al., 2002). The data is represented by a $180 \times 360$ grid, of which 44,219 grid points correspond to sea-surface locations. We standardize the data with its own min and max, which transforms the sensor measurements to within the numerical range of $(0, 1)$. We randomly select 250 sensors from the possible 44,219 locations and set the lag parameter to 52 weeks. The inclusion of 250 sensors is a substantial deviation from previous work with SHRED in which far fewer sensors were used Williams et al. (2024). However, we found greater robustness in the application of E-SINDy to the learned latent state when more sensors were utilized. Thus, for each input-output pair, the input consists of the 52-week trajectories of the selected sensors, while the output is a single temperature field across all 44,219 spatial locations. SINDy-SHRED aims to reconstruct the entire sea surface temperature locations from these randomly selected sparse sensor trajectories. We include the details of the experimental settings of SINDy-SHRED in the Appendix C.1. From the discovered coordinate system, we define the representation of physics - the latent hidden state space - to be $(z_1, z_2, z_3)$. The dynamics progresses forward via the following set of equations:

$$\begin{cases} \dot{z}_1 &= 4.68 z_2 - 2.37 z_3, \\ \dot{z}_2 &= -3.10 z_1 + 3.25 z_3, \\ \dot{z}_3 &= 2.72 z_1 - 5.55 z_2. \end{cases} \tag{5}$$

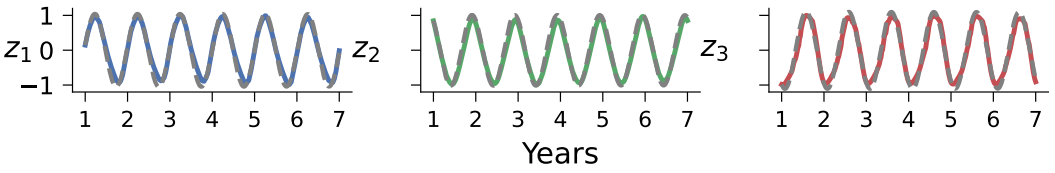



Figure 3: Extrapolation of latent representation in SINDy-SHRED from the discovered dynamical system for SST. Colored: true latent representation. Grey: SINDy extrapolation.

The discovery of a linear system describing the evolution of the latent state is in line with prior work on SST data De Bézenac et al. (2019) in which it as assumed that the underlying physics is an advection-diffusion PDE. In Fig. 3 (a) we further present the accuracy of this discovered system by forward simulating the system from an initial condition for a total of 27 years (c.f. Fig. 15). It is observable how the discovered law is close to the true evolution of latent hidden states and, critically, there appears to be minimal phase slipping. Extrapolating the latent state space via forward integration, we can apply the shallow decoder to return forecasts of the high-dimensional data. Doing

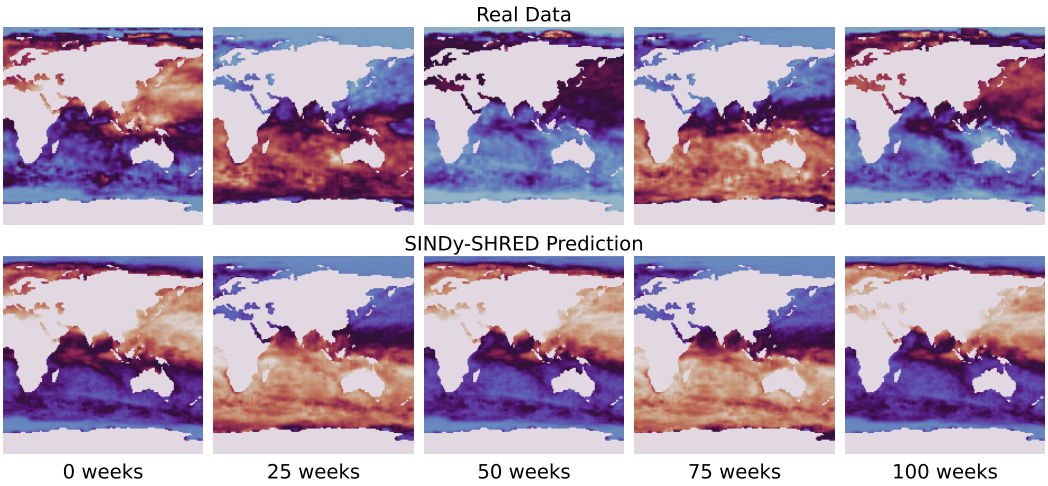



Figure 4: Long-term global sea-surface temperature prediction via SINDy-SHRED from week 0 to week 100. We crop the global temperature map for better visualization.

so, we find an averaged MSE error of $0.57 \pm 0.10°C$ for all prediction lengths in the test dataset. In Fig. 4, we show SINDy-SHRED produces stable long-term predictions for SST data. We further include Fig. 16 to demonstrate the extrapolation of each sensor. The sensor level prediction is based on the global prediction of the future frame, and we visualize the signal trajectory of specific sensor locations. We find SINDy-SHRED is robust for out-of-distribution sensors, though its extrapolation may not accurately capture anomalous events.

**3D Atmospheric ozone concentration** The atmospheric ozone concentration dataset (Bey et al., 2001) contains a one-year simulation of the evolution of an ensemble of interacting chemical species through a transport operator using GEOS-Chem. The simulation contains 1,456 temporal samples with a timestep of 6 hours over one year for 99,360 (46 by 72 by 30) spatial locations (latitude, longitude, elevation). The data presented in this work has compressed by performing an SVD and retaining only the first 50 POD modes. As with the SST data, we standardize the data within the range of $(0, 1)$ and randomly select and fix 3 sensors out of 99,360 spatial locations $(0.5\%)$. We include the details of the experimental settings of SINDy-SHRED in the Appendix C.2. The converged latent representation presents the following SINDy model:

$$\begin{cases} \dot{z}_1 &= -0.002 - 0.013z_2 + 0.007z_3, \\ \dot{z}_2 &= -0.001z_1 + 0.004z_2 - 0.008z_3, \\ \dot{z}_3 &= 0.002 + 0.012z_2 - 0.005z_3. \end{cases} \tag{6}$$

The identified governing physics is close to a linear system with constant terms for damping. Unlike traditional architectures for similar problems, which may include expensive 3D convolution, SINDy-SHRED provides an efficient way of training, taking about half an hour. Although the quantity of data is insufficient to perform long term-predictions, SINDy-SHRED still exhibits interesting behavior for a longer-term extrapolation which converges to the fixed point at $\mathbf{0}$ (as shown in Fig. 17). From the extrapolation of the latent state space, the shallow decoder prediction has an averaged MSE error of $1.5e^{-2}$. In Fig. 6, we visualize the shallow decoder prediction up to 14 weeks. In Fig. 18, we reconstruct the sensor-level predictions which demonstrate the details of the signal prediction. The observations are much noiser than the SST data, but SINDy-SHRED provides a smoothed extrapolation for the governing trends.

**GoPro physics data: flow over a cylinder** In this subsection, we demonstrate the performance of SINDy-SHRED on an example of so-called "GoPro physics modeling." The considered data is collected from a dyed water channel to visualize a flow over a cylinder (Albright, 2017). The Reynolds number is 171 in the experiment. The dataset contains 11 seconds of video taken at 30 frames per second (FPS). We manually perform data augmentation and repeat the latter part of the video once to increase the number of available training samples . We transfer the original RGB

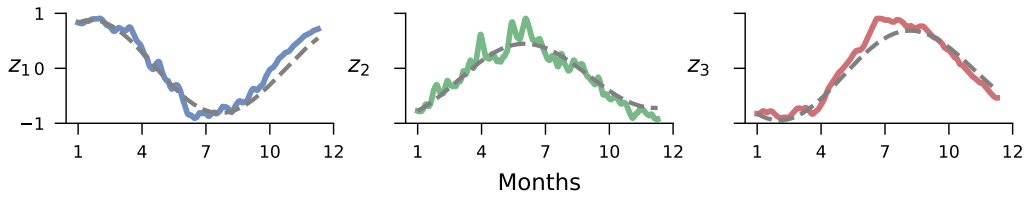



Figure 5: Extrapolation of latent representation in SINDy-SHRED from the discovered dynamical system for Ozone data. Colored: true latent representation. Grey: SINDy extrapolation.

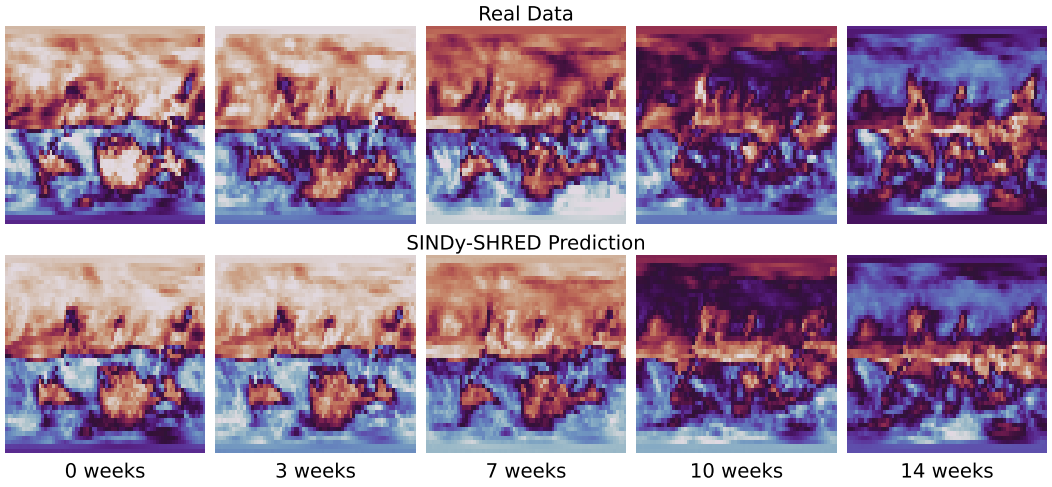



Figure 6: Long-term global Ozone data prediction via SINDy-SHRED with elevation 0 from week 0 to week 14.

channel to gray scale and remove the background by subtracting the mean of all frames. After the prior processing step, the video data has only one channel (gray) within the range $(0, 1)$ with a height of 400 pixels and a width of 1,000 pixels. We randomly select and fix 200 pixels as sensor measurements from the entire $400,000$ space, which is equivalent to only $0.05\%$ of the data. We set the lag parameter to 60 frames. We include the details of the experimental settings of SINDy-SHRED in the Appendix C.4. We define the representation of the hidden latent state space as $(z_1, z_2, z_3, z_4)$. We discover the following dynamical system:

$$\begin{cases} \dot{z}_1 &= -0.69z_2 + 0.98z_3 - 0.40z_4, \\ \dot{z}_2 &= 1.00z_1 - 0.78z_3 1 - 0.31z_2 z_3^2, \\ \dot{z}_3 &= -1.029z_1 + 0.59z_2 + 0.41z_4. \\ \dot{z}_4 &= -0.26z_1^2 - 0.29z_2^2 z_3 - 0.39z_3^3. \end{cases} \tag{7}$$

Compared to the systems discovered in all previous examples, the flow over a cylinder model is much more complex with significant nonlinear interactions. In Eqn. 7, we find that $z_1$ and $z_3$ behave like a governing mode of the turbulence swing; $z_2$ and $z_4$ further depict more detailed nonlinear effects. We also present the result of extrapolating this learned representation. We generate the trajectory from the initial condition at time point 0 and perform forward integration for extrapolation. As shown in Fig. 7, the learned ODE closely follows the dynamics of $z_1$ and $z_3$ up to 7 seconds (210 timesteps); $z_2$ and $z_4$ also have close extrapolation up to 3 seconds.

This learned representation also nicely predicts the future frames in pixel space. The shallow decoder prediction has an averaged MSE error of $0.030$ (equivalently $3\%$) over the entire available trajectory. In Fig. 8, we observe that the autoregressively generated prediction frames closely follow the true data, and further in Fig. 21, we find that the predictions are still stable after 1,000 frames, which is out of the size of the original dataset. The sensor-level prediction in Fig. 20 further demonstrates the accuracy of reconstruction in detail.

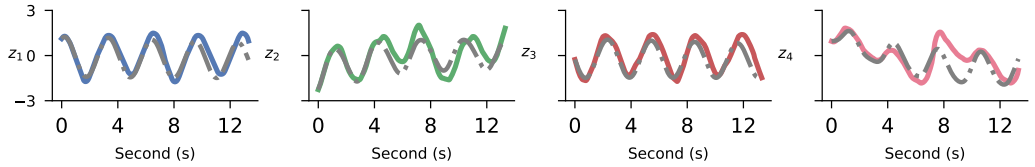



Figure 7: Extrapolation of latent representation in SINDy-SHRED from the discovered dynamical system for flow over a cylinder data. Colored: true latent representation. Grey: SINDy extrapolation.

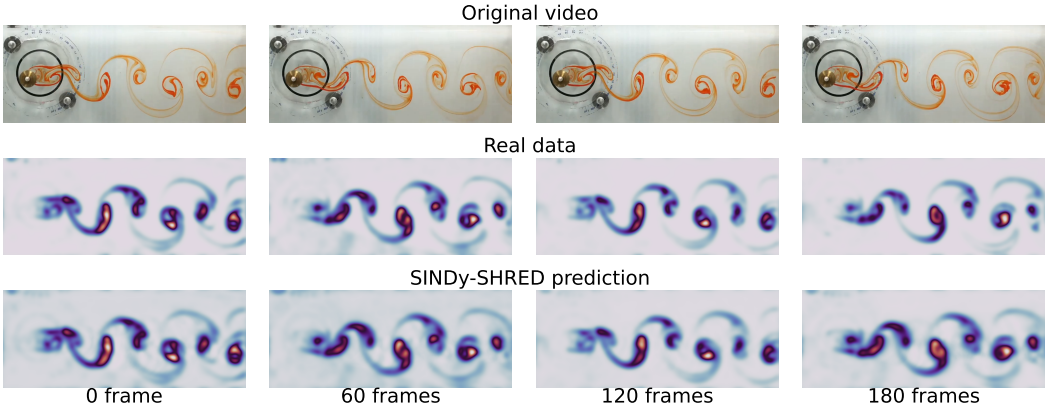



Figure 8: Long-term pixel space video prediction via SINDy-SHRED. We demonstrate the forward prediction outcome up to 180 frames.

**Baseline study: prediction of single shot real pendulum recording** In this subsection, we compare the performance of SINDy-SHRED to other popular existing learning algorithms. We perform the baseline study particularly on video data of a pendulum since many deep learning algorithms are hard to scale up to deal with large scientific data. In the following, we demonstrate the result of video prediction on the pendulum data using ResNet (He et al., 2016), convolutional LSTM (convLSTM) (Shi et al., 2015), and PredRNN (Wang et al., 2017), and SimVP (Gao et al., 2022). The pendulum in our experiment is not ideal and includes complex damping effects. We use a nail on the wall and place the rod (with a hole) on the nail. This creates complex friction, which slows the rod more when passing the lowest point due to the increased pressure caused by gravity. The full model we discovered from the video (as shown in Fig. 9) includes four terms:

$$\ddot{z} = 0.17\dot{z}^2 - 0.06\dot{z}^3 - 10.87\sin(z) + 0.48\sin(\dot{z}). \tag{8}$$

As shown in Table 1, SINDy-SHRED outperforms all baseline methods for total error and long-term predictions. Generally, all baseline deep learning methods perform well for short-term forecasting, but the error quickly accumulatesfor longer-term predictions. This is also observable from the prediction in the pixel space as shown in Fig. 10. SINDy-SHRED is the only method that does not produce collapsed longer-term predictions. In Fig. 22, the sensor level prediction also demonstrates the robustness of the SINDy-SHRED prediction. PredRNN is the second best method as measured by the total error. However, PredRNN is expensive in computation which includes a complex forward pass with an increased number of parameters. It is also notable that the prediction of PredRNN collapses after 120 frames, after which only an averaged frame over the entire trajectory is predicted. ConvLSTM has a relatively better result in terms of generation, but the long-term prediction is still inferior compared to SINDy-SHRED. Additionally, we note that 2D convolution is much more computationally expensive. For larger spatiotemporal domains (e.g. the SST example and 3D ozone data), the computational complexity of convolution will scale up very quickly, which makes the algorithm impractical to execute. Similar computational issues will occur for diffusion models and generative models, which is likely to be impractical to compute, and unstable for longer-term predictions. In summary, we observe that SINDy-SHRED is not only a more accurate long-term model, but is also faster to execute and smaller in size.

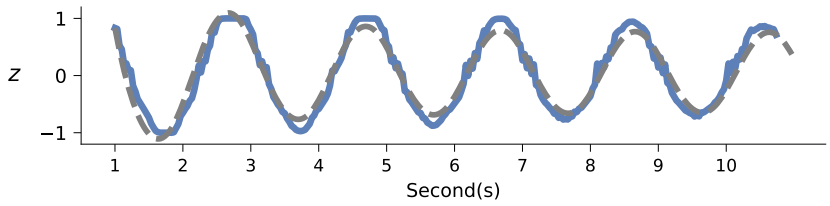



Figure 9: Extrapolation of latent representation in SINDy-SHRED from the discovered dynamical system for the pendulum moving data. Blue: true latent representation. Grey: SINDy extrapolation.

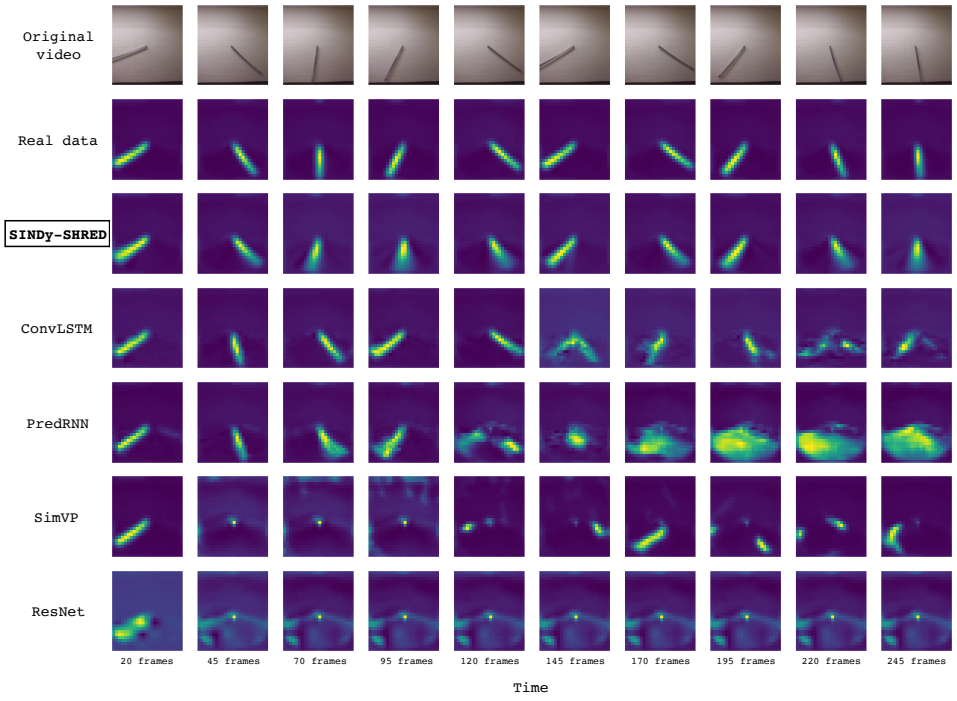



Figure 10: The pendulum video generation outcome from ResNet, SimVP, ConvLSTM, PredRNN, and SINDy-SHRED from frame 20 to frame 245.

| Models | Params # | Training time | $T = [0, 100]$ | $T = [100, 200]$ | $T = [200, 275]$ | Total |
|---|---|---|---|---|---|---|
| ResNet (He et al., 2016) | 2.7M | 24 mins | $2.08 \times 10^{-2}$ | $1.88 \times 10^{-2}$ | $2.05 \times 10^{-2}$ | $2.00 \times 10^{-2}$ |
| SimVP (Gao et al., 2022) | 460K | 30 mins | $2.29 \times 10^{-2}$ | $2.47 \times 10^{-2}$ | $2.83 \times 10^{-2}$ | $2.53 \times 10^{-2}$ |
| PredRNN (Wang et al., 2017) | 444K | 178 mins | $1.02 \times 10^{-2}$ | $1.79 \times 10^{-2}$ | $1.69 \times 10^{-2}$ | $1.48 \times 10^{-2}$ |
| ConvLSTM (Shi et al., 2015) | 260K | 100 mins | $\mathbf{9.24 \times 10^{-3}}$ | $1.86 \times 10^{-2}$ | $1.99 \times 10^{-2}$ | $1.55 \times 10^{-2}$ |
| **SINDy-SHRED*** | **44K** | **17 mins** | $1.70 \times 10^{-2}$ | $\mathbf{9.36 \times 10^{-3}}$ | $\mathbf{5.31 \times 10^{-3}}$ | $\mathbf{1.05 \times 10^{-2}}$ |

Table 1: Comparison table of SINDy-SHRED to baseline methods for parameter size, training time, and mean-squared error over different prediction horizon.

## 5 Conclusion

In this paper, we present SINDy-SHRED, which jointly performs the discovery of coordinate systems and governing equations with low computational cost and strong predictive power. Through experiments, we show that our method can produce robust and accurate long-term predictions for a variety of complex problems, including global sea-surface temperature, 3D atmospheric ozone concentration, flow over a cylinder, and a moving pendulum. SINDy-SHRED achieves state-of-the-art performance in long-term autoregressive video prediction, outperforming ConvLSTM, PredRNN, ResNet, and SimVP with the lowest computational cost and training time.

## REFERENCES


Jacob Albright. Flow visualization in a water channel, 2017. URL https://www.youtube.com/watch?v=30_aADFVL9M. YouTube video.

Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Peter L Bartlett, Sanjeev R Kulkarni, and S Eli Posner. Covering numbers for real-valued function classes. *IEEE transactions on information theory*, 43(5):1721–1724, 1997.

Isabelle Bey, Daniel J Jacob, Robert M Yantosca, Jennifer A Logan, Brendan D Field, Arlene M Fiore, Qinbin Li, Honguy Y Liu, Loretta J Mickley, and Martin G Schultz. Global modeling of tropospheric chemistry with assimilated meteorology: Model description and evaluation. *Journal of Geophysical Research: Atmospheres*, 106(D19):23073–23095, 2001.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? *Proceedings of machine learning and systems*, 2:129–146, 2020.

Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

Salva Rühling Cachay, Brian Henn, Oliver Watt-Meyer, Christopher S Bretherton, and Rose Yu. Probabilistic emulation of a global climate model with spherical dyffusion. *arXiv preprint arXiv:2406.14798*, 2024.

Claudio Canuto, M Yousuff Hussaini, Alfio Quarteroni, and Thomas A Zang. *Spectral methods: evolution to complex geometries and applications to fluid dynamics*. Springer Science & Business Media, 2007.

Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116 (45):22445–22451, 2019.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, et al. Symbolic discovery of optimization algorithms. *Advances in neural information processing systems*, 36, 2024.

Sheng Cheng, Deqian Kong, Jianwen Xie, Kookjin Lee, Ying Nian Wu, and Yezhou Yang. Latent space energy-based neural odes. *arXiv preprint arXiv:2409.03845*, 2024.

Paolo Conti, Giorgio Gobat, Stefania Fresca, Andrea Manzoni, and Attilio Frangi. Reduced order modeling of parametrized systems through autoencoders and sindy approach: continuation of periodic solutions. *Computer Methods in Applied Mechanics and Engineering*, 411:116072, 2023.

Emmanuel De Bézenac, Arthur Pajot, and Patrick Gallinari. Deep learning for physical processes: Incorporating prior scientific knowledge. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124009, 2019.

Megan R Ebers, Jan P Williams, Katherine M Steele, and J Nathan Kutz. Leveraging arbitrary mobile sensor trajectories with shallow recurrent decoder networks for full-state reconstruction. *IEEE Access*, 2024.

Nicola Farenga, Stefania Fresca, Simone Brivio, and Andrea Manzoni. On latent dynamics learning in nonlinear reduced order modeling. *arXiv preprint arXiv:2408.15183*, 2024.

Urban Fasel, J Nathan Kutz, Bingni W Brunton, and Steven L Brunton. Ensemble-sindy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proceedings of the Royal Society A*, 478(2260):20210904, 2022.

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *Advances in neural information processing systems*, 29, 2016.

Kai Fukami, Takaaki Murata, Kai Zhang, and Koji Fukagata. Sparse identification of nonlinear dynamics with low-dimensionalized flow representations. *Journal of Fluid Mechanics*, 926:A10, 2021.

L Gao, Urban Fasel, Steven L Brunton, and J Nathan Kutz. Convergence of uncertainty estimates in ensemble and bayesian sparse model discovery. *arXiv preprint arXiv:2301.12649*, 2023.

Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z Li. Simvp: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3170–3180, 2022.

Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11474–11484, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Philipp Holl, Vladlen Koltun, and Nils Thuerey. Learning to control pdes with differentiable physics. *arXiv preprint arXiv:2001.07457*, 2020.

Rama Kandukuri, Jan Achterhold, Michael Moeller, and Joerg Stueckler. Learning to identify physical parameters from video using differentiable physics. In *DAGM German conference on pattern recognition*, pp. 44–57. Springer, 2020.

Yuehaw Khoo, Jianfeng Lu, and Lexing Ying. Solving parametric pde problems with artificial neural networks. *European Journal of Applied Mathematics*, 32(3):421–435, 2021.

J Nathan Kutz, Maryam Reza, Farbod Faraji, and Aaron Knoll. Shallow recurrent decoder for reduced order modeling of plasma dynamics. *arXiv preprint arXiv:2405.11955*, 2024.

Yunzhu Li, Toru Lin, Kexin Yi, Daniel Bear, Daniel Yamins, Jiajun Wu, Joshua Tenenbaum, and Antonio Torralba. Visual grounding of learned physical models. In *International conference on machine learning*, pp. 5927–5936. PMLR, 2020a.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020b.

Guang Lin, Christian Moya, and Zecheng Zhang. Accelerated replica exchange stochastic gradient langevin diffusion enhanced bayesian deeponet for solving noisy parametric pdes. *arXiv preprint arXiv:2111.02484*, 2021.

Shaowei Liu, Zhongzheng Ren, Saurabh Gupta, and Shenlong Wang. Physgen: Rigid-body physics-grounded image-to-video generation.

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.

Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):1–10, 2018.

L Mars Gao and J Nathan Kutz. Bayesian autoencoders for data-driven discovery of coordinates, governing equations and fundamental constants. *Proceedings of the Royal Society A*, 480(2286): 20230506, 2024.

Daniel A Messenger and David M Bortz. Weak sindy for partial differential equations. *Journal of Computational Physics*, 443:110525, 2021.

Mehdi Mirza. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Damian Mrowca, Chengxu Zhuang, Elias Wang, Nick Haber, Li F Fei-Fei, Josh Tenenbaum, and Daniel L Yamins. Flexible neural representation for physics prediction. *Advances in neural information processing systems*, 31, 2018.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pp. 5301–5310. PMLR, 2019.

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.

Maria Refinetti and Sebastian Goldt. The dynamics of representation learning in shallow, non-linear autoencoders. In *International Conference on Machine Learning*, pp. 18499–18519. PMLR, 2022.

Richard W Reynolds, Nick A Rayner, Thomas M Smith, Diane C Stokes, and Wanqiu Wang. An improved in situ and satellite sst analysis for climate. *Journal of climate*, 15(13):1609–1625, 2002.

Stefano Riva, Carolina Introini, Antonio Cammi, and J Nathan Kutz. Robust state estimation from partial out-core measurements with shallow recurrent decoder for nuclear reactors. *arXiv preprint arXiv:2409.12550*, 2024.

Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science advances*, 3(4):e1602614, 2017.

Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. Graph networks as learnable physics engines for inference and control. In *International conference on machine learning*, pp. 4470–4479. PMLR, 2018.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34:1415–1428, 2021.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. *Advances in neural information processing systems*, 30, 2017.

Jan P Williams, Olivia Zahn, and J Nathan Kutz. Sensing with shallow recurrent decoder networks. *Proceedings of the Royal Society A*, 480(2298):20240054, 2024.

Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. *Advances in neural information processing systems*, 28, 2015.

Jiajun Wu, Erika Lu, Pushmeet Kohli, Bill Freeman, and Josh Tenenbaum. Learning to see physics via visual de-animation. *Advances in neural information processing systems*, 30, 2017.

Tailin Wu, Takashi Maruyama, and Jure Leskovec. Learning to accelerate partial differential equations via latent global evolution. *Advances in Neural Information Processing Systems*, 35:2240–2253, 2022.

Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4389–4398, 2024.

Rose Yu and Rui Wang. Learning dynamical systems from data: An introduction to physics-guided deep learning. *Proceedings of the National Academy of Sciences*, 121(27):e2311808121, 2024.

Zemin Zheng, Yingying Fan, and Jinchi Lv. High dimensional thresholded regression and shrinkage effect. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76(3):627–649, 2014.

Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

## A CHALLENGES IN ROLLING OUT NEURAL NETWORKS FOR FITTING A SIMPLE SINE FUNCTION

In the following example, we consider a simple use case in which we fit a simple sine function using recurrent neural networks. Surprisingly, extrapolating a simple sine function can be highly nontrivial for neural networks.

We implement a GRU network in the following. The GRU network consists of an input layer, three stacked GRU layers with size 500, and a fully connected output layer. We employ the Adam optimizer with a learning rate of 0.001 and used the mean squared error (MSE) as the loss function. We train the GRU network with 150 epochs with a batch size of 1. The input sequences are made up of 50 time steps, normalized to the range [0, 1].

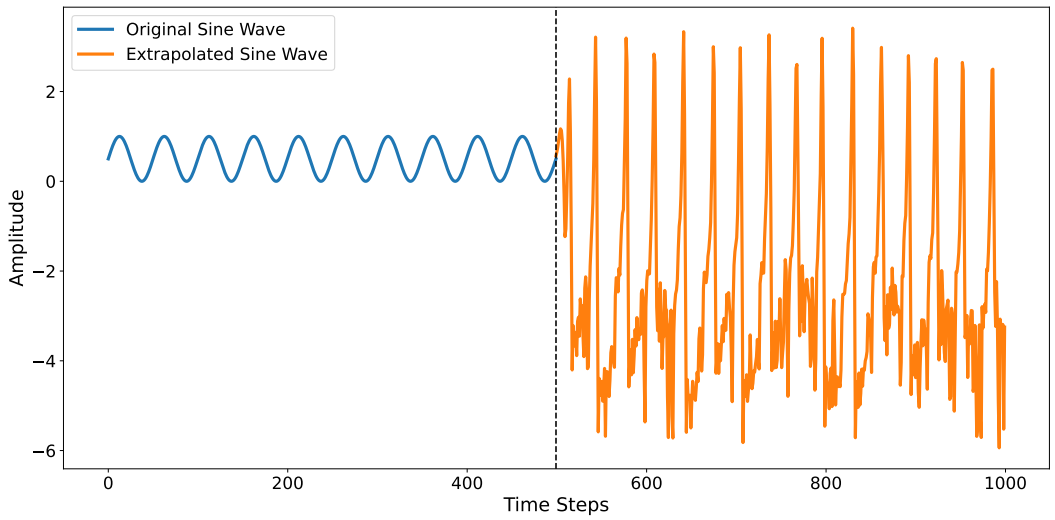



Figure 11: Fitting the sine function using a GRU network could be challenging.

Even though the training loss is near-optimal with $1e^{-7}$. However, the extrapolation is very poor. We note here that by carefully specifying the GRU network and making it a mostly linear unit (within the SINDy-class functional), the extrapolation result could be reasonable. But, the general setting of GRU networks, as shown in Fig. 11, fails to predict well. This motivating example demonstrates the fact that knowledge with physical meanings, e.g. the sine function here, is hard for neural networks to approximate accurately in extrapolation. Therefore, this demonstrates the necessity of integrating well-characterized ODE structures with non-linear functions into deep learning models, as they are important for accurate learning and prediction of physics.

## B QUALITATIVE RESULT ON THE ERROR BOUNDS

We further establish the statistical foundation for dynamical system learning. When the underlying dynamical system can be closely described by a linear combination of the library of functions, obtaining a "governing equation" will have huge benefits for long-term extrapolation. Due to the nature of forward integration,error accumulates rapidly making an approximate system undesirable for extrapolation. In the following, we formalize this statement by analyzing the Rademacher complexity of SINDy-class and neural networks functional.

The system we wish to study has the form that

$$\dot{x} = f(x), \tag{9}$$

which is an ODE describing the trajectory of a dynamical system in a learned latent space.

Suppose that the dynamical system has the form $\dot{\mathbf{x}} = f(\mathbf{x})$, and we have measurements of $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_t, \mathbf{x}_{t+1}, ..., \mathbf{x}_T\}$ with time gap $\Delta t$. We first define the SINDy-class functional:

$$\mathcal{F}_{\text{SINDy}} := \{\Theta\xi : \xi \in \mathbb{R}^p\}, \tag{10}$$

where all $f \in \mathcal{F}_{\text{SINDy}}$ are functions of a convex (linear) combination of functions in $\Theta(\cdot)$, and all functions in the library are within $L^2(P)$. An example of $\Theta(x)$ is to have $[x, x^2, x^3, x^4, \sin(x), \cos(x)]$, and this could represent a dynamical system with the following form

$$\dot{f}(x) = a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 \sin(x) + a_6 \cos(x), \tag{11}$$

where $a_1, a_2, ..., a_6$ are constants.

Then, we consider the target function $f$ to be within $\mathcal{F}_0$.

$$\mathcal{F}_0 := \{f_0 : \sup_{x \in X} |f_0(x) - f(x)| < \epsilon,\ f \in \mathcal{F}_{\text{SINDy}}\}, \tag{12}$$

where $f_0(\cdot)$ is a function within convex (linear) combination of functions in $\Theta(\cdot)$. We further assume that $\mathcal{F}_0$ is $L$-Lipschitz.

We note that $\mathcal{F}_0$ is a wide class of functions. Since $\Theta(\cdot)$ covers polynomials and the Fourier series, the functional class $\mathcal{F}_0$ could model the governing effect for all differentiable functions from Taylor's approximation. Then, we consider the two-layer ReLU functional class $\mathcal{F}_{\text{ReLU}}$ to be

$$\mathcal{F}_{\text{ReLU}} := \{\mathbf{x} \mapsto \theta_2 \sigma(\theta_1 \mathbf{x})\}, \tag{13}$$

where $\theta_1 \in \mathbb{R}^{p \times d}$, $\theta_2 \in \mathbb{R}^{d \times p}$, and $\sigma(x) = \max(0, x)$ represents the ReLU function. In the following, we additionally require the following conditions on the input data that

$$\|\mathbf{x}\|_\infty \leq 1, \quad \|\mathbf{x}\|_2 \leq \sqrt{d}, \quad |f(\mathbf{x})| \leq \frac{1}{2}. \tag{14}$$

Define the error of dynamical system simulation as

$$\mathcal{E}(\mathbf{x}, t) = \mathbf{x}(t) - \hat{\mathbf{x}}(t). \tag{15}$$

**Theorem 1.** *Suppose we have a target function $f_0 \in \mathcal{F}_0$ and the approximation is $\hat{f}$. Suppose that the approximation error of $\hat{f}$ is up to $\epsilon$ as $\sup_x |f_0(x) - \hat{f}(x)| < \epsilon$. The generalization error is up to $E_{gen}$ with probability $1 - \delta$. Then, simulating the system up to time $T$ will reach the error with rate $\mathcal{O}\left(\frac{\epsilon + E_{gen}}{L} \exp(LT)\right)$.*

*Proof.* The error $\mathcal{E}(\cdot)$ is defined as

$$\mathcal{E}(\mathbf{x}, t) = \mathbf{x}(t) - \hat{\mathbf{x}}(t) \tag{16}$$

By taking the derivative with respect to $t$ on both sides, we get

$$\dot{\mathcal{E}}(x, t) = f_0(x) - \hat{f}(\hat{x}) \tag{17}$$

$$= [f_0(x) - f_0(\hat{x})] + \underbrace{[f_0(\hat{x}) - \hat{f}(\hat{x})]}_{\text{generalization error}} \tag{18}$$

The generalization error includes the regret and the approximation error. We first observe that $\frac{d}{dt}||\mathcal{E}(x, t)|| \leq ||\frac{d}{dt}\mathcal{E}(x, t)||$ from Cauchy-Schwarz. Considering the norm, due to the triangle inequality,

$$\frac{d}{dt}\|\mathbf{x}(t) - \hat{\mathbf{x}}(t)\| \leq \|f_0(\mathbf{x}) - f_0(\hat{\mathbf{x}})\| + \left\|f_0(\hat{\mathbf{x}}) - \hat{f}(\hat{\mathbf{x}})\right\| \tag{19}$$

Suppose $\hat{\mathbf{x}}$ is still within the input domain, from the prior result, we know with probability $1 - \delta$ we have

$$\sup_{\mathbf{x}} \left\|f(\mathbf{x}) - \hat{f}(\mathbf{x})\right\| \leq \epsilon + E_{\text{gen}}. \tag{20}$$

Then, since the functional class $\mathcal{F}_0$ is $L$-Lipschitz, we have

$$\frac{d}{dt}\left\|\mathbf{x}(t)-\hat{\mathbf{x}}(t)\right\| \leq L\left\|\mathbf{x}-\hat{\mathbf{x}}\right\| + \epsilon + E_{\text{gen}}. \tag{21}$$

By applying the differential form of Grönwall's inequality, we see

$$\mathcal{E}(x,t) \leq \frac{\epsilon + E_{\text{gen}}}{L}\left(e^{LT}-1\right), \tag{22}$$

which is the upper bound of the error. □

By observing Thm. 1, we found the error has rate $\mathcal{O}\left(\frac{\epsilon+E_{\text{gen}}}{L}e^{LT}\right)$. Therefore, as $T$ increases, the error will be magnified with a factor of $e^T$. This type of error accumulation will create problems for neural networks. By increasing the power of approximation (having a smaller $\epsilon$), the neural network will lose the ability to generalize, leading to a large $E_{\text{gen}}$. Meanwhile, when $E_{\text{gen}}$ is large, the extrapolation may be out of the input domain $\mathcal{X}$, resulting in unstable extrapolations of the dynamical system. However, the SINDy-class functional is inherently designed to avoid this issue. We analyze the error for the SINDy-class functional in the following.

**Theorem 2.** *The SINDy-class functional, with probability $1-\delta$, will have an error with order*

$$\mathcal{E}(T) = \mathcal{O}\left(\frac{\epsilon + \sqrt{\frac{M_n}{n}}}{L}e^{LT}\right) \tag{23}$$

*Proof.* First, we note that by definition, the optimal solution $\xi_0$ can reach an error of $\epsilon$ pointwise. Then, we consider the empirical risk minimizer $\hat{\xi}$, and study the regret:

$$\text{Reg}(\hat{\xi}) = P\ell(\hat{\xi}) - P\ell(\xi_0). \tag{24}$$

In the task of regression, we know this $\ell(\cdot)$ is

$$\ell(x,\xi) = [f_0(x) - \Theta(x)\xi]^2 \tag{25}$$

Define $g \in \mathcal{G} := \{x \mapsto \ell(x,\xi) : \xi \in \Xi\}$. From the empirical process theory, we know

$$\begin{aligned}
\text{Reg}(\hat{\xi}) &\leq P\ell(\hat{\xi}) - P\ell(\xi_0) + \left[P_n\ell(\xi_0) - P_n\ell(\hat{\xi})\right] && (26)\\
&= (P_n - P)\left[\ell(\xi_0) - \ell(\hat{\xi})\right] && (27)\\
&\leq |(P_n - P)\ell(\xi_0)| + (P_n - P)\ell(\hat{\xi})| && (28)\\
&\leq 2\sup_{\xi\in\Xi}|(P_n - P)\ell(\xi)| = 2\sup_{g\in\mathcal{G}}|(P_n - P)g| = 2\left\|P_n - P\right\|_{\mathcal{G}} && (29)
\end{aligned}$$

So we need to study this functional class $\mathcal{G}$.

We expand the definition of $\mathcal{G} := \left\{x \mapsto [f_0(x) - \Theta(x)\xi]^2 : \xi \in \Xi\right\}$. Using the definition of $f_0$, we have $\mathcal{G} := \left\{x \mapsto [\Theta(x)(\xi_0 - \xi) + \eta(x)]^2 : \xi \in \Xi\right\}$ where $\eta(x)$ denotes some quantity that is upper bounded by $\eta > 0$.

We see from Markov's inequality that for $t > 0$ and some $a > 0$,

$$\begin{aligned}
\mathbb{P}\left(\text{Reg}(\hat{\xi}) > \left(\frac{M_n}{n}\right)^a t\right) &\leq \mathbb{P}\left(2\left\|P_n - P\right\|_{\mathcal{G}} > \left(\frac{M_n}{n}\right)^a t\right) && (30)\\
&\leq \frac{2\mathbb{E}\|P_n - P\|_{\mathcal{G}}}{\left(\frac{M_n}{n}\right)^a t} && (31)\\
&\leq \frac{2\mathbb{E}\|R_n\|_{\mathcal{G}}}{\left(\frac{M_n}{n}\right)^a t}, && (32)
\end{aligned}$$

where $\mathbb{E}\|R_n\|_{\mathcal{G}}$ denotes the Rademacher complexity. $R_n(g) = \frac{1}{n}\sum_{i=1}^{n}\epsilon_i g(x_i)$; $\epsilon_i$'s are independent Rademacher random variables Unif$(0,1)$, and $\|R_n\|_{\mathcal{G}} = \sup_{g\in\mathcal{G}}|R_n(g)|$.

We start from the case that $x$ is univariate. We assume that the data we have have the form $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ where $\mathbf{x}_i$ has a bounded domain and $\mathbf{y}_i$ is within range $[-\frac{1}{2}, \frac{1}{2}]$. Define the empirical risk minimizer $\xi_0 : w \mapsto \mathbb{E}_P[\mathbf{y} \mid \mathbf{x} = \mathbf{x}_i]$. The estimator $\hat{\xi}_n$ minimizes empirical MSE $\mathcal{R}_n : \xi \mapsto \frac{1}{n}\sum_{i=1}^{n}[\mathbf{y}_i - \xi(\mathbf{x}_i)]^2$. We denote $\Theta_M$ be the class of functions whose range falls in $[-\frac{1}{2}, \frac{1}{2}]$ with total variation not larger than $M$. The total variation is defined as a class of functions $h : [0,1] \mapsto \mathbb{R}$ that satisfies $\forall k \in \mathbb{N}$, and $\forall 0 = x_0 < x_1 < ... < x_{n-1} < x_k < 1$:

$$\sum_{i=1}^{n-1}|h(x_{i+1}) - h(x_i)| \leq M. \tag{33}$$

By plugging in the definition of $\mathcal{G}$, we find

$$\sum_{i=0}^{n-1}|g(x_{i+1}; f_0, \xi) - g(x_i; f_0, \xi)| \tag{34}$$

$$= \sum_{i=0}^{n-1}|(\Theta(x_{i+1})(\xi_0 - \xi) + \epsilon(x_{i+1}))^2 - (\Theta(x_i)(\xi_0 - \xi) + \epsilon(x_i))^2| \tag{35}$$

$$= \sum_{i=0}^{n-1}|(\xi - \xi_0)^2[\Theta(x_{i+1})^2 - \Theta(x_i)^2] + 2(\xi_0 - \xi)[\Theta(x_{i+1})\eta(x_{i+1}) - \Theta(x_i)\eta(x_i)] + [\eta(x_{i_1})^2 - \eta(x_i)^2]| \tag{36}$$

We see that the above quantity is bounded because $\xi$ has a bounded domain, $\Theta(\cdot)$ is Lipschitz within $[0,1]$, and $\epsilon(\cdot)$ is bounded by a constant. We note this constant as $M_n$, which corresponds to $M_n$ in Eq. 30.

Say $Q$ is a continuous distribution on $[0,1]$. The covering number of the functional class $\mathcal{G}$ and the covering number of $\Theta_M$ is connected through $\mathcal{N}(\epsilon, \mathcal{G}, L^2) \leq \mathcal{N}(\epsilon, \Theta_M, L^2)$.

Suppose that we have an $\epsilon/4$-cover of $\theta_M$ which contains $\{\theta_1, \theta_2, ..., \theta_n\}$. We see that $\{\ell(\theta_j)\}$ is an $\epsilon$-cover of $\mathcal{G}$ by

$$\|\ell(\theta) - \ell(\theta_j)\|^2_{\mathcal{L}^2(Q)} = \int[(y - \theta x)^2 - (y - \theta_j x)^2]^2 dQ(x) \tag{37}$$

$$= \int[(2y - \theta x - \theta_j x)(\theta x - \theta_j x)]^2 dQ(x) \tag{38}$$

$$\leq \sup_{x,y}(2y - \theta x - \theta_j x)^2 \int(\theta x - \theta_j x)^2 dQ(x) \tag{39}$$

$$[\{\theta_j\} \text{ is an } \frac{\epsilon}{4}\text{-cover}] \leq \epsilon^2. \tag{40}$$

Then, from Lemma 1, we know that $\mathcal{N}(\epsilon, \Theta_M, \mathcal{L}^2)$ is bounded by $\frac{13}{\epsilon}$. Since the Rademacher random variable is sub-Gaussian, we can control the Rademacher complexity further using the Dudley's entropy integral,

$$\mathbb{E}\|R_n\|_{\mathcal{G}} \leq 8n^{-1/2}\sup_{Q}\int_0^{\infty}\sqrt{\log\mathcal{N}(\epsilon, \Theta_{M_n}, \mathcal{L}^2(Q))}d\epsilon \tag{41}$$

$$[D := \sup_{x,y} d(x,y)] = 8n^{-1/2}\sup_{Q}\int_0^{D}\sqrt{\log\mathcal{N}(\epsilon, \Theta_{M_n}, \mathcal{L}^2(Q))}d\epsilon \tag{42}$$

$$\leq 8n^{-1/2}\sup_{Q}\int_0^{D}\sqrt{\log\mathcal{N}(\epsilon, \Theta_{M_n}, \mathcal{L}^2(Q))}d\epsilon \tag{43}$$

$$\leq \frac{CM_n}{\sqrt{n}}. \tag{44}$$

From a similar process, we can bound the multivariate input version by some other constant $C_p$ that bounds the norm of the possible parameters (i.e. $\|\xi\|_2 \leq C_p$). By applying the differential form of Grönwall's inequality, we see

$$\mathcal{E}(x,t) \leq \left(\frac{2CM_n}{L\sqrt{n}} + \epsilon\right)\left(e^{LT} - 1\right), \tag{45}$$

which is the upper bound of the error.

We note here that in theory it is possible to push $\mathcal{E}(x,t) = O_P\left(n^{-2/3}\right)$ under the condition $M_n = o(n)$, which is better than the current rate $O_P\left(n^{-1/2}\right)$. □

Here, we find that the generalization error is independent of $\epsilon$ for the SINDy-class functional. Therefore, the approximation-generalization tradeoff does not hurt SINDy when the target function is within $\mathcal{F}_0$.

Now, we see the proof for neural networks. There are two factors that might cause the generalization to be large. First, when the target functional to learn is very close to $\mathcal{F}_{\text{SINDy}}$ with small $\epsilon$. In this case, to obtain a better approximation, the network has to increase the size of the hidden layer. This will lead to an increase in terms of the Rademacher complexity and therefore the generalization error. In addition, the input data for neural networks are different. Unlike the ODE forward integration structure in SINDy, neural networks frequently use the previous $L$ data points as input to predict the value at $t_{L+1}$. This will lead to a larger input space, which will also increase the generalization error. We show the details of the theorem in the following.

**Theorem 3.** *If we use a function in $\mathcal{F}_{ReLU}$ to learn a dynamical system, we expect to have an error of up to*

$$\mathcal{E}(T) = \mathcal{O}\left(\epsilon e^{LT} + \frac{W_1 W_2}{\sqrt{n}}\sqrt{\log\left(\frac{\Lambda(d)}{\epsilon}\right)}e^{LT}\right), \tag{46}$$

*where $W_1, W_2, \Lambda(d)$ satisfies $\|\theta_1\|_2 \leq W_1$, $\|\theta_2\|_2 \leq W_2$, and $\int_{\mathbb{R}^d} \|\omega\|_2 |\hat{f}(\omega)| d\omega \leq \Lambda(d)$, where $\int \|\omega\|_2 |\hat{f}(\omega)| d\omega$.*

*Proof.* We know that to reach an approximation error of $\epsilon$ uniformly, from Barron's approximation theorem, we know:

$$\inf_{\hat{f}\in\mathcal{F}_{\text{ReLU}}} \sup_x |f_0(x) - \hat{f}(x)| \leq \frac{\Lambda(d)}{p}, \tag{47}$$

where $\Lambda(d)$ is the Barron constant and $p$ is the size of hidden layer. We know, under the same $\Lambda(d), W_1, W_2$, we need $p \geq \left(\frac{\Lambda(d)}{\epsilon}\right)^2 = \mathcal{O}\left(\frac{\Lambda(d)^2}{\epsilon^2}\right)$.

We can know the Radamacher complexity from classical result (Bartlett & Mendelson, 2002) (c.f. Thm. 18) is bounded by

$$\hat{\mathcal{R}}_n(\mathcal{F}) \leq \frac{CW_1W_2\sqrt{\log(p)}}{\sqrt{n}}, \tag{48}$$

for some constant $C$.

From the required precision for approximation, we know that the generalization error scales with

$$E_{\text{gen}} = \mathcal{O}\left(\frac{W_1W_2}{\sqrt{n}}\sqrt{\frac{\Lambda(d)}{\epsilon}}\right). \tag{49}$$

Using the Grönwall's inequality again, we see

$$\mathcal{E}(T) = \mathcal{O}\left(\epsilon e^{LT} + \frac{W_1W_2}{\sqrt{n}}\sqrt{\log\left(\frac{\Lambda(d)}{\epsilon}\right)}e^{LT}\right). \tag{50}$$

□

From Thm. 3, we see the dilemma of neural networks. The generalization error will increase as $\epsilon$ gets smaller. In this case, when $f_0$ can be closely represented via $\mathcal{F}_{\text{SINDy}}$ with $\epsilon \to 0$, the neural network will fail to perform reasonable extrapolations. In practice, regularization techniques can be applied to mitigate unstable predictions. However, this will reduce the accuracy of the approximation and make the theoretical behavior difficult to characterize.

Another way is to show the bound using (Bartlett & Mendelson, 2002) will incorporate a factor of $\mathcal{O}\left(\sqrt{d}\right)$. It is also possible to show the instability of neural networks due to the magnification of the input space by the factor of $L$ (length of the input trajectory).

### B.1 Technical lemmas

**Lemma 1.** *(Bartlett et al., 1997). For all $\epsilon < \frac{1}{12}$, we have*

$$(\log_2 e)\frac{1}{54\epsilon} \leq \log_2 \mathcal{N}(\epsilon, \mathcal{F}_1, \mathcal{L}^2) \leq \frac{13}{\epsilon}, \tag{51}$$

*where $\mathcal{F}_1$ has bounded variation and takes value from $[0,1]$ to $[-1/2, 1/2]$. $\mathcal{L}^2$ denotes the norm $||f||_{\mathcal{L}^1(P)} = \int |f(x)| dP(x)$.*

**Definition 1.** *A function $g : S \times S \mapsto [0, \infty)$ is called a pseudometric on $S$ if:*

*(a) $d(x,x) = 0,\ \forall x \in S$.*

*(b) $d(x,y) = d(y,x),\ \forall x, y \in S$.*

*(c) $d(x,z) \leq d(x,y) + d(y,z),\ \forall x, y, z \in S$*

We denote a pseudometric space by $(S, d)$.

**Definition 2.** *Let $(S,d)$ be a pseudometric space and let $T \subseteq S$. $T_1 \subseteq T$ is called an $\epsilon$-cover if $\forall \theta \in T$, there is a $\theta_1 \in T_1$ s.t. $d(\theta, \theta_1) \leq \epsilon$.*

*And the $\epsilon$-covering number of $T$ is defined as*

$$\mathcal{N}(\epsilon, T, d) = \min\{|T_1| : T_1 \text{ is an } \epsilon\text{-cover of } T\}, \tag{52}$$

*where $|T_1|$ denotes the cardinality of set $T_1$.*

**Lemma 2.** *(Barron, 1993) To reach an approximation error of $\epsilon$, fixing the norm of network parameters, we need to satisfy*

$$p \leq \frac{C \left\|f\right\|_{\mathcal{B}}^2 W^2}{\epsilon^2}. \tag{53}$$

## C Experimental details

### C.1 Sea-surface temperature data

For the SST data in SINDy-SHRED, we set the latent dimension to 3 because we observe only minor impacts on the reconstruction accuracy when the latent dimension is $\geq 4$. We include 2 stacked GRU layers and consider the , and a two-layer ReLU decoder with 350 and 400 neurons. For the E-SINDy regularization, we set the polynomial order to be 3 and the ensemble number is 10. In the latent hidden-state forward simulation, we use Euler integration with $dt = \frac{1}{520}$, which will generate the prediction of next week via 10 forward integration steps. During training, we apply the AdamW optimizer with a learning rate of $1e^{-3}$ and a weight decay of $1e^{-2}$. The batch size is 128 with 1,000 training epochs. The thresholds for E-SINDy range uniformly from 0.1 to 1.0, and the thresholding procedure will be executed every 100 epochs. We use dropout to avoid overfitting with a dropout rate of 0.1. The training time is within 30 minutes from a single NVIDIA GeForce RTX 2080 Ti.

### C.2 3D atmospheric ozone concentration

For the ozone data, we set the lag parameter is set to 100. Thus, for each input-output pair, the input consists of the 62.5 day measurements of the selected sensors, while the output is the measurement

across the entire 3D domain. In SINDy-SHRED, we follow the same network architecture as in the SST experiment. We set $dt = 0.025$, and the thresholds for E-SINDy range uniformly from 0.015 to 0.15. The thresholding procedure will be executed every 300 epochs, and we apply AdamW optimizer with learning rate $1e^{-3}$.

### C.3 Flow over a cylinder

In the flow over a cylinder experiment, we follow the same settings as in the prior experiments and select the latent dimension to be 4. The forward integration time step is set to $dt = \frac{1}{300}$ corresponding to the frame rate of 30 FPS. We set the batch size at $64$ and the learning rate to $5e^{-4}$. The thresholding procedure is executed every 300 epochs with thresholds ranging from $(1e^{-4}, 1e^{-3})$.

### C.4 Baseline experiment on pendulum

**Autoregressive training.** The raw pendulum data are collected from a 14-second GoPro recording. The raw data are present difficulties during training because of their high-dimensionality ($\mathbf{1}080 \times 960$), so we follow the same preprocessing procedure as in (Mars Gao & Nathan Kutz, 2024) to obtain a set of training data with 390 samples, width 24 and height 27. For most of the models, we apply autoregressive training to help the model achieve better long-term prediction capabilities. From the initial input $\{\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_L\}$ with lag $L$, the model autoregressively predicts the next frame $\hat{\boldsymbol{X}}_{L+1}$ and use it as a new input $\{\boldsymbol{X}_2, \boldsymbol{X}_3, ..., \hat{\boldsymbol{X}}_{L+1}\}$. This step will be repeated $L$ times to obtain $\{\hat{\boldsymbol{X}}_{L+1}, \hat{\boldsymbol{X}}_{L+2}, ..., \hat{\boldsymbol{X}}_{2L}\}$. We treat this as the prediction and optimize the loss from this quantity. In the following baseline models, we uniformly set $L = 20$.

#### C.4.1 Baseline methods and SINDy-SHRED setting

**ResNet.** We use the residual neural network (ResNet) (He et al., 2016) as a standard baseline. We set the input sequence length to 20, and we predict the next frames autoregressively. For ResNet, the first convolutional layer has 64 channels with kernel size 3, stride 1 and padding 1. Then, we repeat the residual block three times with two convolutional layers. We use ReLU as the activation function. After the residual blocks, the output is generated via a convolutional layer with kernel size 1, stride 1, and padding 0. We set the batch size to 8, and we use AdamW optimizer with learning rate $1e^{-3}$, weight decay $1e^{-2}$ for the training of 500 epochs.

**SimVP.** SimVP (Gao et al., 2022) is the recent state-of-the-art method for video prediction. This method utilizes ConvNormReLU blocks with a spatio-temporal features translator (i.e. CNN). The ConvNormReLU block has two convolutional layers with kernel size 3, stride 1, and padding 1. After 2D batch normalization and ReLU activation, the final forward pass includes a skip connection unit before output. The encoder first performs a 2D convolution with 2D batch normalization and ReLU activation. Then, three ConvNormReLU blocks will complete the input sequence encoding process. The translator in our implementation is a simple CNN which contains two convolutional layers. The decoder has a similar structure to the encoder by reversing its structure. We similarly set the batch size to 8 with AdamW optimizer for 500 epochs.

**ConvLSTM.** Convolutional Long Short-Term Memory (Shi et al., 2015) is a classical baseline for the prediction of video sequence and scientific data (e.g. weather, radar echo, and air quality). The ConvLSTM utilizes features after convolution and performs LSTM modeling on hidden states. The ConvLSTM model has two ConvLSTM cells that have an input 2D convolutional layer with kernel size 3 and padding 1 before the LSTM forward pass. The decoder is a simple 2D convolution with kernel size 1, and zero padding. We similarly set the batch size to 8 with AdamW optimizer for 500 epochs.

**PredRNN.** PredRNN (Wang et al., 2017) is a recent spatiotemporal modeling technique that builds on the idea of ConvLSTM. We follow the same network architecture setting as in ConvLSTM and similarly set the batch size to 8 with AdamW optimizer for 500 epochs.

**SINDy-SHRED.** We select and fix 100 pixels as sensor measurements from the entire $648$ dimensional space. We remove non-informative sensors, defined as remaining constant through the entire

video. We set the lag to 60. For the setting of network architecture in SINDy-SHRED, we follow the same settings as in the prior experiments but with latent dimension of 1. The timestep of forward integration is set to $dt = \frac{1}{300}$ corresponding to frame rate of the video at 30 FPS. We set the batch size at $8$ and the learning rate to $5e^{-4}$. The thresholding procedure is executed every 300 epochs with thresholds ranging from $(0.4, 4.0)$. We include 3 stacked GRU layers, and a two-layer ReLU decoder with 16 and 64 neurons. We use dropout to avoid overfitting with a dropout rate of 0.1. SINDy-SHRED discovers two candidate models.

## D Experiment on the 2D Kolmogorov flow

The 2D Kolmogorov flow data is a chaotic turbulent flow generated from the pseudospectral Kolmogorov flow solver (Canuto et al., 2007). The solver numerically solves the divergence-free Navier-Stokes equation:

$$\begin{cases} \nabla \cdot \boldsymbol{u} = 0 \\ \partial_t \boldsymbol{u} + \boldsymbol{e} \nabla \boldsymbol{u} = -\nabla \boldsymbol{p} + \mathrm{v} \Delta \boldsymbol{u} + f \end{cases}, \tag{54}$$

where $\boldsymbol{u}$ stands for the velocity field, $\boldsymbol{p}$ stands for the pressure, and $f$ describes an external forcing term. Setting the Reynolds number to 30, the spatial field has resolution $80 \times 80$. We simulate the system forward for 180 seconds with $6,000$ available frames. We standardize the data within the range of $(0, 1)$ and randomly fix 10 sensors from the 6,400 available spatial locations (0.16%). The lag parameter is set to 360.

For the setting of SINDy-SHRED, we slightly change the neural network setting because the output domain is 2D. Therefore, after the GRU unit, we use two shallow decoders to predict the output of the 2D field. The two decoders are two-layer ReLU networks with 350 and 400 neurons. We set the latent dimension to 3. The time step for forward integration is set to $dt = 0.003$ which corresponds to the FPS during data generation. We set the batch size to 256 and the learning rate to $5e^{-4}$ using the Lion optimizer (Chen et al., 2024). The thresholding procedure is executed every 100 epoch with the total number of training epochs as 200. The thresholds range from $(0.4, 4)$.

As a chaotic system, the latent space of the Kolmogorov flow is much more complex than all the prior examples we considered. Thus, we further apply seasonal-trend decomposition from the original latent space. We define the representation of the latent hidden state space after decomposition as $(z_1, z_2, z_3, z_4, z_5, z_6)$, where $(z_{2i}, z_{2i+1})$ is the seasonal trend pair of the original latent space.

$$\begin{cases} \dot{z}_1 &= -0.007 z_3 + 0.009 z_5, \\ \dot{z}_2 &= -0.207 z_4, \\ \dot{z}_3 &= -0.011 z_1 - 0.008 z_5, \\ \dot{z}_4 &= 0.103 z_2, \\ \dot{z}_5 &= -0.012 z_1 + 0.006 z_3. \\ \dot{z}_6 &= 0.151 z_1 z_2. \end{cases} \tag{55}$$

In Eqn. 55, we find that $z_1, z_3, z_5$ are essentially a linear system. $z_2, z_4, z_6$ capture higher-order effects that are difficult to model without signal separation. We generate the trajectory from the initial condition at time point 0 and perform forward integration in Fig. 12. As we increase the Reynolds number, the discovery fails to produce robust predictions.

This representation also demonstrates nice predictions for future frames. In Fig. 13, the future prediction has an averaged MSE error of $0.035$ for all available data samples. The sensor-level prediction in Fig. 23 further demonstrates the details of the reconstruction.

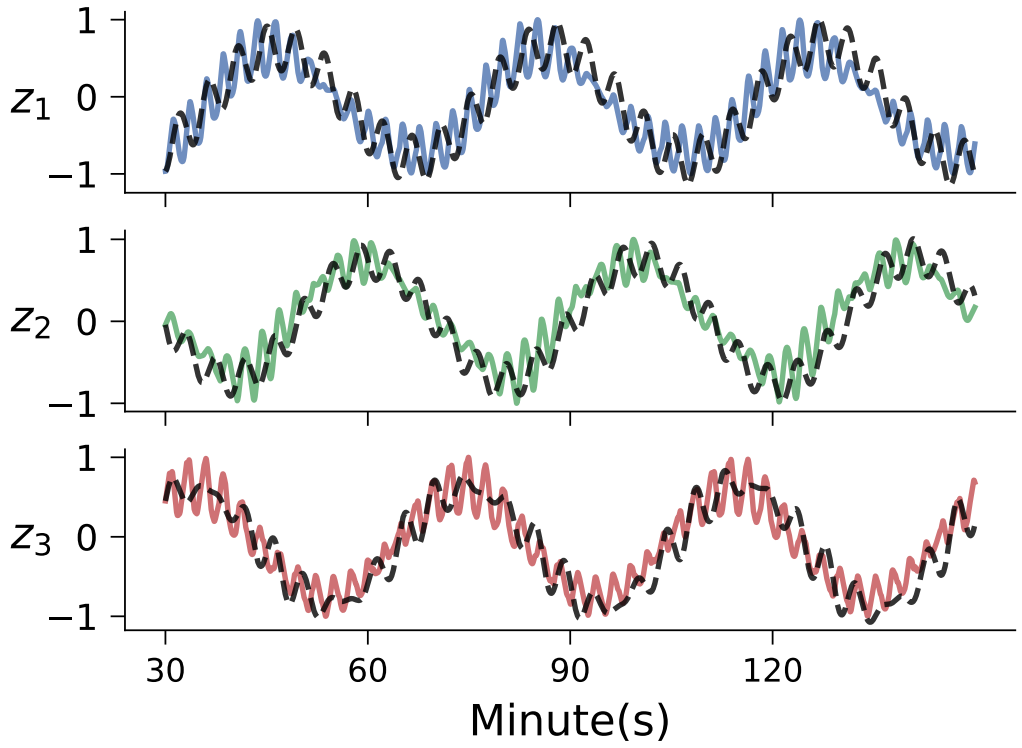
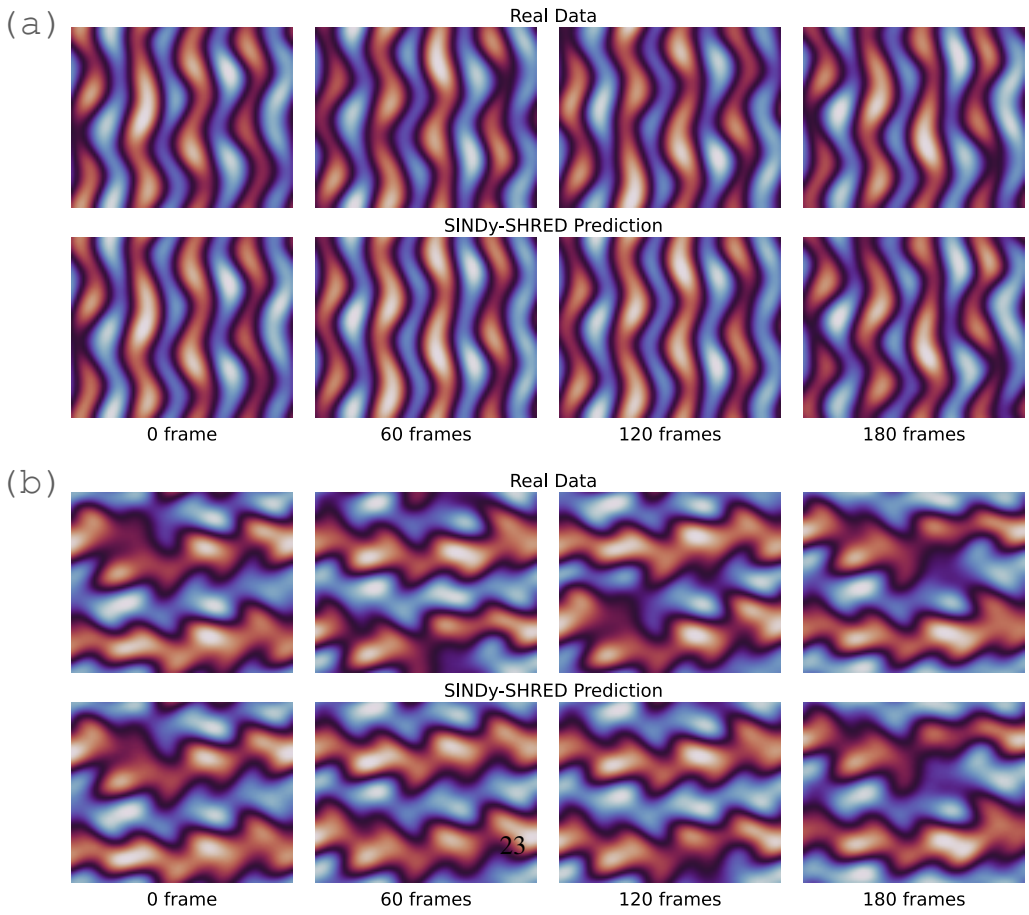



Figure 12: Extrapolation of latent representation in SINDy-SHRED from the discovered dynamical system for the 2D Kolmogorov flow data. Colored: true latent representation. Black: SINDy extrapolation.



# E SENSOR LEVEL PLOTS OF EXPERIMENTS

## E.1 SEA SURFACE TEMPERATURE

**3D visualization of SINDy-SHRED**

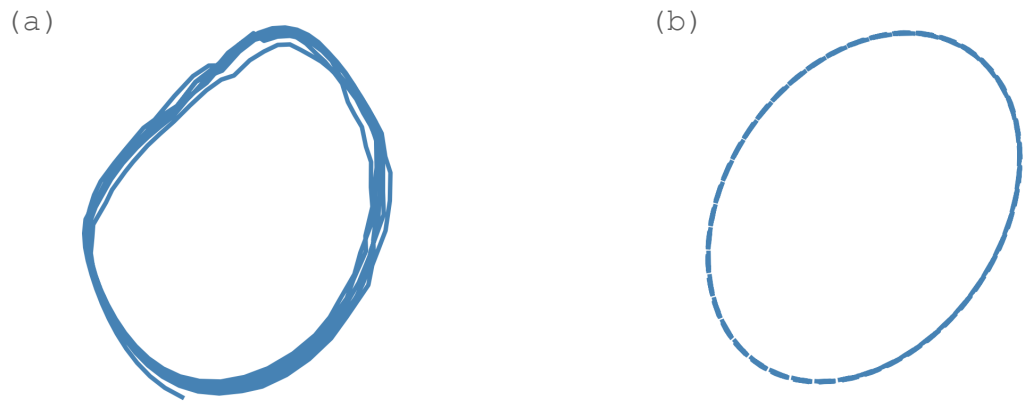



Figure 14: 3D reconstruction of the original latent space and SINDy simulated latent space.

**Long-term extrapolation of SINDy-SHRED.**

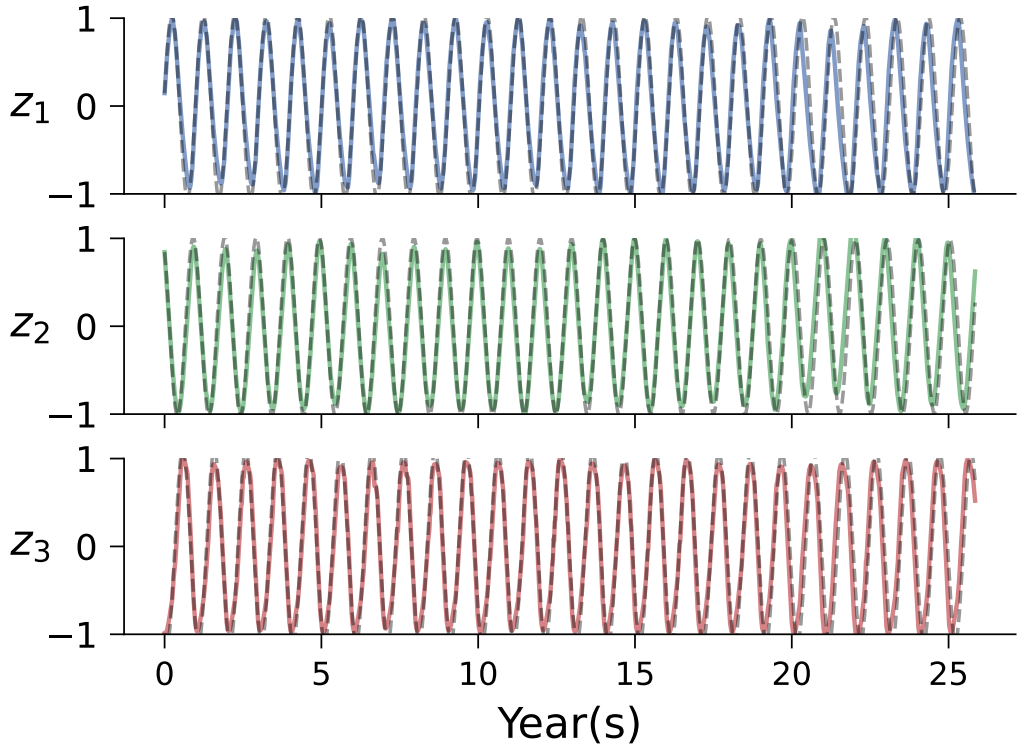



Figure 15: Extrapolation of latent representation in SINDy-SHRED from the discovered dynamical system for SST over the entire 27 years. Colored: true latent representation. Grey: SINDy extrapolation.

**Sensor-level prediction on the SST dataset.**

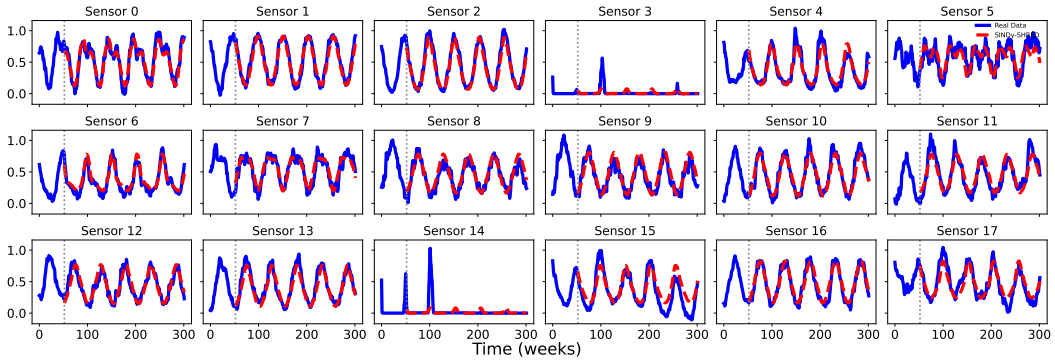



Figure 16: Extrapolation of SINDy-SHRED for sensor-level predictions on the SST data. We randomly picked 18 sensors from spatial locations that are not in the sparse sensor training. The extrapolation shows the SINDy-SHRED prediction for the following 300 weeks.

### E.2 OZONE DATA

**Convergence behavior of SINDy-SHRED on the Ozone dataset.**

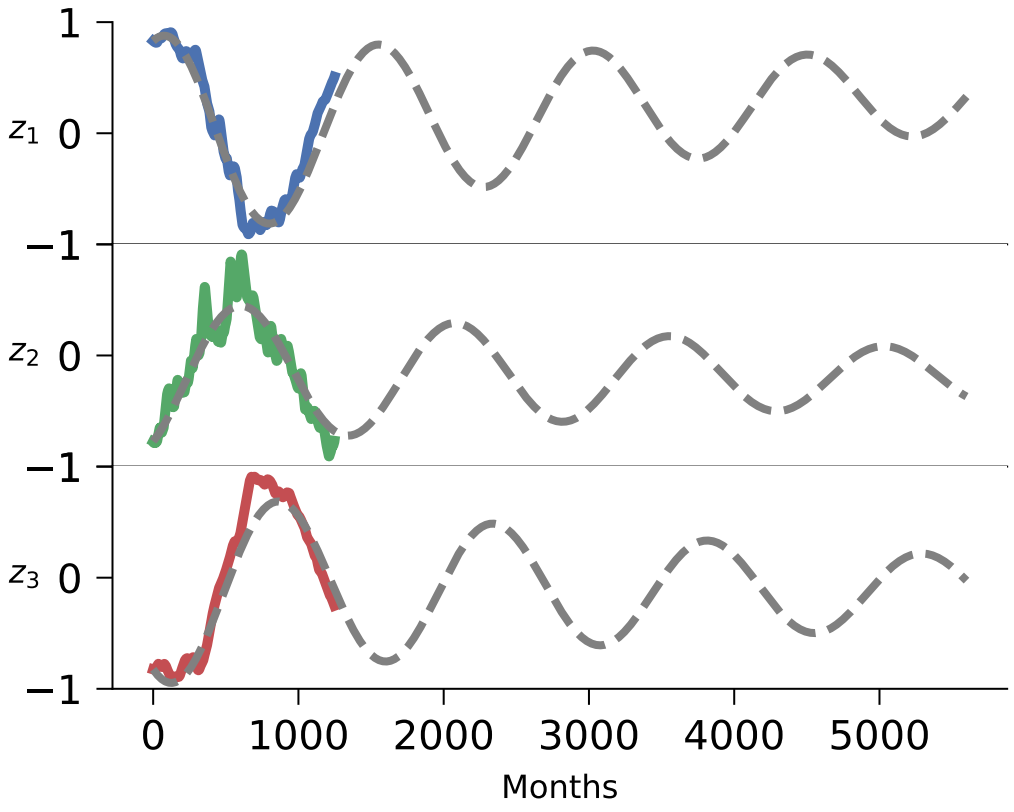



Figure 17: Long term extrapolation of Ozone data. The latent SINDy model presents a convergence behavior towards the mean-field solution.

**Sensor-level prediction on the Ozone dataset.**

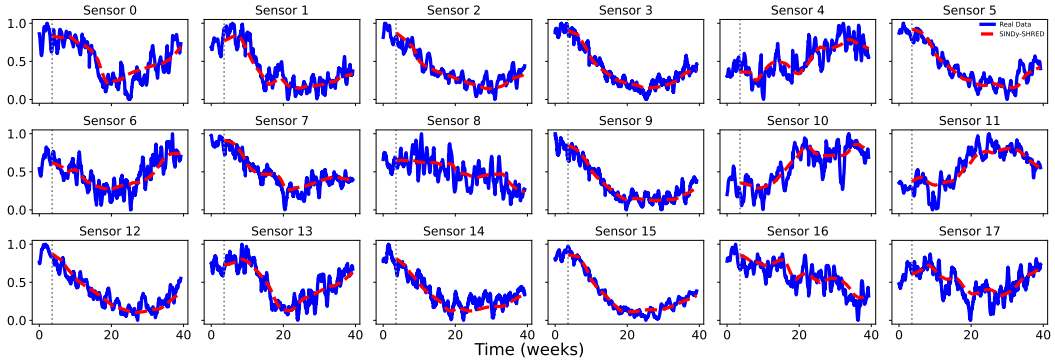



Figure 18: Extrapolation of SINDy-SHRED for sensor-level predictions on the Ozone data. We randomly picked 18 sensors from spatial locations that are not in the sparse sensor training. The extrapolation shows the SINDy-SHRED prediction for the following 40 weeks.

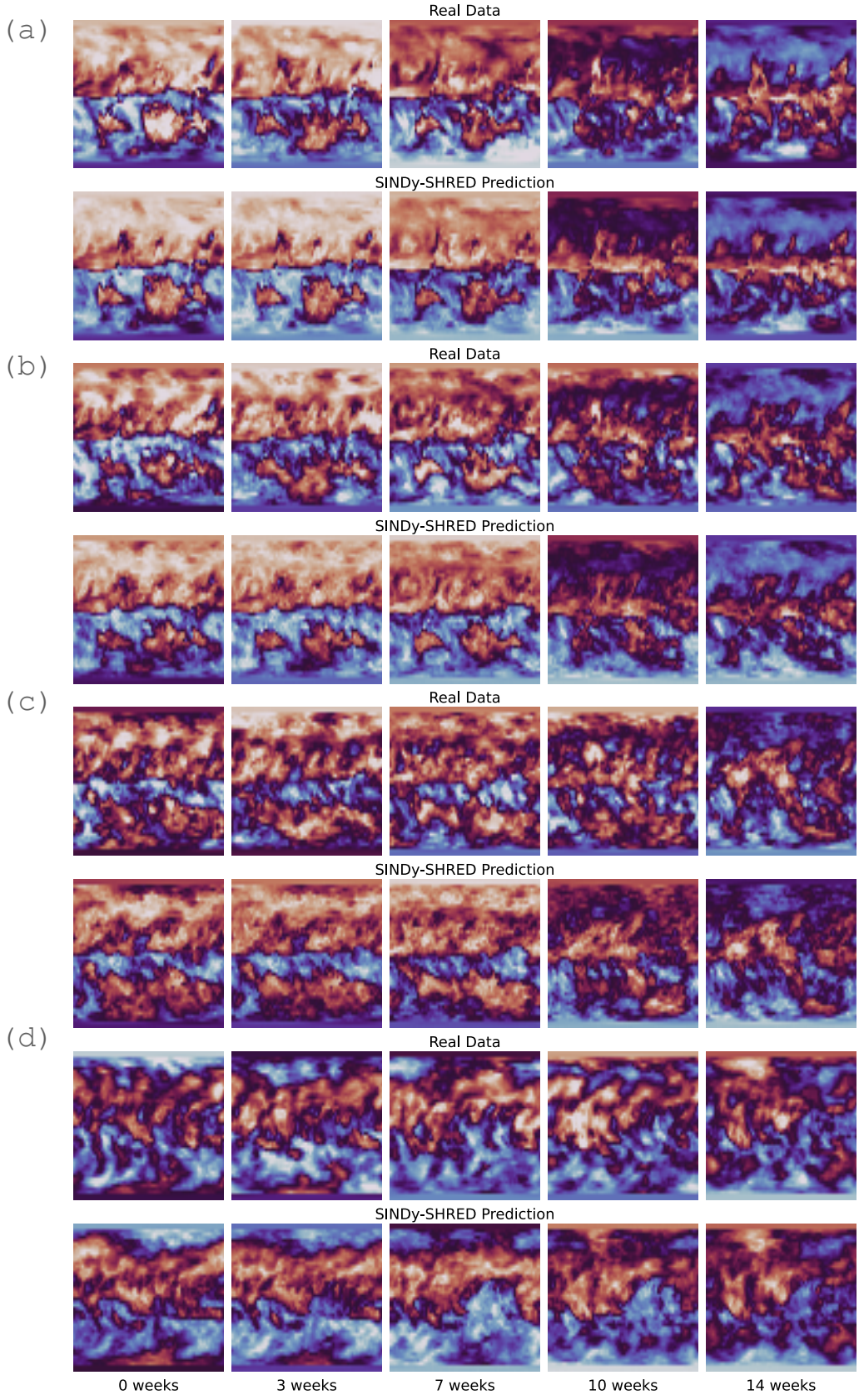



Figure 19: Reconstruction of atmospheric ozone concentration data for different elevation (a) 0 km (b) 4 km (c) 8 km (d) 12 km.

### E.3 FLOW OVER A CYLINDER

**Sensor-level prediction on the flow over a cylinder dataset.**

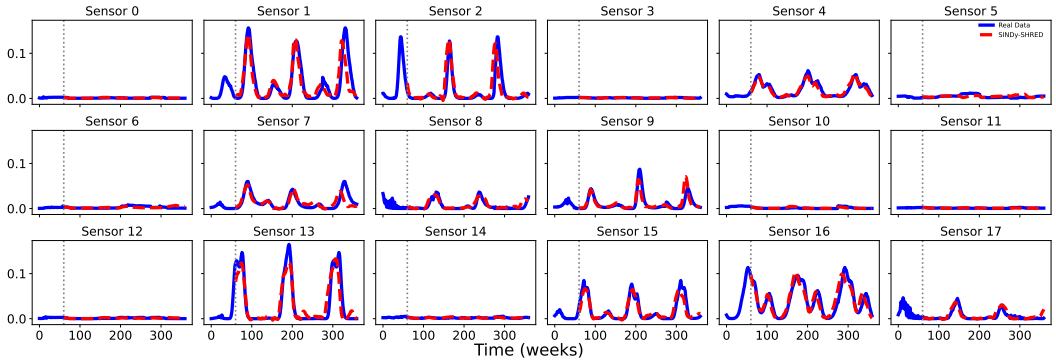



Figure 20: Extrapolation of SINDy-SHRED for sensor-level predictions on the flow over a cylinder data. We randomly picked 18 sensors from spatial locations that are not in the sparse sensor training. The extrapolation shows the SINDy-SHRED prediction for the following 400 frames.

**Long-term extrapolation on the flow over a cylinder dataset.**

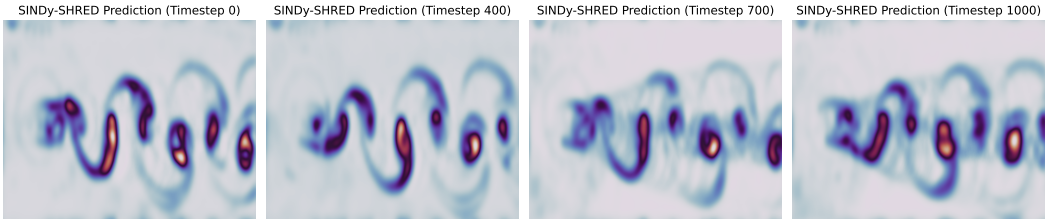



Figure 21: Prediction of the flow over a cylinder data from time step 0 (reconstruction) to 1000 frames. We note this extrapolation is completely out of the dataset. The real data for testing is only available up to 500 frames.

### E.4 PENDULUM

**Sensor-level prediction on the moving pendulum dataset.**

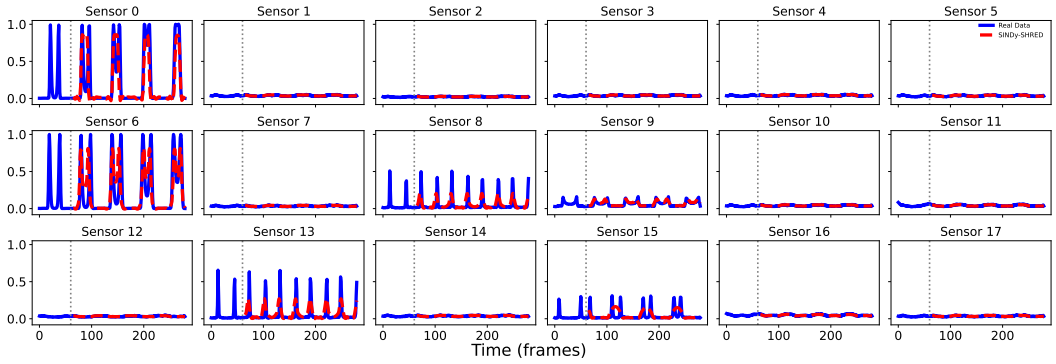



Figure 22: Extrapolation of SINDy-SHRED for sensor-level predictions on the moving pendulum data. We randomly picked 18 sensors from spatial locations that are not in the sparse sensor training. The extrapolation shows the SINDy-SHRED prediction for the following 382 frames.

### E.5 KOLMOGOROV FLOW

**Sensor-level prediction on the chaotic 2D Kolmogorov flow dataset.**

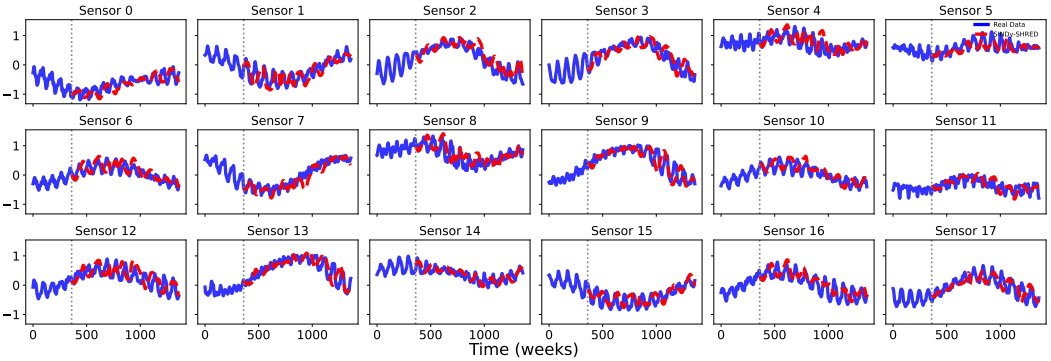



Figure 23: Extrapolation of SINDy-SHRED for sensor-level predictions on the 2D Kolmogorov flow data. We randomly picked 18 sensors from spatial locations that are not in the sparse sensor training. The extrapolation shows the SINDy-SHRED prediction for the following 1500 frames.