

Interpretable machine learning and governing law discovery

Mars Liyao Gao

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington
2026

Reading Committee:
J. Nathan Kutz, Chair
Matthew Golub
Sewoong Oh

Program Authorized to Offer Degree:
Computer Science & Engineering

© Copyright 2026

Mars Liyao Gao

University of Washington

Abstract

Interpretable Machine Learning and Governing Law Discovery

Mars Liyao Gao

Chair of the Supervisory Committee:

Professor J. Nathan Kutz

Departments of Applied Mathematics and Electrical and Computer Engineering

Modern machine learning has greatly expanded our ability to model complex scientific data, but prediction alone is not sufficient for scientific discovery. In many real-world systems, the goal is not only to approximate observations, but to recover interpretable coordinates, identify governing equations, quantify uncertainty, and make reliable predictions from sparse, noisy, and partially observed measurements. In this thesis, I present my work toward developing machine learning methods for interpretable physical law discovery in realistic data regimes. In the first work, I developed Bayesian SINDy autoencoders for jointly discovering latent coordinates, governing equations, and physical constants from high-dimensional observations. By combining representation learning, sparse regression, and Bayesian uncertainty quantification, this method enables more reliable discovery when data are limited or noisy, including from real video data where the intrinsic physical variables are not directly observed. Next, I introduced SINDy-SHRED and Koopman-SHRED, which extend governing law discovery to high-dimensional spatiotemporal systems measured only through sparse sensors. These models use sensor histories to learn informative latent states, reconstruct full fields, and impose parsimonious nonlinear or linear dynamical structure on the latent space, enabling interpretable long-term prediction for systems such as fluids, sea-surface temperature, and video dynamics. I then studied the reliability of sparse model discovery

through ensemble and Bayesian uncertainty estimation, showing how uncertainty estimates can support stable variable selection and distinguish reliable terms from artifacts of noise or limited data. Moving beyond structured-grid assumptions, I proposed mesh-free sparse identification of nonlinear dynamics, using neural networks and automatic differentiation to discover PDEs from scarce, noisy, and irregularly sampled measurements. Finally, I developed UQ-SHRED, a distributional sparse-sensing framework that learns conditional uncertainty over full spatiotemporal fields from limited sensor histories. Collectively, this thesis advances a unified view of scientific machine learning in which models do not merely fit data, but discover interpretable structure, quantify uncertainty, and adapt physical law discovery to the imperfect data conditions encountered in real scientific systems.

Acknowledgement

I have always been told that the acknowledgments are the best part of a thesis. After years of research, writing, uncertainty, revision, and growth, I now understand why. A dissertation is often written in one person's name, but it is never the work of one person alone. It is shaped by family, teachers, mentors, collaborators, friends, and all the people who offered support, patience, perspective, and kindness along the way.

First and foremost, I would like to thank my parents and family. I am deeply grateful to S.W. Gao, C. Cui, K.L. Gao, and X.Y. Gao for their unconditional love, support, and belief in me. I also thank Xuelan Fan, Jing Cui, my grandfather, and my grandparents, whose care and strength have shaped who I am. I am grateful to my aunt Gao, my brothers Tianliang Gao, Tianming Gao, Min Hu, and Dayong Yang, and my nephew Boran Yang, Qiuxue Yang. My family has been the foundation beneath everything I have done. Your support gave me the courage to leave home, pursue difficult questions, and continue forward even when the path was uncertain.

I am grateful to my friends, who made this journey much more human and much more joyful. I thank Xiaomei Mo, Wok Lee, and many others who shared conversations, meals, advice, encouragement, and companionship with me during different stages of my Ph.D. These friendships reminded me that life is larger than research, and that the difficult parts of graduate school become easier when there are people who understand, listen, and stay.

I am also deeply grateful to my colleagues and friends: Zhihan Zhang, Yikun Zhang, Rulin Shao, Yuxuan Bao, Xingyue Zhang, Jan P. Williams, Alexey Yermakov, Amy S. Rude, Pavel Komarov, Carrie Yuan, Yiyu Ni, Motoya Ohnishi, Rohan Baijal, Harine Ravichandiran, Tyler Han, Lancelot Wathieu, Chandler Petersen, Jason Hoffman, Gian Marco Visani, Jonathan

Hayase, Ananya Harsh Jha, Ruqi Shen, Weihang Xu, Yiping Wang, Mengyi Shan, Ziyun Chen, Helen He, Kai Hidajat, Matthew Shell, Dileep Avula, Wenhao Chai, Yingqing He, Peter Bolun Liu, Houjie Wang, Zhexiao Lin, Wenlong Ji, David Zoro, David Ye, Yiming Wang, Meiyao Li, Danyang Li, Hongshan Li, Yifan Du, Dongxia Wu, Sam Otto, Anastasia Bizyaeva, Dave Zhiwei Li, Elise DeGoede, and Viola Wang. I have been fortunate to be surrounded by people who are talented, generous, and inspiring. Many of them have influenced my thinking, helped me through research problems, or simply made graduate school a more enjoyable and meaningful experience.

I would also like to thank the mentors, collaborators, and advisors who have shaped my scientific development. I am grateful to Matthew Golub, Brad Lipovsky, Sewoong Oh, Marine Denolle, Banghua Zhu, Wei Zhu, Yi-An Ma, Guang Lin, Rose Yu, Wei Deng, Steven L. Brunton, Lucas Janson, Marco Carone, Alex Luedtke, and Emily B. Fox for their guidance, encouragement, and intellectual influence. Their perspectives helped broaden my view of machine learning, dynamical systems, statistics, and scientific discovery. I have learned from their technical insights, their standards for research, and their examples as scientists.

I am deeply grateful to my research advisor, J. Nathan Kutz. Nathan has been far more than an advisor to me. He gave me the freedom to explore ambitious ideas, the confidence to pursue difficult problems, and the guidance to turn those ideas into research. His vision for scientific machine learning, dynamical systems, and data-driven discovery deeply shaped the direction of this dissertation. He taught me how to think broadly, write clearly, connect mathematics with physical intuition, and look for the essential structure beneath complex phenomena. Just as importantly, he supported me through the uncertain and difficult moments of graduate school with patience, trust, and generosity. Many of the ideas in this thesis grew from conversations with him, and much of my growth as a researcher came from his mentorship. I am deeply grateful for the opportunity to have been his student.

Finally, I thank everyone who, in ways large or small, helped me reach this point. This dissertation is a record not only of research, but also of the people, relationships, and experiences that made the research possible.

Table of Contents

Abstract	1
Preface	3
1 Introduction	1
2 Bayesian autoencoders for equation discovery	6
2.1 Introduction	6
2.2 Background	9
2.2.1 Sparse identification of nonlinear dynamics	9
2.2.2 Bayesian and sparse deep learning	12
2.3 Method: Bayesian SINDy Autoencoders	13
2.3.1 Likelihood setting for Bayesian SINDy Autoencoders	13
2.3.2 Stochastic Gradient Langevin Dynamics	16
2.3.3 Empirical Bayes Variable Selection in SINDy Autoencoder	17
2.3.4 Prediction with Bayesian SINDy Autoencoder	19
2.4 Experiments	20
2.4.1 Learning Physics from Synthetic Video Data	21
2.4.2 Learning dynamical system from GoPro recording on a moving rod	27

2.5	Conclusion and Future works	33
3	SINDy and Koopman Operators with Shallow Recurrent Decoder Networks .	39
3.2	Related works	42
3.2.1	Sparse identification of nonlinear dynamics (SINDy)	44
3.3	Methods	45
3.3.1	Learning representations for physics discovery using SHRED	46
3.3.2	Latent space regularization via SINDy and Koopman	47
3.3.3	Theoretical analysis on dynamical system learning	52
3.4	Computational Experiments	53
3.4.1	GoPro physics video data: flow over a cylinder	54
3.4.2	Prediction and baseline study of single shot pendulum video	57
3.4.3	NOAA sea-surface temperature data	59
3.4.4	Predicting isotropic turbulence flow	61
3.4.5	3D Atmospheric ozone concentration	63
3.4.6	Visualizing the convex loss landscape	65
3.5	Discussion and conclusion	65
3.6	Challenges in rolling out neural networks for fitting a simple sine function	68
3.7	Theoretical justification on the generalization error between SINDy and the neural networks	70
3.7.1	Theoretical setup	70
3.7.2	The error of SINDy-class	72
3.7.3	The error of neural networks	77
3.8	Observation on the convex landscape	83
3.9	Ablation study	85
3.9.1	Effect on the recurrent units	85
3.9.2	A comparison to other baseline methods	87
3.9.3	Quantitative comparison by training SHRED with and without SINDy loss terms	88

3.9.4	Latent space discoveries from different sensor placements are equivalent under linear transformations	92
3.9.5	Hyperparameter robustness	93
3.9.6	Energy spectrum analysis on the isotropic turbulent flow	94
3.10	Experimental details	95
3.10.1	The design of SINDy unit	95
3.10.2	Evaluation metrics	96
3.10.3	Flow over a cylinder	97
3.10.4	Baseline experiment on pendulum	97
3.10.5	Sea-surface temperature data	100
3.10.6	3D atmospheric ozone concentration	100
3.10.7	Isotropic turbulent flow	100
3.11	Experiment on the 2D Kolmogorov flow	101
3.12	Sensor level plots of experiments	104
3.12.1	Sea surface temperature	104
3.12.2	Ozone data	105
3.12.3	Flow over a cylinder	108
3.12.4	Isotropic turbulent flow	109
3.12.5	Pendulum	110
3.12.6	Kolmogorov flow	110
3.13	Analysis of learned ODEs	111
3.13.1	Sea-surface temperature	111
3.13.2	Ozone data	111
3.13.3	Flow over a cylinder	112
3.13.4	Isotropic turbulence flow	113
3.14	Algorithm	113
4	Theoretical guarantees of SINDy and E-SINDy	115
4.1	Introduction	115

4.2	Background	120
4.2.1	Sparse identification of nonlinear dynamics	120
4.2.2	Variable selection, bootstrap resampling, and uncertainty quantification	122
4.3	Sequential thresholding least-square in sparse regression	123
4.3.1	Thresholding least-squares estimate	124
4.4	Bagging sequential thresholding least-square estimator	127
4.4.1	Bagging inclusion probability STLS	128
4.4.2	The analysis on bagging inclusion probability estimation	129
4.5	Uncertainty quantification with bootstrap	131
4.5.1	Consistency of bootstrap inference	132
4.5.2	Connection to Bayesian uncertainty quantification	133
4.6	Experiments	136
4.6.1	Implementation	136
4.6.2	Simulation of linear models: consistent variable selection	137
4.6.3	Lotka–Volterra model discovery via Ensemble SINDy	141
4.7	Conclusion and discussion	144
5	Mesh-Free equation discovery	167
5.1	Related works	169
5.2	Mesh-free sparse identification of nonlinear dynamics	170
5.3	Numerical experiments	173
5.3.1	Burgers’ equation	173
5.3.2	1D heat equation	176
5.3.3	Korteweg–De Vries equation	177
5.3.4	2D advection-diffusion equation	178
5.3.5	Comparison with previous works	180
5.4	Conclusion	181
5.5	Experimental details	183

5.5.1	Error Metrics	183
5.5.2	Burgers' equation	184
5.5.3	Heat equation	186
5.5.4	KdV equation	187
5.5.5	Advection-diffusion equation	189
6	Uncertainty quantification for SHRED via engression	192
6.1	Introduction	192
6.2	Preliminaries	195
6.2.1	The SHRED Architecture	195
6.2.2	Engression and the Energy Score	196
6.2.3	Uncertainty quantification methods in deep learning	197
6.3	Method: UQ-SHRED	197
6.3.1	Architecture	198
6.3.2	Inference	200
6.3.3	Theoretical guarantee	200
6.4	Experiments	205
6.4.1	Evaluation metrics	205
6.4.2	Sea-surface temperature data	206
6.4.3	Isotropic turbulent flow	207
6.4.4	Neural activity data	208
6.4.5	Solar activity data	209
6.4.6	1D RDE transient Stage	210
6.4.7	Ablation study on SST data	213
6.5	Conclusion	216
6.6	Additional figs	218
7	Conclusion	223

List of Figures

2.1	General structure of the Bayesian SINDy autoencoder architecture. (a) An autoencoder architecture is used to discover intrinsic coordinates $\mathbf{z}$ from high-dimensional input data $\mathbf{x}$. The encoder $f(\mathbf{x})$ transforms the high-dimensional input into the a low-dimensional space $\mathbf{z}$, and the decoder $g(\mathbf{z})$ reconstructs $\mathbf{x}$ from the low-dimensional subspace. (b) A Bayesian SINDy model infers the underlying dynamics of the latent dimension with uncertainty estimation. The active terms are identified by Ξ . The color represents the inclusion probability $[0.0, 1.0]$ from grey (non-active) to blue (active). The loss function encourages the network to minimize both the reconstruction error and the SINDy loss in $\mathbf{z}$ and $\mathbf{x}$. The Bayesian prior works as a regularization term on Ξ for sparse inference.	13
2.2	Video prediction with uncertainty estimation from latent dimension in pixel space.	19

2.3	Bayesian estimation and uncertainty quantification visualization of SINDy coefficient for Lorenz system under Laplace prior. The color bar represents the inclusion probability estimate given the coefficient magnitude. The color represents the inclusion probability $[0.0, 1.0]$ from white (non-active) to dark blue (active).	22
2.4	(a) Trajectory prediction for in-distribution data with uncertainty quantification generated by posterior samples from Bayesian inference on SINDy coefficient. (b) Trajectory prediction for out-of-distribution data with uncertainty quantification generated by posterior samples from Bayesian inference on SINDy coefficient. (c.1) Discovered Lorenz systems and (c.2) transformation to the standard space. (c.3) Lorenz system generated from the ground truth.	22
2.5	(a) Bayesian estimation and uncertainty quantification visualization of SINDy coefficient for reaction-diffusion under Spike-and-slab prior. The color represents the inclusion probability $[0.0, 1.0]$ from light blue (non-active) to dark blue (active). (b) Visualization of in-distribution (top) and out-of-distribution (bottom) predicted dynamics. (c) The generated attractor from the mean of Bayesian inference.	25
2.6	(a) Bayesian estimation and uncertainty quantification visualization of SINDy coefficient for pendulum data under Spike-and-slab prior. The color represents the inclusion probability $[0.0, 1.0]$ from light blue (non-active) to blue (active). (b) Visualization of predicted dynamics with uncertainty quantification from 6 different initializations.	27
2.7	Preprocessing pipeline to high-dimensional video data to remove auxiliary information and reduce the size of video frames.	28
2.8	Left: SSGL regularization plot. Right: Model selection via test error.	29

2.9	(a) Bayesian estimation and uncertainty quantification visualization of SINDy coefficient for real moving rod data under Spike-and-slab prior. The color represents the inclusion probability $[0.0, 1.0]$ from light blue (non-active) to dark (active). (b) Visualization of prediction in both latent dynamics and pixel space with uncertainty quantification. (c) Reconstruction of Bayesian SINDy autoencoder for video frame, and temporal derivatives.	30
2.10	Left: by running the experiment 5 times with 2000 training epochs and 1500 refinement epochs, we plot the mean and standard deviation of testing loss under the log scale. Right: by running the experiment once with 7500 training epochs and 1500 refinement epochs, we plot the testing loss under the log scale.	31
2.11	Latent dimension visualization after rescaling (blue curve) versus manually labelled moving rod angle by human (grey curve).	32
2.12	(a) Bayesian estimation and uncertainty quantification visualization of SINDy coefficient for real moving rod data under Laplace prior. The color represents the inclusion probability $[0.0, 1.0]$ from light blue (non-active) to dark blue (active). (b) Latent dimension visualization after rescaling (blue curve) versus manually labelled moving rod angle by human (grey curve). (c) Visualization of predicted dynamics in latent space with uncertainty quantification.	35
2.13	Left: Laplace prior (LASSO) regularization plot. Right: Model selection via test error.	36

-
- 3.1 (a) Illustration of the SINDy-SHRED and Koopman-SHRED architecture. SINDy-SHRED transfers the original sparse sensor signal (red) to an interpretable latent representation (purple) that falls into the SINDy-class functional. This framework can be adapted into Koopman-SHRED by restricting the library $\Theta(\cdot)$ to be linear. The shallow decoder performs a reconstruction in the state space. We obtain an interpretable linear model for the sea-surface temperature data considered (details in Sec. 3.4.3). (b) The loss function consists of three parts: (i) the SHRED loss controls the reconstruction accuracy of the state space, (ii) the Ensemble SINDy loss helps to model the parsimonious dynamics of the latent space, and (iii) the sparsity constraint identifies the governing equation within this optimization framework. (c) We visualize the globally convex loss landscape of SINDy-SHRED as in [168]. 41
- 3.2 (a) Extrapolation of latent representation in SINDy-SHRED from the discovered dynamical system for flow over a cylinder data. Colored: **latent representation**. Grey: SINDy extrapolation from derived latent dynamical system. (b) Evolution of dynamical system (3.5). The left two panels are the phase plane z_1 vs z_2 and z_3 vs z_4 respectively. The magenta plus symbols are the fixed points. The nonlinear limit cycle behavior is shown in the right panel for $z_j(t)$. (c) Long-term pixel space video prediction via SINDy-SHRED. We demonstrate the forward prediction outcome up to 180 frames. 55
- 3.3 (a) Extrapolation of latent representation in SINDy-SHRED from the discovered dynamical system for the pendulum moving data. Blue: true latent representation. Grey: SINDy extrapolation. (b) The pendulum video generation outcome from ResNet, SimVP, ConvLSTM, PredRNN, SINDy, SHRED, SINDy-autoencoder, and SINDy-SHRED from frame 20 to frame 245. 58

3.4	(a) Extrapolation of latent representation in SINDy-SHRED from the discovered dynamical system for SST. Colored: true latent representation. Grey: SINDy extrapolation. (b) Decoder reconstruction of three independent directions z_1, z_2, z_3 in the latent space. (c) Long-term global sea-surface temperature prediction via SINDy-SHRED from week 0 to week 100. We crop the global map for better visualization.	61
3.5	Visualizations of latent spaces and predictions for SINDy-SHRED applied to PDE simulations of isotropic turbulence, atmospheric chemistry, and 2D Kolmogorov flow datasets. Left: Extrapolation of latent representations using SINDy-SHRED from the discovered dynamical system for the ozone data, showing the true latent representation (colored lines) and SINDy extrapolation (grey lines). Right: Long-term spatiotemporal predictions from SINDy-SHRED compared to ground truth. The details of the 2D Kolmogorov experiment can be found in the SI Appendix.	62
3.6	Loss landscape plot of SHRED (top row) and SINDy-SHRED (bottom row) for all datasets presented: sea-surface temperature, Ozone, flow over a cylinder, isotropic flow, and pendulum.	65
S1	Fitting a simple sine function using SINDy and the GRU network.	69
S2	(a) Visualization of loss landscape from parameters ranging from $[-5, 5]$. (b) Visualization of loss landscape from parameters ranging from $[-10, 10]$. (c) Visualization of loss landscape from parameters ranging from $[-100, 100]$. (d) Visualization of loss landscape from parameters ranging from $[-1000, 1000]$	84
S3	Visualization of loss landscapes for SHRED parameters along random directions, showing how loss values change when moving from the origin toward randomly generated points in parameter space. The figure provides a detailed view of loss behavior at different initial points. (a) shows the parameter range $[-100, 100]$ and (b) shows the parameter range $[-1, 1]$	84

S4	Comparison of latent spaces learned by SHRED (no SINDy regularization) with different recurrent units on the test set. The x-axis represents the number of years in the sea-surface temperature dataset.	87
S5	Comparison of latent spaces learned by SHRED (top row) and SINDy-SHRED (top row). The color represents the induced latent space evolution from early time (blue) to the end of training (green).	88
S6	Top row: the latent space evolution with increasingly stronger SINDy regularization parameter, ranging from 0.1 to 10.	89
S7	Comparison of the latent space discovery from SHRED and SINDy-SHRED. Without SINDy regularization, SHRED has no guarantee to compress the raw observation into a physically interpretable latent space. SINDy-SHRED finds an accurate ODE model that can be used to achieve long term forecasts. . . .	90
S8	Comparison of long-term prediction between SHRED and SINDy-SHRED. The frame error is evaluated by each frame (left) and the cumulative error is calculated by summing up all previous frame errors (right).	91
S9	Comparison of long-term prediction between SHRED and SINDy-SHRED. The frame error is evaluated by each frame (left) and the cumulative error is calculated by summing up all previous frame errors (right).	92
S10	Latent space discovery from the sea-surface temperature data under two different sensor placements. The raw discoveries appear to be different but the two discoveries are close under a linear transformation.	93
S11	Error decay with training epochs using different hyperparameter settings: learning rates (left) batch sizes (right). All configurations converge to a similar level of training error.	94
S12	SINDy-SHRED prediction captures important statistical properties for long-term predictions.	95
S13	Diagram of the RNN form of SINDy.	96

S14	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.	103
S15	Long-term prediction via SINDy-SHRED for 2D Kolmogorov flow data. . . .	104
S16	3D reconstruction of the original latent space and SINDy simulated latent space.	104
S17	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.	105
S18	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.	105
S19	Long term extrapolation of Ozone data. The latent SINDy model presents a convergence behavior towards the mean-field solution.	106
S20	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.	106
S21	Reconstruction of atmospheric ozone concentration data for different elevation (a) 0 km (b) 4 km (c) 8 km (d) 12 km.	107
S22	Extrapolation of latent representation in Koopman-SHRED from the discovered dynamical system for flow over a cylinder data. Colored: true latent representation. Grey: SINDy extrapolation.	108
S23	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.	108

S24	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.	108
S25	Extrapolation of all 16 latent representation in SINDy-SHRED from the discovered dynamical system for isotropic turbulent flow data for 250 frames. Colored: true latent representation. Grey: SINDy extrapolation.	109
S26	Extrapolation of SINDy-SHRED for sensor-level predictions on the isotropic turbulent 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 250 frames.	109
S27	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.	110
S28	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.	110
4.1	(a) Top: convergence of distributional estimation of the Bayesian SINDy (green) and E-SINDy algorithm (orange) over different sample sizes from the Lotka-Volterra equation coefficient α (as defined in Eqn. 4.26). Bottom: Wasserstein-2 distance between E-SINDy and Bayesian SINDy with Spike-and-slab prior. The shaded area indicates the 95% confidence band. (b) Exponential convergence for E-SINDy in terms of False Discovery Probability and True Discovery Probability for the synthetic Lotka-Volterra dataset. We numerically approximate the constant as our bounds only captures the rate of convergence.	117

4.2	Relative frequency of three synthetic data generation models. Subfigures in the first line show the simulation outcome of Model 1; subfigures in the second line show the simulation outcome of Model 2; subfigures in the third line show the simulation outcome of Model 3. The BLASSO and BSTLS stand for bagging LASSO and bagging STLS.	139
4.3	Empirical false discovery probability and success true discovery probability with increasing number of samples under different sparsity levels . For better comparison, the hyperparameters of all methods are fine tuned for STDP that all non-zero coefficients are identified. We compare each method's performance via false discovery control. The BLASSO and BSTLS stand for bagging LASSO and bagging STLS. On the y-axis, EFDP represents the empirical FDP (Eqn. (4.11)) and ETDP represents the empirical TDP (Eqn. (4.12)).	142
4.4	Empirical false discovery probability and success true discovery probability with increasing number of samples under various levels of noises . For better comparison, the hyperparameters of all methods are tuned to control the worse case of ETDP is around 50%. We compare each method's performance via false discovery control. The BLASSO and BSTLS stand for bagging LASSO and bagging STLS. On the y-axis, EFDP represents the empirical FDP (Eqn. 4.11) and ETDP represents the empirical TDP (Eqn. 4.12).	142
4.5	Distributional approximation with uncertainty quantification of Bayesian SINDy (green-colored) and E-SINDy (orange-colored) to the ground-truth value (dotted line) for all active terms.	143
4.6	Convergence in Wasserstein-2 distance between E-SINDy and the Bayesian SINDy estimates for all four active indices. Shaded regions denote 95% confidence intervals.	144
4.7	Extended plottings of Lotka–Volterra experiment with E-SINDy and Bayesian SINDy.	166

5.1	Illustration of the architecture of mesh-free SINDy. From N sensor measurements at random space-time points x_i^n , the neural network approximates the system state $u_j^n(x_i^n)$. As shown in the 1-D example, the auto-differentiation framework produces $\partial_t u$ (also noted u_t), and a library of K candidate terms composed of spatial or mixed derivatives which will be used in the sparse regression.	168
5.2	Uncertainty quantification and epoch-wise evaluation of the discovered SINDy model for the Burgers' equation.	175
5.3	Uncertainty quantification and epoch-wise evaluation of the discovered SINDy model for the heat equation.	177
5.4	Uncertainty quantification and epoch-wise evaluation of the discovered SINDy model for the KdV equation.	179
5.5	Uncertainty quantification and epoch-wise evaluation of the discovered SINDy model for the 2D advection-difusion equation.	180
5.6	Epoch-wise training and validation loss convergence with different noise levels for the Burgers' equation.	185
5.7	Real (left) vs. predicted (right) solution of the Burgers' equation.	185
5.8	Epoch-wise training and validation loss convergence with different noise levels for the heat equation.	186
5.9	Real (left) vs. predicted (right) solution of the heat equation.	187
5.10	Epoch-wise training and validation loss convergence with different noise levels for the KdV equation.	188
5.11	Real (left) vs. predicted (right) solution of the KdV equation.	188
5.12	Epoch-wise training and validation loss convergence with different noise levels for the advection-diffusion equation.	190
5.13	Real (left) vs. predicted (right) solution of the advection-diffusion equation. .	190

6.1	(a) UQ-SHRED architecture. The noise vector ε is concatenated to the sensor input at each lag step and kept constant across the temporal window. The random noises are concatenated to the sensor input to the GRU, as stated in Eqn. 6.4. (b) The function $G(\cdot)$ represents the SHRED architecture with temporal units and shallow decoder networks. Two independent forward passes with different noise draws produce $G_{\theta}(\mathbf{x}, \varepsilon_1)$ and $G_{\theta}(\mathbf{x}, \varepsilon_2)$, which define the energy score loss (6.7). (c) At inference, K forward passes with independently resampled noise yield an empirical predictive distribution from which quantile-based confidence intervals are extracted. (d) Uncertainty estimation from the Monte Carlo samples for test sensor locations. Black: ground truth, blue: 95% uncertainty bounds.	198
6.2	UQ-SHRED reconstruction of the SST dataset. Shaded regions: 50%, 70%, and 95% confidence intervals; blue dashed: median; black solid: ground truth. . .	206
6.3	UQ-SHRED reconstruction of the isotropic turbulent flow. Shaded regions: 50%, 70%, and 95% confidence intervals; blue dashed: median; black solid: ground truth.	207
6.4	UQ-SHRED reconstruction of the neural activity dataset. Shaded regions: 50%, 70%, and 95% confidence intervals; blue dashed: median; black solid: ground truth.	208
6.5	UQ-SHRED reconstruction of the solar activity dataset. Shaded regions: 50%, 70%, and 95% confidence intervals; blue dashed: median; black solid: ground truth.	209
6.6	Calibration diagrams for UQ-SHRED across all four datasets. Dashed line: perfect calibration. Blue: UQ-SHRED empirical coverage.	209
6.7	UQ-SHRED reconstruction of the 1D RDE transient stage at selected time instances. Top rows: pressure; bottom rows: temperature. Shaded regions: 50%, 70%, and 95% confidence intervals; red dashed: median; black solid: ground truth; red dots: sensor locations.	211

6.8	Calibration diagrams for UQ-SHRED on the 1D RDE transient stage. Dashed line: perfect calibration. Blue: UQ-SHRED empirical coverage.	212
6.9	Stability comparison between UQ-SHRED and deterministic SHRED under temporal extrapolation on the 1D RDE data. Visualizations are presented for temporal traces at a selected spatial location $x = 14.50$ on the temperature field.	213
6.11	Spatial reconstruction of UQ-SHRED on SST data. Columns: ground truth, median reconstruction, absolute error, and 95% CI width.	218
6.12	Spatial reconstruction of UQ-SHRED on isotropic turbulent flow data. Columns: ground truth, median reconstruction, absolute error, and 95% CI width. . . .	219
6.13	Spatial reconstruction of UQ-SHRED on neuro activity data. Columns: ground truth, median reconstruction, absolute error, and 95% CI width.	219
6.14	Spatial reconstruction of UQ-SHRED on solar activity data. Columns: ground truth, median reconstruction, absolute error, and 95% CI width.	220
6.15	Spacetime overview for UQ-SHRED on the 1D RDE transient ignition stage. Columns: ground truth, median reconstruction, absolute error, and 95% CI width. Top rows: pressure; bottom rows: temperature. The 95% CI width concentrates along the detonation fronts, matching the spatial distribution of reconstruction error.	221
6.16	Temporal traces at selected spatial locations for UQ-SHRED on the 1D RDE transient ignition stage. Confidence bands widen during detonation-wave passage and contract during quiescent intervals.	222

List of Tables

3.1	Comparison table of SINDy-SHRED to baseline methods for parameter size, training time, and error over different time horizons.	59
3.2	Comparison of different SINDy-based and SHRED-based methods in model discovery, spatiotempora learning, and reproducibility.	87
4.1	The computational difference in CPU time (seconds) for Bayesian UQ-SINDy [126] and Ensemble SINDy [90] in a Lotka–Volterra system.	135
5.1	Performance of Mesh-free SINDy on tested PDEs	173
5.2	PDE discovery success rate (%) under different sample sizes and noise levels for the Burgers’ equation.	175
5.3	PDE discovery success rate (%) under different sample sizes and noise levels for the heat equation.	177
5.4	PDE discovery success rate (%) under different sample sizes and noise levels for the KdV equation.	178
5.5	PDE discovery success rate (%) under different sample sizes and noise levels for the 2D advection-difussion equation.	179

5.6	Comparison of mesh-free SINDy with previous methods for the Burgers' equation experiment.	181
5.7	Loss of mesh-free SINDy under different sample sizes and noise levels for the Burgers' equation.	185
5.8	Error metrics of SINDy under different noise levels with 5000 samples for the Burgers' equation.	186
5.9	Loss of mesh-free SINDy under different sample sizes and noise levels for the heat equation.	187
5.10	Error metrics of SINDy for under different noise levels with 5000 samples for the heat equation.	187
5.11	Loss of mesh-free SINDy under different sample sizes and noise levels for the KdV equation.	189
5.12	Error metrics of SINDy under different noise levels with 1000 samples for the KdV equation.	189
5.13	Loss of mesh-free SINDy under different sample sizes and noise levels for the advection-diffusion equation.	190
5.14	Error metrics of SINDy under different noise levels with 6000 samples for the advection-diffusion equation.	191
6.1	UQ-SHRED metrics across the four experimental datasets. Calibration reports observed coverage (%) at each nominal confidence level. CRPS represents the continuous ranked probability score. Sharpness is the average CI width. . . .	210
6.2	UQ-SHRED metrics on the 1D RDE transient stage for two test trajectories with different initial-condition perturbation magnitudes. Calibration reports observed coverage (%) at each nominal confidence level. CRPS represents the continuous ranked probability score. Sharpness is the average CI width. . . .	212

1

Introduction

Physical laws are the foundational principles that make the natural world predictable, explainable, and ultimately controllable. It provides a strong basis for all modern science and technology which has shaped modern civilization. For example, mechanics enabled transportation, electromagnetism enabled communication, thermodynamics enabled energy systems, and quantum mechanics enabled modern materials. Unlike purely empirical predictors, physical laws expose the structure of nature itself including invariants, constraints, symmetries, and mechanisms, and therefore support reasoning far beyond the specific measurements from which they were inferred. Discovering such laws remains one of the central goals of science.

At the same time, modern machine learning has demonstrated an extraordinary power of learning via universal approximation. Neural networks and other flexible function approximators have achieved remarkable success across vision, language, robotics, and many other domains. This approximation-first paradigm is powerful because it avoids the need to manually specify the governing structure of a system: given sufficient data, model capacity, and computation, complex input-output relationships can often be learned directly from observations. In this sense, machine learning has expanded the reach of data-driven modeling and has provided a new computational language for representing complex phenomena.

However, approximation alone is not enough for scientific discovery. A model that interpolates accurately within a training distribution may fail when asked to extrapolate to new initial conditions, parameter regimes, forcing functions, geometries, or boundary conditions. Universal approximation theorems guarantee that sufficiently expressive models

can approximate broad classes of functions on compact domains, but they do not by themselves provide reliable generalization. Physical models, which often derived from first principles, nevertheless, impose laws and constraints that can remain valid far beyond the observed data. This distinction is essential: scientific models must not only fit what has been measured, but also reveal what can happen under conditions that have not yet been observed.

In other words, to model the real physical world, data are never truly enough. In many physical, biological, and engineering systems, measurements are sparse, noisy, expensive, irregularly sampled, or only partially observed. Full-state observations are often unavailable, experiments may be difficult to repeat, and high-fidelity simulations may be computationally prohibitive. We can see examples from climate science, where observations are sparse and incomplete because many regions of the Earth system are difficult to measure continuously. We can also see this in healthcare, where data are systematically biased toward routine and safe clinical settings, since severe disease states, rare adverse events, or risky interventions cannot be deliberately sampled for ethical reasons. In such regimes, purely data-driven approximation can fail because the necessary data may never be available. The goal is therefore not simply to fit the available data, but to infer the governing structure that explains why the observed data arise and how the system may behave beyond what has been measured.

Therefore, for real-world, high-stakes applications, discovering governing laws can be highly beneficial, while failing to understand them can be dangerous. In climate modeling, medicine, robotics, materials design, and engineering control, a model may give accurate predictions in familiar regimes but fail silently under domain or distribution shift, leading to incorrect conclusions or unsafe decisions. Conversely, identifying the governing natural laws or conserved structures can produce models that are more stable, reliable, and transferable beyond the observed data. Law discovery therefore serves an important role that prediction alone cannot: it converts data into governing principles.

Machine learning and AI provide a new opportunity to accelerate this process of physical law discovery. Modern computational methods can search over large spaces of candidate equations, learn latent coordinates from high-dimensional observations, infer dynamics from sparse sensors, estimate uncertainty, and combine mechanistic structure with flexible

representation learning. Yet the current workflow for discovering governing laws remains far from optimized. Much of scientific modeling still relies on brute-force simulation, manual coordinate selection, hand-designed features, or ad hoc model selection. The central question motivating this dissertation is how to use machine learning not merely to approximate data, but to discover interpretable physical structure from limited, noisy, and partially observed measurements.

Despite its importance, data-driven physical law discovery remains challenging. The correct state variables may be hidden, the governing equations may be unknown, measurements may be indirect or incomplete, spatial domains may be irregular, and uncertainty in the discovered model may be difficult to quantify.

This dissertation develops a sequence of methods for discovering governing structure from increasingly realistic scientific data. The progression moves from latent law discovery in low-dimensional systems, to sparse sensing of high-dimensional dynamics, to uncertainty quantification, and finally to mesh-free and distributional formulations. Together, these chapters address a common question: how can machine learning recover physical laws when the relevant coordinates are hidden, observations are sparse or noisy, and the resulting model must remain reliable beyond interpolation?

Chapter 2 develops Bayesian data-driven discovery of coordinates, governing equations, and fundamental constants [191]. Classical sparse identification methods often assume that the correct state variables are directly observed. In many scientific settings, however, the measured variables are not the intrinsic coordinates in which the governing laws take a simple form. This chapter addresses this problem by combining autoencoder-based representation learning with sparse equation discovery and Bayesian uncertainty quantification. The goal is not only to fit trajectories, but to infer a coordinate system in which the dynamics become parsimonious, interpretable, and physically meaningful. By placing sparsity-promoting Bayesian priors over the candidate governing equations, the method quantifies uncertainty over both the discovered structure and its coefficients. This provides a principled framework for learning latent coordinates, governing equations, and physical constants jointly from noisy and limited data.

Chapter 3 extends sparse law discovery to high-dimensional spatiotemporal systems observed through only a small number of sensors [104]. Many physical systems of interest, such as fluids, climate fields, and video-based dynamics, evolve in high-dimensional spaces where full-state measurements are expensive or unavailable. This chapter introduces sparse identification of nonlinear dynamics and Koopman operators with shallow recurrent decoder networks. The key idea is to use sensor time histories to construct an informative latent representation, reconstruct the full field through a shallow decoder, and impose parsimonious dynamical structure on the latent state. In the nonlinear case, the latent dynamics are constrained by SINDy; in the linear case, the model recovers a Koopman-style representation. This chapter therefore connects sparse sensing, latent dynamics, and interpretable model discovery, showing that governing equations can be identified even when the system is only partially observed.

Chapter 4 studies the reliability of sparse model discovery through the convergence of uncertainty estimates in ensemble and Bayesian methods [98]. Discovering a sparse governing equation is not sufficient if the model cannot indicate which terms are reliable and which may be artifacts of noise, limited data, or numerical differentiation. This chapter analyzes uncertainty quantification for sparse identification, with particular attention to ensemble sparse regression and Bayesian sparse model discovery. The work shows how bootstrapping-based ensemble methods can provide computationally efficient uncertainty estimates and, under suitable conditions, converge toward correct variable selection. This chapter establishes uncertainty quantification as a necessary component of scientific model discovery: a discovered equation should come not only with coefficients, but also with confidence in its structure.

Chapter 5 removes another major restriction of classical sparse identification: the dependence on structured meshes and uniform sampling [102]. Many real scientific measurements are collected from irregular sensor locations, moving probes, adaptive experiments, or incomplete spatial domains. Standard PDE discovery pipelines often require gridded data and finite-difference derivative estimates, which can become inaccurate or inapplicable in these settings. This chapter introduces mesh-free sparse identification of nonlinear dynamics,

using neural network approximation and automatic differentiation to compute candidate derivatives from arbitrarily sampled spatiotemporal data. This makes sparse PDE discovery applicable to irregular geometries, nonuniform temporal sampling, scarce measurements, and noisy observations. The chapter broadens the practical scope of SINDy-style discovery from idealized simulation data to more realistic scientific sensing environments.

Chapter 6 develops uncertainty quantification for shallow recurrent decoder networks in sparse sensing via engression [101]. While SHRED-based models can reconstruct high-dimensional fields from very few sensors, deterministic reconstructions are insufficient when the system is stochastic, high-frequency, partially observed, or intrinsically ambiguous from the available measurements. This chapter introduces UQ-SHRED, a distributional sparse-sensing framework that learns the conditional distribution of the full spatiotemporal state given sparse sensor histories. By injecting stochastic noise into the model input and training with an energy-score objective, the method produces uncertainty-aware reconstructions without requiring ensembles, additional networks, or expensive Bayesian inference. This extends sparse sensing from point prediction to calibrated conditional distribution learning, allowing the model to express what is known, what is uncertain, and where sparse measurements are insufficient.

Across these chapters, we advance a unified view of scientific machine learning: artificial intelligence should not only approximate observed data, but also discover coordinates, identify governing laws, quantify uncertainty, and adapt physical model discovery to the sparse, noisy, irregular, and partially observed data regimes encountered in real scientific problems.

2

Bayesian Data-Driven Discovery of Coordinates, Governing Equations, and Fundamental Constants

2.1 Introduction

Calculus-based models fundamentally relate the rates of change of quantities of interest in time and space through differential and partial differential equations. From population models to turbulence, physics and engineering principles are rooted in such governing equations. In the modern era of big data, there is a growing demand to transform rich spatio-temporal data into descriptive physical models in an automated, data-driven fashion. Video data, for example, contain optical snapshots of an observed dynamical system described by some specific governing equations. To understand the underlying physics, it is important not only to identify the equations, but to discover dependent state variables (sparse representations) directly from the video frames. To discover such an underlying sparse representation, it is essential to find a correct coordinate transformation (latent space or manifold) to compress the data into a low-dimensional space. Principal Component Analysis (PCA) is often applied to obtain a low-dimensional subspace with linearity constraints. For nonlinear transformations, Autoencoders with neural networks are frequently applied as a nonlinear extension of PCA [125, 10]. To identify the governing equation given the latent sparse representations, one can apply data-driven methods via sparse regression [160, 238, 234]. The sparse regression-based model discovery enables a computationally efficient way of identifying governing equation with convergence guarantees [314]. In this case of physics discovery from videos, Champion et

al. [46] propose SINDy autoencoders which can jointly discover on coordinates and equations for synthetic video data. More recently, Chen et al. [53] introduce the Neural State Variables learning from video data, which enables automated discovery of the latent state variables from the underlying dynamical system. [53].

The automated discovery of coordinates and equations in real video is significantly more challenging compared to the prior works [47, 53]. The challenges are coming from two parts. First, real raw videos are much more noisy compared to simulated data. For example, the required temporal derivatives are not directly available from the real video recordings. To resolve the missing temporal derivatives, one has to find an approximation for the temporal derivatives from discrete video frames, which is frequently numerically unstable and will inject a high level of noise. Moreover, lighting effects alone can greatly compromise derivate estimates. Second, the number of training data for the real video dataset is very limited in comparison with synthetic videos [46]. For instance, in Sec. 2.4.2, we have a training dataset with only 382 video snapshots, which is noticeably small for most deep learning video tasks. These pragmatic constraints restrict and force the algorithm to deal with a statistically insufficient sample size. Thus, automated discovery from real video data is much more challenging due to being in the low-data and high-noise limit.

In this case, Bayesian sparse regression methods have shown significant advantages in the low-data and high-noise regime [196, 107, 242]. Bayesian methods can not only achieve better results with limited sample size, but could also perform robust discovery with uncertainty quantification in the high-noise environments. The spike-and-slab prior [137] (e.g. Bernoulli-Gaussian, Bernoulli-Laplace [5]) with hierarchical Bayesian settings has a proven success for both sparse variable selection and uncertainty quantification [227]. In data-driven model discovery, Bayesian SINDy [126] with the spike-and-slab prior has also shown an advantage for correct model identification in the case study of Lynx-hare population [123] under the very low-data limit. However, a significant limitation of Bayesian methods comes from its high computational cost that hinders both speed and scalability. The MCMC-based hierarchical Bayesian model sampling requires a considerably long run because the underlying stochastic binary search grows exponentially with the number of parameters. Additionally, it is very

costly to compute the full gradient of the entire video dataset for the MCMC sampling. Therefore, even if the Bayesian methods are much more powerful in the low-data regime, it is highly nontrivial to design a feasible Bayesian solution for the discovery of governing equations and coordinate systems given the computationally intractability.

In this paper, we propose Bayesian SINDy autoencoders that extends SINDy Autoencoder [46] into a Bayesian learning framework. Bayesian SINDy autoencoders can perform a joint discovery for governing equations and coordinate systems in a computationally tractable manner. Specifically, we apply the Spike-and-slab Gaussian-Laplace (SSGL) prior to the intermediate SINDy module to accelerate the sparse identification process of governing equation discovery. To resolve the computational burden arising from the hierarchical Bayesian model, we consolidate the Bayesian sampling procedure via Stochastic Gradient Langevin Dynamics (SGLD) with an adaptive empirical Bayesian variable selection method using Expectation–maximization. Instead of computing the full batch gradient, SGLD evaluates mini-batch gradients with injected random Gaussian noise, which is theoretically valid to generate Langevin-based proposal distribution [289, 49]. The mini-batch gradient learning naturally fits into the training of deep neural networks and relaxes the scalability issue at the same time. On the other hand, the adaptive Bayesian Expectation–maximization variable selection (EMVS) performs variable selection by optimizing the latent inclusion probability of each variable, which avoids the previously required lengthy binary stochastic search. Adapting these two ideas into the SINDy module, we can simultaneously perform an accelerated governing equation identification under the low-data, high-noise limit with a trustworthy uncertainty quantification through the power of Bayesian estimation.

In our numerical experiments, Bayesian SINDy autoencoders not only expedite the discovery process with unchanged sample sizes but also maintain robust performance with limited data. In addition to the model discovery in synthetic datasets, from a real video data on a single moving pendulum, we obtain correct governing equation discovery with a close estimate of the gravity constant $\hat{g} = -9.876$ with only 390 data snapshots. With precise understanding of the underlying physics, Bayesian SINDy autoencoder enables explainable, trustworthy, and robust video predictions. In summary, the contribution of this paper is

threefold:

1. We propose the Bayesian SINDy autoencoder with Spike-and-slab Gaussian-Laplace prior to accelerated sparse inference under low-data and high-noise environments.
2. We utilize Stochastic Gradient Langevin Dynamics to perform posterior sampling in Bayesian SINDy autoencoder with valid uncertainty estimations.
3. We conduct extensive experiments with successful joint discoveries on coordinate systems and governing equations for both synthetic and real datasets. Remarkably, we achieve a correct physics discovery with a 14 seconds recording on our experiment on a pendulum.

2.2 Background

The current work is built upon two primary mathematical innovations: (i) sparse regression used in the SINDy algorithm, including learning latent representations, and (ii) Bayesian learning with sparse priors. A quick review of each is given in order to better inform the reader how they are combined into our Bayesian SINDy autoencoder framework.

2.2.1 Sparse identification of nonlinear dynamics

We review the *sparse identification of nonlinear dynamics* (SINDy) [32] algorithm, which utilizes sparse regression to identify the latent dynamical system from snapshot data. SINDy takes snapshot data $\mathbf{x}(t) \in \mathbb{R}^n$ and aims to discover the underlying dynamical system

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)). \quad (2.1)$$

The snapshots are collected by measurements at time $t \in [t_1, t_m]$, and the function $\mathbf{f}$ characterizes the dynamics. Assuming the temporal derivatives of the snapshot are available from

data, SINDy forms data matrices in the following way:

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}(t_1) \\ \mathbf{x}(t_2) \\ \vdots \\ \mathbf{x}(t_m) \end{pmatrix}, \quad \dot{\mathbf{X}} = \begin{pmatrix} \dot{\mathbf{x}}_1(t_1) \\ \dot{\mathbf{x}}_1(t_2) \\ \vdots \\ \dot{\mathbf{x}}_1(t_m) \end{pmatrix},$$

with $\mathbf{X}, \dot{\mathbf{X}} \in \mathbb{R}^{m \times n}$. The candidate function library is constructed by p candidate model term θ_j 's that $\boldsymbol{\Theta}(\mathbf{X}) = [\theta_1(\mathbf{X}) \cdots \theta_p(\mathbf{X})] \in \mathbb{R}^{m \times p}$. A common choice of candidate functions are polynomials in $\mathbf{x}$ targeting common canonical models of dynamical systems [114]. The Fourier library is also very common with $\sin(\cdot)$ and $\cos(\cdot)$ terms. In summary, we build a model between $\mathbf{X}$ and $\dot{\mathbf{X}}$ that

$$\dot{\mathbf{X}} = \boldsymbol{\Theta}(\mathbf{X})\boldsymbol{\Xi}$$

where the unknown matrix $\boldsymbol{\Xi} = (\xi_1 \ \xi_2 \ \cdots \ \xi_n) \in \mathbb{R}^{p \times n}$ is the set of coefficients. The sparse inference on $\boldsymbol{\Xi}$ enables sparse identification of the dynamical system $\mathbf{f}$. For high-dimensional systems, the goal is to jointly identify a low-dimensional state $\mathbf{z} = \varphi(\mathbf{x})$ with dynamics $\dot{\mathbf{z}} = \mathbf{g}(\mathbf{z})$. The standard SINDy approach uses a sequentially thresholded least squares algorithm to perform sparse inference [32], which is a proxy for ℓ_0 optimization [320] with convergence guarantees [314]. Small et al [248] and Yao and Bollt [306] previously formulated the dynamical system identification without sparsity constraints. These methods provide a computationally efficient counterpart to other model discovery frameworks [241, 283, 292].

In an alternative approach, Hirsh et al. [126] utilize Bayesian sparse regression techniques in model discovery, which have improved the robustness of SINDy under high-noise and low-data settings. Bayesian sparse inference models the sparsity with various probabilistic priors $p(\boldsymbol{\Xi})$ like the Spike-and-slab [137], regularized horseshoe [44, 45], Laplace priors [213, 268] and so on. After defining the Bayesian likelihood function with sparsifying priors, Bayesian SINDy generates posterior distribution via the No-U-Turn MCMC sampler [130]. Since MCMC for sparse inference can be extremely computationally demanding, [227, 228] propose coordinate descent sparse inference methods via Spike-and-slab prior, targeting the mode

detection. As an approximation to Bayesian inference, Fasel et al. [90] combine ensembling techniques via bootstrapping to perform uncertainty estimation and accelerated variable selection for system identification.

SINDy has been widely applied to model discovery for many scientific scenarios including fluid dynamics [180, 181, 179, 113, 69, 38], nonlinear optics [254], turbulence closures [18, 19, 240], ocean closures [312], chemical reaction [131], plasma dynamics [67, 4, 144], structural modeling [162], and for model predictive control [140]. There are many extensions of SINDy, including the identification of partial differential equations [234, 238], multiscale physics [47], parametrically dependent dynamical models [233], time-dependent PDEs [52], switching dynamical systems [188], rational function nonlinearities [189, 139], control inputs [140], constraints on symmetries [180], control for stability [143], control for robustness [4, 239, 223, 224, 194, 194], stochastic dynamical systems [28, 37], and multidimensional approximation on tensors [106].

A related and important extension to the SINDy framework is the SINDy autoencoder [46, 9], which embeds SINDy into the training process of deep autoencoders. The SINDy autoencoder achieves remarkable performances on high-dimensional synthetic data to jointly discover coordinate systems and governing equations. Due to over-parametrization and non-convexity, deep neural networks do not generally have explainable guarantees for inference and predictions. Dynamical system learning and forecasting is typically an extrapolatory problem by nature. Therefore, the interpretability of models is very important to understand. In this case, SINDy autoencoder is satisfactory due to the transparency of the learning process, and the parsimony in the dynamical system modeling [159, 157]. Specifically, the encoder and decoder focus on the specific task of learning only a coordinate transformation, with the SINDy layer targeting the inference of latent dynamical systems. The SINDy autoencoder can not only identify the governing equation from high-dimensional data but can also perform trustworthy predictions based on future dynamics. The limitations of the SINDy autoencoder are exactly what our Bayesian framework addresses.

2.2.2 Bayesian and sparse deep learning

Bayesian deep learning achieves outstanding success in various machine learning tasks like computer vision [117], physics-informed modeling [304], and complex dynamical system control [288]. From previous works, the main advantages of Bayesian deep learning come from two parts. First, the Bayesian framework can unify uncertainty quantification in deep learning. This includes the uncertainty from the neural network parameters, task-specific parameters, and exchanging information [281, 1], which is applicable to computer vision [149], spatiotemporal forecasting [297], weather forecasting [280], and so on [324]. Second, the Bayesian framework allows theoretically grounded ensemble neural networks via model averaging [294]. Bayesian neural networks apply weights distribution to neurons [27]. It is important to avoid computational requirements from Bayesian when applying it to deep learning. To perform Bayesian sampling in deep learning, Welling and Teh establish Stochastic Gradient Langevin Dynamics as an MCMC sampler in mini-batch settings [289]. Neal introduces MCMC via Hamiltonian dynamics [203], and Chen et al. extend into deep learning via Stochastic gradient Hamiltonian Monte Carlo (SGHMC) [56, 187]. Other techniques include Nosé–Hoover thermostat [75], replica-exchange SGHMC [70], cyclic SGLD [315], contour SGLD [71], preconditioned SGLD [285], and adaptively weighted SGLD [72]. Variational methods could also help to perform approximate inference for Bayesian deep learning [122, 27].

Sparse deep learning is of emerging interest due to its lowered computational cost and improved interpretability. The discussion on sparse neural networks dates back decades [83, 199]. Glorot et al. use Rectifier Neurons with sparsity constraints to obtain sparse representations [110]. Liu et al. extend to Convolutional neural networks under sparse settings [177]. Then, several follow-up works aimed to train sparse neural networks efficiently [184, 255]. By adapting Bayesian sparse inference methods, the sparse inference process could be accelerated with valid uncertainty quantification via various prior options [260, 73, 286]. Variational methods are also applied for efficient Bayesian inference [8]. There have also been theoretical discussion of Bayesian sparse deep learning [259, 218, 261]. Overall, sparse deep learning has

the potential to make computationally tractable the construction of a Bayesian model.

2.3 Method: Bayesian SINDy Autoencoders

This section presents the SINDy autoencoder with a Bayesian learning process that incorporates sparsifying priors and posterior sampling. We first introduce the Sparse identification of nonlinear dynamics (SINDy) autoencoder in Sec. 2.3.1. Then, we propose a Bayesian learning framework that includes SINDy autoencoder in the likelihood function and specifies various setups for different sparsifying priors. Finally, we discuss Stochastic Gradient Langevin Dynamics (SGLD) to generate posterior samples from the Bayesian learning model.

2.3.1 Likelihood setting for Bayesian SINDy Autoencoders

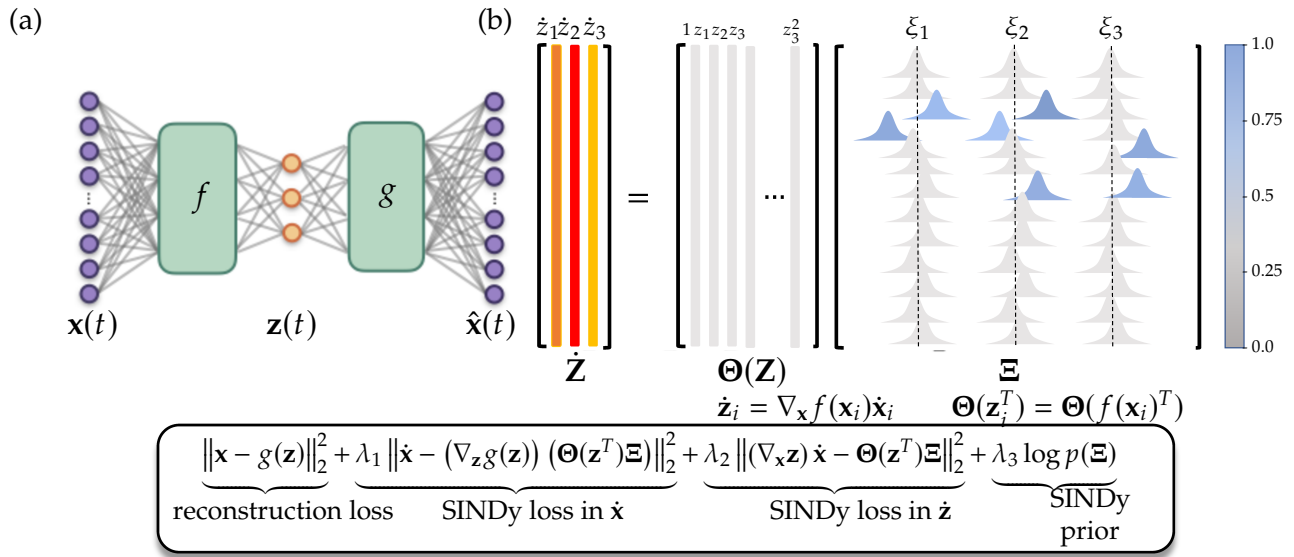


Figure 2.1: General structure of the Bayesian SINDy autoencoder architecture. (a) An autoencoder architecture is used to discover intrinsic coordinates $\mathbf{z}$ from high-dimensional input data $\mathbf{x}$. The encoder $f(\mathbf{x})$ transforms the high-dimensional input into the a low-dimensional space $\mathbf{z}$, and the decoder $g(\mathbf{z})$ reconstructs $\mathbf{x}$ from the low-dimensional subspace. (b) A Bayesian SINDy model infers the underlying dynamics of the latent dimension with uncertainty estimation. The active terms are identified by Ξ . The color represents the inclusion probability $[0.0, 1.0]$ from grey (non-active) to blue (active). The loss function encourages the network to minimize both the reconstruction error and the SINDy loss in $\mathbf{z}$ and $\mathbf{x}$. The Bayesian prior works as a regularization term on Ξ for sparse inference.

The SINDy autoencoder enables a joint discovery of sparse dynamical models and

coordinates. Figure 2.1 (a) provides an overview of an autoencoder. The input data $x(t) \in \mathbb{R}^d$ is mapped by an encoder function $f_{\theta_1}(\cdot)$ to a latent space $z(t) \in \mathbb{R}^{d_z}$, $d_z < d$. This latent space $z(t)$ contains sufficient information to recover $x(t)$ via a decoder function $g_{\theta_2}(\cdot)$.

SINDy autoencoder combines SINDy with autoencoders by constraining the latent space governed by a sparse dynamical system. The encoder function $f_{\theta_1}(\cdot)$ performs coordinate transformation to map the high-dimensional inputs into an appropriate latent subspace. The latent space $z(t) = f_{\theta_1}(x(t))$ has an associated sparse dynamical model governed by

$$\frac{d}{dt}z(t) = \Phi(z(t)) = \Theta(z(t))\Xi, \quad (2.2)$$

where $\Theta(z) = [\theta_1(z), \theta_2(z), \dots, \theta_p(z)]$ is a library of candidate basis functions, and a set of coefficients $\Xi = [\xi_1, \xi_2, \dots, \xi_p]$.

A statistical understanding of the model formulates the SINDy autoencoder as a parametric model $\mathcal{M}_\theta$ where $\theta = \{\theta_1, \theta_2, \Xi\}$ contains all parameters. The likelihood of this model $p(\mathcal{D}|\theta)$ is defined as

$$p(\mathcal{D}|\theta) \propto \exp \left(\|\mathbf{x} - g_{\theta_2}(\mathbf{z})\|_2^2 + \lambda_1 \|\dot{\mathbf{x}} - (\nabla_{\mathbf{z}} g_{\theta_2}(\mathbf{z})) (\Theta(\mathbf{z}^T)\Xi)\|_2^2 + \lambda_2 \|(\nabla_{\mathbf{x}} \mathbf{z}) \dot{\mathbf{x}} - \Theta(\mathbf{z}^T)\Xi\|_2^2 \right). \quad (2.3)$$

The log-likelihood of this statistical model is similar to the setting in [46]. In order to promote sparsity on Ξ , we consider two sets of priors in what follows.

Laplace prior. The Laplace prior can be understood as a Bayesian LASSO [268, 213]. We define the Laplace prior such that

$$\Xi_j \sim \mathcal{L}(0, v_0), \quad (2.4)$$

where $\mathcal{L}(\cdot, \cdot)$ denotes the Laplace distribution defined as $f(\Xi_i; 0, v_0) = \frac{1}{2v_0} \exp\left(-\frac{|\Xi_i|}{v_0}\right)$. We can see the equivalence of the Laplace prior and the LASSO in the negative log-likelihood, where $\frac{1}{v_0} \|\Xi_i\|_1$ is included as a regularizer.

Spike-and-slab Gaussian-Laplace (SSGL) prior. We define the SSGL prior as

$$\Xi_j | \gamma_j \sim (1 - \gamma_j) \mathcal{L}(0, \sigma v_0) + \gamma_j \mathcal{N}(0, \sigma^2 v_1), \quad (2.5)$$

where γ_j is a binary variable, $\Xi \in \mathbb{R}^p$, $\sigma, v_0, v_1 \in \mathbb{R}$, $\mathcal{L}(\cdot, \cdot)$ denotes a Laplace distribution and $\mathcal{N}(\cdot, \cdot)$ denotes a Normal distribution. We assign a Bernoulli prior to $\gamma \sim \text{Ber}(\delta)$, $\delta \in [0, 1]$. The prior for θ_1, θ_2 is specified with a Gaussian, which is equivalent to an ℓ_2 regularization implementation-wise. For simplicity, we set δ, σ, v_0, v_1 as tunable hyperparameters.

Prior selection. In general, there is no optimal Bayesian prior for all statistical models since every prior has its own advantages and drawbacks. Therefore, the prior selection typically requires an assessment of both theory and experimental outcomes. If one selects the prior of Ξ to be a Laplace distribution, the setting will be identical to the basic SINDy autoencoder model, which is equivalent to adding a ℓ_1 regularization term (c.f. Eqn. (7) and Figure 1 in [46]). Even if the Laplace prior has a benefit in computation, its performance suffers for small sample sizes and large observation noises. Different from the Laplace prior, if one selects the SSGL prior for Ξ , it typically requires a slightly increased cost in computation. However, the SSGL prior typically has dominating performances for cases with very large noise and limited sample size, which is preferred in our case to learn the actual video data.

Bayesian formulation. Using the Bayes formula, we can construct the posterior distribution from the likelihood function and prior that

$$\pi(\theta, \gamma | \mathcal{D}) \propto p(\mathcal{D} | \theta) p(\theta_1) p(\theta_2) p(\Xi | \gamma) p(\gamma). \quad (2.6)$$

We aim to depict the posterior distribution $p(\Xi | \mathcal{D})$ via the joint distribution $p(\theta | \mathcal{D})$ under the sparsifying SSGL prior. The approximation of $p(\Xi | \mathcal{D})$ is accessible from posterior samples of $p(\theta | \mathcal{D})$ when dropping θ_1, θ_2 . In the setting of deep neural networks, we can sample the posterior distribution via Stochastic Gradient methods using mini-batches. The mini-batch setting not only naturally fits into the training of deep neural networks but also could

accelerate the Bayesian posterior sampling process.

2.3.2 Stochastic Gradient Langevin Dynamics

To perform posterior sampling in mini-batch settings, Stochastic Gradient Langevin Dynamics (SGLD) is a popular method that combines stochastic optimization and Langevin Dynamics [289]. Denote the learning rate at epoch t by $\epsilon^{(t)}$ which decreases to zero, and the dataset $\mathcal{D} = \{d_i\}_{i=1}^N$. The mini-batch setting estimates the gradient $\nabla_{\theta} L(\theta)$ from a subset (batch) $\mathcal{B} = \{d_i\}_{i=1}^n$. The injected noise from mini-batches facilitates the generation of posterior samples while reducing the high computational cost in computation for full-batch gradients. We follow the classical SGLD setting where

$$\Delta\theta_{t+1} = \frac{\epsilon^{(t)}}{2} \left(\nabla \log p(\theta_t) + \frac{N}{n} \sum_{i=1}^n \nabla \log p(X_i|\theta_t) \right) + \eta_t, \quad \eta_t \sim \mathcal{N}(0, \epsilon^{(t)}). \quad (2.7)$$

Here, $p(\theta_t)$ denotes the prior specified in Eqn. (2.4), (2.5) and $p(X_i|\theta_t)$ denotes the data likelihood. Prior works study the asymptotic convergence of SGLD to the target distribution, which validates SGLD for posterior sampling in theory [317, 265, 129]. An advantageous property of SGLD in posterior sampling is that with decaying step size $\epsilon^{(t)}$, SGLD automatically transfers from a stochastic optimization algorithm to a posterior sampling procedure (c.f. Sec. 4.1 [289]). The Metropolis-Hasting correction can be ignored since the rejection rate for sampling goes to zero asymptotically, resulting from $\epsilon^{(t)} \rightarrow 0$ when $t \rightarrow \infty$. The discretization error similarly decreases as ϵ goes to zero.

Cyclical SGLD. The cyclical SGLD method [315] consists of exploration and sampling stages via a cyclical step-size schedule for $\epsilon^{(t)}$. In the training of deep autoencoders, the optimization process is highly non-convex with a very complex loss landscape. The cyclical step-size schedule helps to explore the parameter space when ϵ is large as well as sample local mode when ϵ is small. It is also possible to understand cyclical SGLD from a parallel SGLD perspective [70]. We apply this idea in our experiments to perform better inference.

2.3.3 Empirical Bayes Variable Selection in SINDy Autoencoder

The empirical bayesian method infers prior hyperparameters from data. In this case, we aim to optimize γ (ignoring the uncertainty) and sample $\theta|\mathcal{D}$. The posterior distribution of $\Xi|\mathcal{D}$ can be derived from posterior samples of $\theta|\mathcal{D}$.

Using the mini-batch setting, the posterior distribution follows

$$\pi(\theta, \gamma|\mathcal{B}) \propto p(\mathcal{B}|\theta)^{\frac{N}{n}} p(\theta_1) p(\theta_2) p(\Xi|\gamma) p(\gamma). \quad (2.8)$$

The term $p(\mathcal{B}|\theta)$ can be evaluated from the loss of SINDy autnencoder for the current mini-batch $\mathcal{B}$; the term $p(\theta_1), p(\theta_2)$ can be computed from the ℓ_2 loss; the term $p(\Xi|\gamma)$ can be computed from Eqn. (2.5); and the term $p(\gamma)$ can be known from the Bernoulli prior setting.

As a binary variable, γ is difficult to be optimized for due to non-continuity and non-convexity. An important trick to perform the optimization on γ is to alternatively optimize the adaptive posterior mean $\mathbb{E}_{\gamma|\theta^{(k)}, \mathcal{D}} [\pi(\theta, \gamma|\mathcal{D})]$ which treats γ as a latent variable. Following previous works with similar settings [70, 227], we could indirectly evaluate $\pi(\theta, \gamma|\mathcal{D})$ by a strict lower bound $Q(\cdot | \cdot)$ that

$$Q(\theta|\theta^{(k)}) = \mathbb{E}_{\mathcal{B}} \left[\mathbb{E}_{\gamma|\theta^{(k)}, \mathcal{D}} [\log \pi(\theta, \gamma|\mathcal{B})] \right] \leq \log \mathbb{E}_{\gamma|\theta^{(k)}, \mathcal{D}} [\mathbb{E}_{\mathcal{B}} [\pi(\theta, \gamma|\mathcal{B})]]. \quad (2.9)$$

The inequality holds by Fubini's theorem and Jensen's inequality (c.f. Eqn. (7) in [70]).

The variable $Q(\theta|\theta^{(k)})$ can be decomposed into

$$Q(\theta|\theta^{(k)}) = \frac{N}{n} \log \pi(\mathcal{B}|\theta) - \|\theta_1\| - \|\theta_2\| - \sum_{i \in \Xi} \left[\frac{|\Xi_i| \kappa_{i0}}{\sigma} + \frac{\Xi_i^2 \kappa_{i1}}{2\sigma^2} \right] + \sum_{i \in \Xi} \log \left(\frac{\delta}{1 - \delta} \right) \rho_i + C, \quad (2.10)$$

where $\kappa_{i0} = \mathbb{E}_{\gamma|\theta^{(k)}, \mathcal{D}} \left[\frac{1}{v_0(1-\gamma_i)} \right]$, $\kappa_{i1} = \mathbb{E}_{\gamma|\theta^{(k)}, \mathcal{D}} \left[\frac{1}{\gamma_i} \right]$, $\rho_i = \mathbb{E}_{\gamma|\theta^{(k)}, \mathcal{D}} [\gamma_i]$, and $C \in \mathbb{R}$ is a constant. Notice here γ is treated as a latent variable, and we only consider the expectation given the conditional distribution of γ given $\theta^{(k)}, \mathcal{D}$. The ρ_i could be considered as an inclusion probability estimate of Ξ_i . In this way, ρ softens γ from a binary variable into a continuous

one.

The term κ performs an elastic net-like approach which adaptively optimizes the ℓ_1 and ℓ_2 coefficients. Suppose Ξ_i is identified with a high probability to be a sparse variable (e.g., $\rho_i < 0.05$), the term κ_{i0} will be large, strengthening the sparsity constraints. Otherwise, if Ξ_i is identified as a non-sparse variable (e.g. $\rho_i > 0.95$), the term κ_{i1} will be small, and the ℓ_2 constraint will be dominated instead.

Stochastic Approximation from Expectation Maximization of ρ and κ

We could derive an asymptotically correct posterior distribution on $\pi(\theta, \kappa, \rho)$ following the steps in [70]:

1. Sample θ from $Q(\cdot)$ that

$$\theta^{(k+1)} = \theta^{(k)} + \eta^{(k)} \nabla_{\theta} Q(\rho^{(k)}, \kappa^{(k)} | \mathcal{B}^{(k)}) + \mathcal{N}(0, 2\eta^{(k)}) \quad (2.11)$$

2. Perform stochastic approximation to latent variables ρ, κ that

$$\rho^{(k+1)} = (1 - \omega^{(k+1)})\rho^{(k)} + \omega^{(k+1)}\tilde{\rho}^{(k+1)}, \quad (2.12)$$

$$\kappa^{(k+1)} = (1 - \omega^{(k+1)})\kappa^{(k)} + \omega^{(k+1)}\tilde{\kappa}^{(k+1)}. \quad (2.13)$$

Here, $\tilde{\rho}^{(k+1)}, \tilde{\kappa}^{(k+1)}$ is the Expectation Maximization (EM) estimation of ρ, κ given iteration k [227]. The EM estimation of the inclusion probability ρ_i is

$$\tilde{\rho}_i^{(k+1)} = \mathbb{E}_{\gamma|\theta^{(k)}, \mathcal{B}}[\gamma_i] = P(\gamma_i = 1 | \Xi^{(k)}) = \frac{a_i}{a_i + b_i}, \quad (2.14)$$

where $a_i = \pi(\Xi_i^{(k)} | \gamma_i = 1)P(\gamma_i = 1 | \delta)$ and $b_i = \pi(\Xi_i^{(k)} | \gamma_i = 0)P(\gamma_i = 0 | \delta)$. In our case, the term $\pi(\Xi_i^{(k)} | \gamma_i = 1)$ is the probability of $\Xi_i^{(k)}$ given the Gaussian prior distribution, and the term $\pi(\Xi_i^{(k)} | \gamma_i = 0)$ is the probability of $\Xi_i^{(k)}$ from the Laplace distribution. The latter term $P(\gamma_i = 1 | \delta) = \delta$ given Bernoulli prior in the previous setting. Following a similar process, for

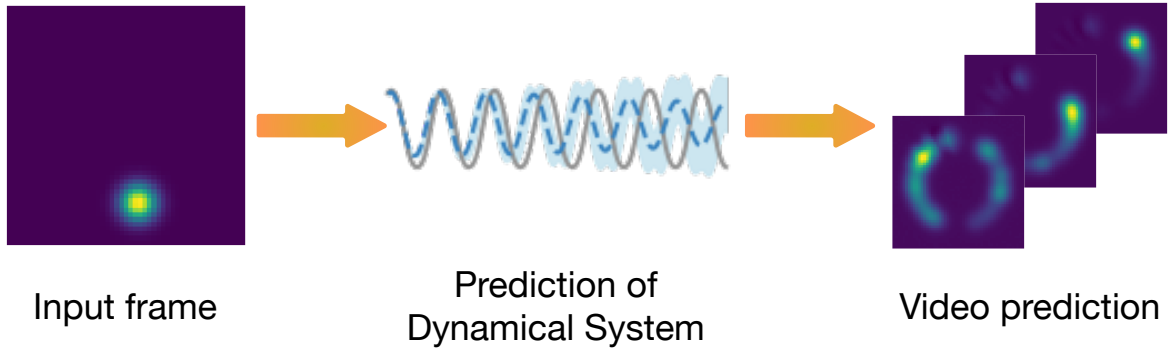


Figure 2.2: Video prediction with uncertainty estimation from latent dimension in pixel space.

κ , we have

$$\tilde{\kappa}_{i0}^{(k+1)} = \mathbb{E}_{\gamma|\theta^{(k)}, \mathcal{B}} \left[\frac{1}{v_0(1 - \gamma_i)} \right] = \frac{1 - \rho_i}{v_0}, \quad (2.15)$$

$$\tilde{\kappa}_{i1}^{(k+1)} = \mathbb{E}_{\gamma|\theta^{(k)}, \mathcal{B}} \left[\frac{1}{v_1 \gamma_i} \right] = \frac{\rho_i}{v_1}. \quad (2.16)$$

2.3.4 Prediction with Bayesian SINDy Autoencoder

Bayesian SINDy autoencoder has a trustworthy application in our application of video prediction whereby we learn the dynamics and coordinates of the latent space. Based on an inference process established in previous sections, the video prediction can precisely understand the underlying dynamical system, which allows accurate, robust, and interpretable future forecasting. Additionally, the Bayesian framework enables precise uncertainty quantification for video prediction from posterior samples. We visualize the prediction process in Fig. 2.2. In this case of a pendulum video, the uncertainty of video prediction grows with larger t , and gradually fails to predict by showing a ring-like prediction.

From the procedure in Sec. 3.3, we can generate samples from $p(\theta|\mathcal{D})$, which is the posterior distribution of neural network parameters θ given observed data $\mathcal{D}$. Suppose the posterior samples are $\Xi^1, \Xi^2, \dots, \Xi^m \sim p(\theta|\mathcal{D})$. The full posterior predictive distribution is

defined as

$$\begin{aligned} p(\hat{x}(t)|x_0, \mathcal{D}) &= \int p(\hat{x}(t)|\theta, x_0)p(\theta|\mathcal{D})d\theta \\ &= \int p(\hat{x}(t)|\theta_2, \hat{z}(t))p(\hat{z}(t)|\Xi, \hat{z}(0))p(\hat{z}(0)|\theta_1, x_0)p(\theta_1, \theta_2, \Xi|\mathcal{D})d\theta \end{aligned} \quad (2.17)$$

Therefore, we can approximately generate samples from $p(\hat{x}(t)|x_0, \mathcal{D})$ using Monte Carlo estimation. For simplicity, we only consider the Maximum likelihood Estimation $\theta_1^{\text{MLE}}, \theta_2^{\text{MLE}}$. We have the following process:

1. From the input image x_0 , compute $z(0) = f_{\theta_1^{\text{MLE}}}(x_0)$.
2. From $z(0)$, using posterior samples of Ξ , generate $\hat{z}^{(i)}(t)$ samples via

$$z^{(i)}(t) = z(0) + \int_0^t \Theta(z(t'))\Xi^{(i)} dt'.$$

3. From samples of $z^{(i)}(t)$, generate $\hat{x}^{(i)}(t) = g_{\theta_2^{\text{MLE}}}(z^{(i)}(t))$.

2.4 Experiments

In the following subsections, we conduct four case studies on Bayesian SINDy autoencoder for governing equations and coordinate system discovery for video data. In Sec. 2.4.1, we study similar cases to those in [46] with synthetic high-dimensional data generated from the Lorenz system, reaction-diffusion, and a single pendulum. We explore the Laplace prior for chaotic Lorenz system and the SSGL prior for reaction-diffusion and single pendulum. In Sec. 2.4.2, we study real video data that consists of 390 video frames of a moving rod. This setting is particularly challenging due to the high dimensionality in video data, noisy observation, missing temporal derivatives, and prior setting for correct learning. In both experiments on synthetic and real video data, we observe the Bayesian SINDy autoencoder can accurately perform nonlinear identification of dynamical system under the correct setting of preprocessing, prior, and training parameters. the experiments are implemented and run on a single NVIDIA GeForce RTX 2080 Ti with maximum 11264 MB.

2.4.1 Learning Physics from Synthetic Video Data

Chaotic Lorenz System via the Laplace prior

We consider the chaotic Lorenz system in the following

$$\begin{cases} \dot{z}_1 = -\sigma z_1 + \sigma z_2 \\ \dot{z}_2 = \rho z_1 - z_2 - z_1 z_3 \\ \dot{z}_3 = z_1 z_2 - \beta z_3, \end{cases} \quad (2.18)$$

where $z = [z_1, z_2, z_3] \in \mathbb{R}^3$ and σ, ρ, β are constants. The Lorenz system is very representative of the chaotic and nonlinear system, which is an ideal example of applying model discovery techniques. In the numerical simulation, we first set $\sigma = 10, \rho = 28, \beta = -2.7$. We only generate partial Lorenz via time range $t = [0, 5]$ with $\Delta t = 0.02$ for 1024 different Lorenz systems from random initial conditions. The initial condition follows a uniform distribution centered at $[0, 0, 25]$ with width $[36, 48, 41]$ respectively.

From the underlying dynamical system, we create a high-dimensional dataset via six fixed spatial models given by Legendre polynomials that $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_6 \in \mathbb{R}^{128}$. We transfer from the low-dimensional dynamical system into a high-dimensional dataset via the following rule:

$$\mathbf{x}(t) = \mathbf{u}_1 z_1(t) + \mathbf{u}_2 z_2(t) + \mathbf{u}_3 z_3(t) + \mathbf{u}_4 z_1(t)^3 + \mathbf{u}_5 z_2(t)^2 + \mathbf{u}_6 z_3(t)^3. \quad (2.19)$$

We set the autoencoder with latent dimension $d = 3$ corresponding to the latent system in the 3D coordinate system with z_A, z_B, z_C . We include polynomials with the highest order 3 composing a library $[1, z_A, z_B, z_C, z_A^2, z_A z_B, z_A z_C, z_B^2, z_B z_C, z_C^2, z_A^3, z_A^2 z_B, z_A^2 z_C, \dots, z_C^3]$. Via the autoencoder, we wish to identify the correct active terms as well as the value of the coefficients. The coefficient of Ξ is uniformly initialized from constant 1. The loss coefficients are $\lambda_1 = 0.0, \lambda_2 = 1 \times 10^{-4}$. For the encoder and decoder, we use the sigmoid activation function with widths $[64, 32]$. For optimization, we select Adam optimizer with learning rate 1×10^{-3} and the batch size to be 1024. For the Laplace prior setting, we set $\lambda_3 = 1 \times 10^{-5}$.

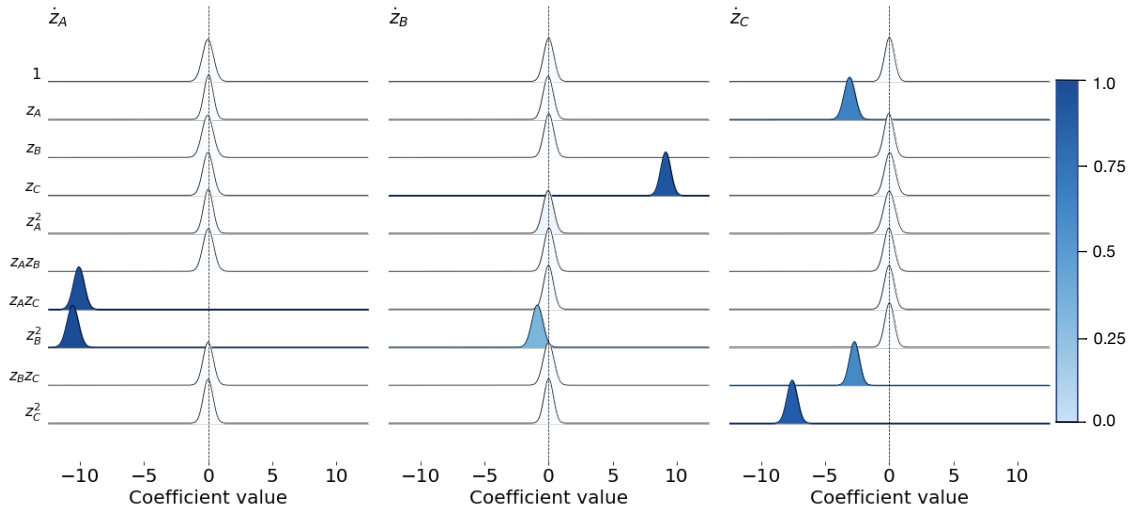


Figure 2.3: Bayesian estimation and uncertainty quantification visualization of SINDy coefficient for Lorenz system under Laplace prior. The color bar represents the inclusion probability estimate given the coefficient magnitude. The color represents the inclusion probability $[0.0, 1.0]$ from white (non-active) to dark blue (active).

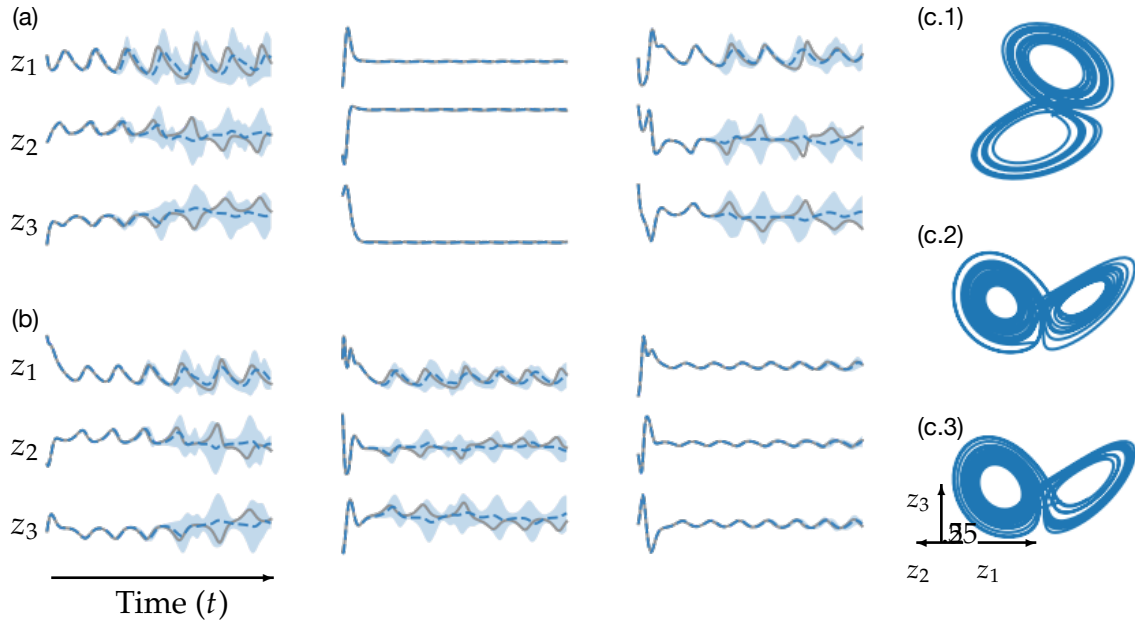


Figure 2.4: (a) Trajectory prediction for **in-distribution** data with uncertainty quantification generated by posterior samples from Bayesian inference on SINDy coefficient. (b) Trajectory prediction for **out-of-distribution** data with uncertainty quantification generated by posterior samples from Bayesian inference on SINDy coefficient. (c.1) Discovered Lorenz systems and (c.2) transformation to the standard space. (c.3) Lorenz system generated from the ground truth.

By training with 5,000 epochs following the setting from [46], in terms of the error metrics, the best test error of the decoder reconstruction achieves 2×10^{-5} of the fraction of the variance from the input. The fraction of unexplained variances are 2×10^{-4} for the reconstruction of $\dot{z}$, and 1.3×10^{-3} for the reconstruction of $\dot{x}$. Notice here decoder reconstruction is better compared to SINDy autoencoder without uncertainty quantification, but the reconstruction of $\dot{z}$ is slightly worse compared to point estimation for SINDy autoencoder.

The coefficient estimate from the Bayesian SINDy autoencoder is shown in Fig. 2.3. From the figure, we can observe the uncertainty quantification in the parameter space. The outcome of this model identifies 7 active terms marked with deeper blues. It is known that the identification of Lorenz dynamics suffers from the symmetry in the coordinate system as described in [46]. The discovered governing equation could be equivalently transformed via the affine group transformation on the coefficients and a permutation group transformation on the latent variables. Therefore, the identification is still correct in Fig. 2.3 from the following transformations. (a) We inversely transform the permutation group via assigning z_C to z_1 , z_B to z_2 , and z_A to z_3 . (b) We inversely perform affine transformation by $z_1 = 1.0$, $z_2 = -0.94z_2$, and $z_3 = 0.55z_3 - 2.81$. We demonstrate the effectiveness of the affine transformation process in Fig. 2.4 (c) that the discovered model (c.1) can be transformed into (c.2), which is close to the ground truth model (c.3). The discovered governing equation from the Laplace prior is

$$\begin{cases} \dot{z}_1 = -10.09z_1 + 10.00z_2 \\ \dot{z}_2 = 27.09z_1 - 0.86z_2 - 5.35z_1z_3 \\ \dot{z}_3 = 5.35z_1z_2 - 2.71z_3, \end{cases} \quad (2.20)$$

Stepping further from the uncertainty quantification of coefficients, we can see the uncertainty quantification in the prediction space in Fig. 2.4 (a) (b). In general, the uncertainty coverage correctly captures the true trajectory when the model fails to predict correctly and mildly covers the trajectory when the model prediction is confident.

Reaction Diffusion via Spike-and-slab prior

Reaction-diffusion is governed by a partial differential equation (PDE) that has complex interactions between spatial and temporal dynamics. We define a lambda-omega reaction-diffusion system by

$$\begin{aligned} u_t &= (1 - (u^2 + v^2))u + \beta(u^2 + v^2)v + d_1(u_{xx} + u_{yy}) \\ v_t &= -\beta(u^2 + v^2)u + (1 - (u^2 + v^2))v + d_2(v_{xx} + v_{yy}), \end{aligned} \quad (2.21)$$

which $d_1 = d_2 = 0.1, \beta = 1$.

For the synthetic data, we first generate the latent dimensions $u(x, y, t)$ and $v(x, y, t)$ by 2.21 from $t = [0, 500]$ with time step $\Delta t = 0.05$. Then, we generate the snapshots of the dynamical system spatially in xy -domain from its latent dimensions into a video with shape (10000, 100, 100). We obtain a training dataset with size 9,000. We also generate a testing dataset with 1,000 samples.

We setup the autoencoder with latent dimension $d = 2$, targeting the u, v , and apply a first-order library of functions including $[1, u_x, v_x, u_x^2, u_x v_x, v_x^2, u_x^3, u_x^2 v_x, u_x v_x^2, v_x^3, \sin(u_x), \sin(v_x)]$. We hope to discover the two oscillating spatial modes for this nonlinear oscillatory dynamics. The coefficient of Ξ is randomly initialized from Gaussian $\mathcal{N}(0, 0.1)$. The loss coefficients are $\lambda_1 = 1.0 \times 10^{-2}, \lambda_2 = 1.0 \times 10^{-1}, \lambda_3 = 20.0$. For the encoder and decoder, we use sigmoid activation function with width [256]. For optimization, we select Adam optimizer with learning rate $1e^{-3}$, and batch size to be 1000. For the setting of SSGL prior, we set $\delta = 0.08, v_0 = 0.1, v_1 = 3.0, \omega^{(k)} = 0.02 \times (0.999)^k$.

We follow the same setting in [46]. The SSGL prior only requires 1,500 epochs for training while LASSO setting frequently needs more than 3,000 epochs for training. The best test error is 2.1×10^{-3} for decoder loss, 2.1×10^{-5} for the reconstruction of $\dot{z}$, and 1.7×10^{-4} for the reconstruction of $\dot{x}$. The fraction of unexplained variance of decoder reconstruction is 9.1×10^{-5} . For SINDy predictions, the fraction of unexplained variance of $\dot{x}$ is 0.013. The fraction of unexplained variance of $\dot{z}$ is 0.001. These results all improve from LASSO based SINDy Autoencoder [46]. The posterior samples from Bayesian SINDy are shown in Fig. 2.5.

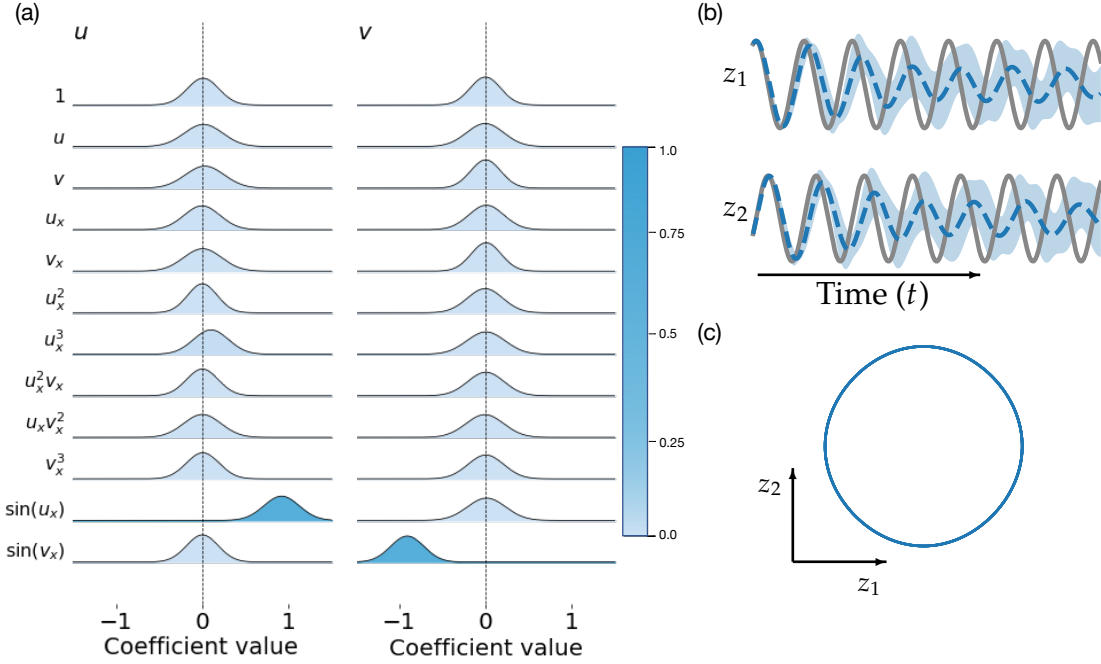


Figure 2.5: (a) Bayesian estimation and uncertainty quantification visualization of SINDy coefficient for reaction-diffusion under Spike-and-slab prior. The color represents the inclusion probability $[0.0, 1.0]$ from light blue (non-active) to dark blue (active). (b) Visualization of in-distribution (top) and out-of-distribution (bottom) predicted dynamics. (c) The generated attractor from the mean of Bayesian inference.

The discovered governing equation from the Bayesian SINDy autoencoder is

$$\begin{cases} \dot{u}_x = 0.91 \sin(u_x) \\ \dot{z}_2 = -0.91 \sin(v_x) \end{cases} \quad (2.22)$$

Nonlinear Pendulum via the Spike-and-slab prior

We consider simulated video of a nonlinear pendulum in pixel space with two spatial dimensions. The nonlinear pendulum is governed by the following second-order differential equation:

$$\ddot{z} = -\sin z. \quad (2.23)$$

We generate the synthetic dataset following the settings in [46]. The synthetic data first generates the latent dimension z as the angle of pendulum from $t = [0, 10]$ with time step $\Delta t = 0.02$. Then, we form a Gaussian ball around the mass point with angle z and length l .

This process transfers the dynamical system from its latent dimension z into a video with shape $(500, 51, 51)$. By simulating this process with 100 times, we obtain a training dataset with size 50,000. We also generate a testing dataset with 5,000 samples.

We set up the autoencoder with latent dimension $d = 1$, targeting the angle of pendulum, and apply a second-order library of functions including $[1, z, \dot{z}, z^2, z\dot{z}, \dot{z}^2, z^3, z^2\dot{z}, z\dot{z}^2, \dot{z}^3, \sin(z), \sin(\dot{z})]$. We hope to infer the real coefficient that $\Xi = [0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0]$. The coefficient of Ξ is randomly initialized from Gaussian. The loss coefficients are $\lambda_1 = 5e^{-3}$, $\lambda_2 = 5e^{-5}$, $\lambda_3 = 8e^{-4}$. For the encoder and decoder, we use the sigmoid activation function with widths $[128, 64, 32]$. For optimization, we select Adam optimizer with a learning rate $1e^{-3}$, and batch size to be 1000. For the setting of SSGL prior, we set $\delta = 0.08$, $v_0 = 0.05$, $v_1 = 3.0$, $\omega^{(k)} = 0.05 \times (0.995)^k$.

We follow the same setting in [46]. The success discovery rate is 80% from 15 training instances, which improves from 50% success discovery rate in the LASSO setting. Additionally, Bayesian SINDy Autoencoder with the SSGL prior requires only 1,500 epochs for training, while the LASSO setting frequently needs 5,000 epochs for training. The best test error is 6.5×10^{-8} for decoder loss, 1.1×10^{-3} for the reconstruction of $\ddot{x}$, and 8.1×10^{-4} for the reconstruction of $\ddot{z}$. The best fraction of unexplained variance of decoder reconstruction is 4.5×10^{-4} . For SINDy predictions, the best fraction of unexplained variance of $\ddot{x}$ and $\ddot{z}$ reconstruction are 2.3×10^{-4} and 5.5×10^{-3} . The posterior samples from Bayesian SINDy are shown in Fig. 2.6 (a). The discovered governing equation from the Bayesian SINDy autoencoder is

$$\ddot{z} = -0.99 \sin(z). \quad (2.24)$$

The prediction of trajectory with uncertainty quantification is shown in Fig. 2.6 (b) with 6 different initial conditions that $\theta_0 = [\frac{2}{3}\pi, \frac{1}{2}\pi, \frac{1}{3}\pi, \frac{1}{4}\pi, \frac{1}{8}\pi, \frac{1}{16}\pi]$ from time $t = [0, 100]$. The error and uncertainty grow with longer time interval, generally starting from $t = 20$.

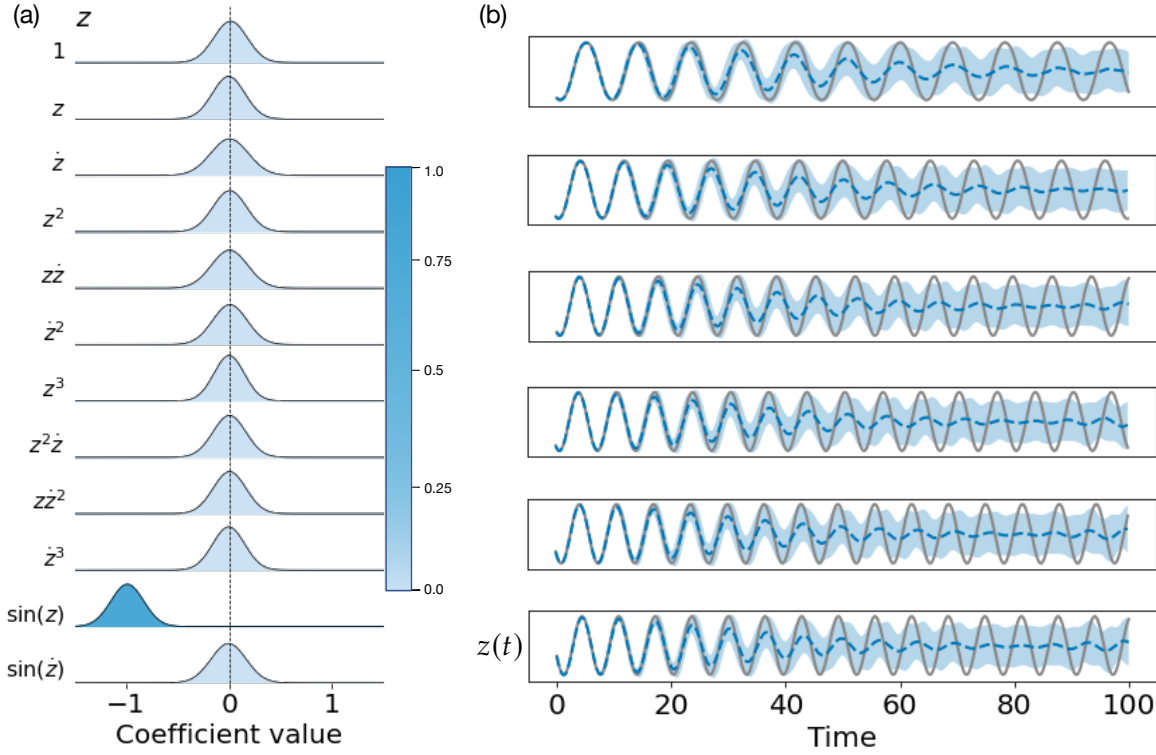


Figure 2.6: (a) Bayesian estimation and uncertainty quantification visualization of SINDy coefficient for pendulum data under Spike-and-slab prior. The color represents the inclusion probability $[0.0, 1.0]$ from light blue (non-active) to blue (active). (b) Visualization of predicted dynamics with uncertainty quantification from 6 different initializations.

2.4.2 Learning dynamical system from GoPro recording on a moving rod

Experimental setting and dataset description. The raw video data has 14 seconds recording of a moving rod with 390 frames in total. Due to the high computational requirement to access the temporal derivative in the latent space, we process the raw video via the following steps.

- (1) We first move the RGB channels of a video to greyscale, normalizing to $[0, 1]$. This is step (a) to (b) in Fig. 2.7.
- (2) We estimate the background in this video. In our setting, we compute an averaged frame over the entire video as the background. In a wider context, one shall remove low-rank information via linear models [169, 296]. This is step (b) to (c) in Fig. 2.7.
- (3) Apply Gaussian filters with appropriate variance. The selection of variance varies from

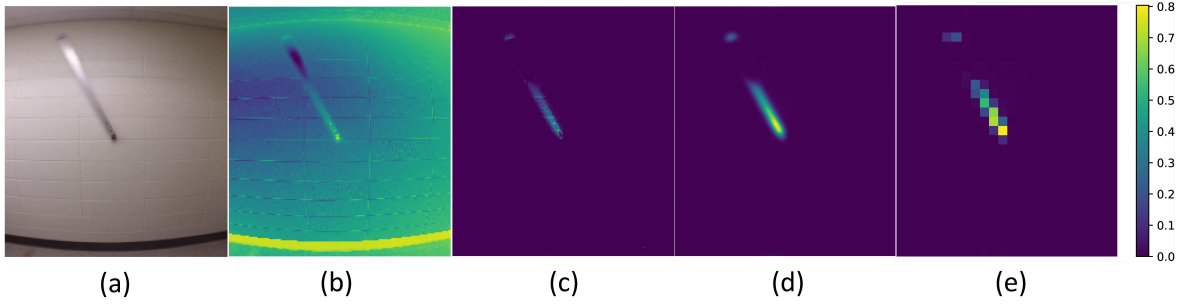


Figure 2.7: Preprocessing pipeline to high-dimensional video data to remove auxiliary information and reduce the size of video frames.

case to case. A good hyperparameter setting should keep the mix-max gap of the original image and could also smoothen the sharp edges of the objects. This is step (c) to (d) in Fig. 2.7.

- (4) The final step downsample the frames from $(1080, 960)$ to $(27, 24)$ via interpolation. We choose this size of subsampling to fit into our experimental environment, and balance the training speed.

After these preprocessing steps, we obtain a training dataset with 390 samples and shape $(27, 24)$. The downsampling nature of our preprocessing steps and the finite difference estimation of temporal derivatives inject large corruption noise to the video. Empirically, the preprocessed data exhibits 3% of random Gaussian noise across the entire image domain, and could experience up to 100% of random noise for areas with the moving rod, e.g. pixel missing, lighting issues, error from downsampling. To be more specific, in Fig. 2.7 (b), the illuminated part is processed as 0, implying the absence of that section of the pendulum. These random missingness from lightings, while potentially unnoticeable to the human eye, manifest large and noisy spikes in the temporal derivative approximation with the magnitude of 10^3 randomly across the pixel space. This type of corruption noise is one major challenge in this real video dataset for governing equation identification.

Bayesian masked autoencoder setup We set up the autoencoder with latent dimension $d = 1$, with a second-order library of functions similar to synthetic video data. We remove the constant term for simplicity of the inferential process. The loss coefficients are $\lambda_1 = 5e^{-7}$, $\lambda_2 =$

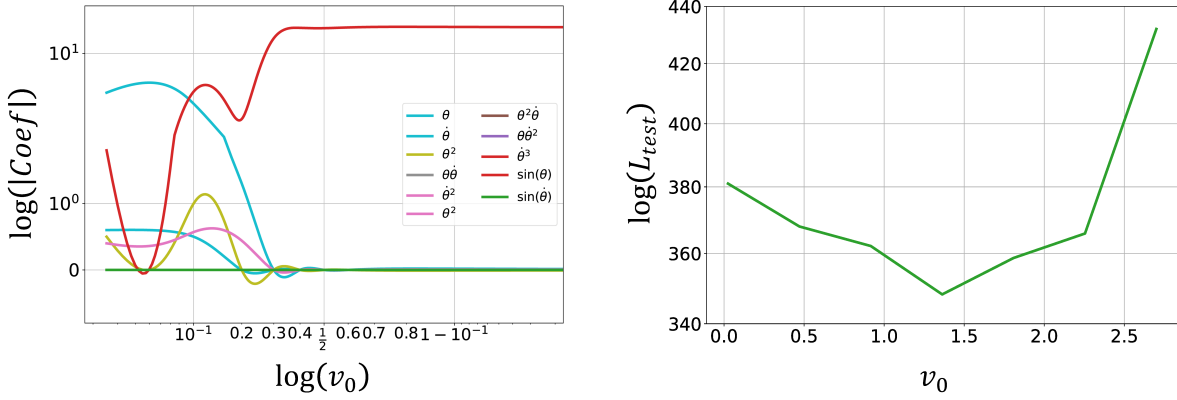


Figure 2.8: Left: SSGL regularization plot. Right: Model selection via test error.

$5e^{-8}$, and $\lambda_3 = 1 \times 10^{-4}$. For the encoder and decoder, we use the sigmoid activation function with widths $[64, 32, 16]$. For optimization, we select Adam optimizer with a learning rate $1e^{-3}$, and batch size to be 10. Real video data is much harder compared to synthetic video data since it contains rich information, including object colors, background, and processing noises. At the same time, only 390 snapshot samples are available to train the Bayesian autoencoder. To resolve this problem, we involve masked autoencoder [119], which applies 50% random masking to the input data. The effect of masked autoencoder could be understood as data augmentation, which allows the encoder to focus more on the key information [303]. For the setting of SSGL prior, we set $\delta = 0.09$, $v_0 = 1.5$, $v_1 = 5.0$, $\omega^{(k)} = 0.01 \times (0.999)^k$.

Results from the Bayesian discovery via SSGL prior

Model selection and the regularization path Following the standard practice [227], we study the regularization path of the SSGL prior by increasing the spike variance. Fig. 2.8 (left) shows gradual sparsification of the discovered governing equation. The discovered model is null when spike variance $v_0 \geq 2.6$, which is not shown in the regularization plot. We combine this regularization plot with the testing loss plot in Fig. 2.8 (right). We select the sparsest model with the lowest testing error from the Occam’s razor principle. The full regularization path study indicates that there exists governing physics law (with the term $\sin(z)$) in this video recording.

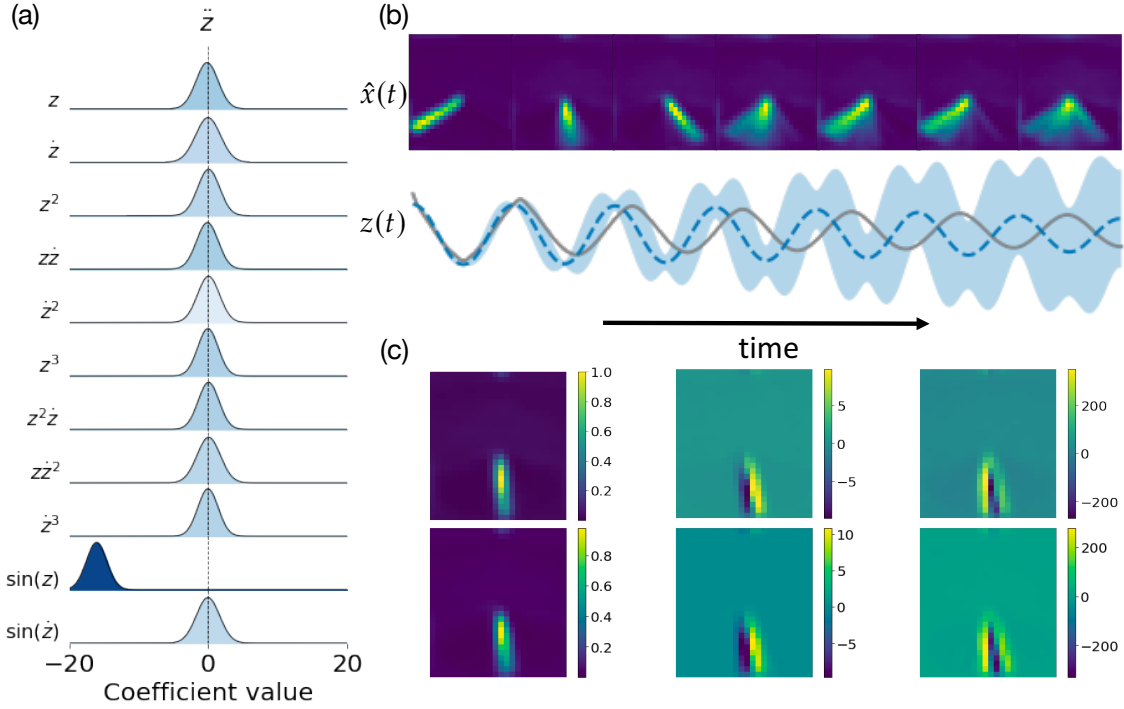


Figure 2.9: (a) Bayesian estimation and uncertainty quantification visualization of SINDy coefficient for real moving rod data under Spike-and-slab prior. The color represents the inclusion probability [0.0, 1.0] from light blue (non-active) to dark (active). (b) Visualization of prediction in both latent dynamics and pixel space with uncertainty quantification. (c) Reconstruction of Bayesian SINDy autoencoder for video frame, and temporal derivatives.

Discovered governing equation The success discovery rate of Bayesian SINDy Autoencoder is 100% from 10 training instances, and it only requires 1,500 epochs for training. The best test error is 0.767 for decoder loss, 0.305 for the reconstruction of $\ddot{x}$, and 0.0263 for the reconstruction of $\ddot{z}$. The fraction of the unexplained variance of decoder reconstruction is 0.003. For SINDy predictions, the fraction of unexplained variance of $\ddot{x}$ and $\ddot{z}$ reconstruction are 0.081 and 0.961.

The posterior samples from Bayesian SINDy are shown in Fig. 2.9 (a), and the prediction of trajectory with uncertainty quantification of latent dynamics is shown in the bottom figure of Fig. 2.9 (b). The discovered governing equation from the SSGL prior is

$$\ddot{z} = -16.06 \sin(z). \quad (2.25)$$

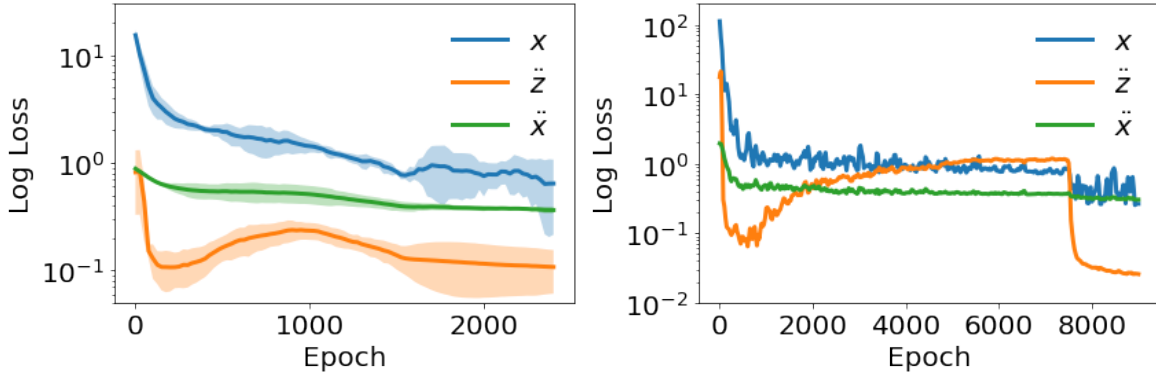


Figure 2.10: Left: by running the experiment 5 times with 2000 training epochs and 1500 refinement epochs, we plot the mean and standard deviation of testing loss under the log scale. Right: by running the experiment once with 7500 training epochs and 1500 refinement epochs, we plot the testing loss under the log scale.

Video reconstruction, prediction, and uncertainty quantification The latent dimension $\ddot{z}$ with SINDy correctly identifies $\sin(z)$ as the active term and generates an uncertainty estimate. Utilizing the uncertainty estimation in the parameter space, we could perform uncertainty quantification on the prediction space as in Fig. 2.9 (b). The training outcome of Bayesian SINDy autoencoder are shown in Fig. 2.9, 2.10. First, from Fig. 2.9 (c), we note that the reconstruction of the Bayesian SINDy autoencoder is very promising. The top line of figures is input data, and the second line of figures is generated by the autoencoder. Even if the scale is slightly different, it is convincing that the overall reconstructions on x , $\dot{x}$ and $\ddot{x}$ are very good. This can be suggested by Fig. 2.10 that the log loss converges with more training epochs. Throughout these experiments, we uniformly set training epochs to be 1,500 and refinement epochs to be 1,500. This decision is suggested by the observation in a longer run (7,500 training epochs and 1,500 refinement epochs) that the log loss remains stable after 1000 epochs. For the reconstruction of $\ddot{z}$, having longer training epochs will harm the testing error due to overfitting. We note here the discovery by SSGL prior has dominant performance in testing loss comparing to SINDy autoencoder as shown in 2.5. This suggests the superiority of the SSGL prior in the real video data setting.

As suggested in Sec. 3.4, the Bayesian SINDy autoencoder enables a trustworthy solution in deep learning for video prediction. In Fig. 2.9 (b), we predict the following frames of video using the decoder and the simulation of latent dynamical systems. We observe that in the

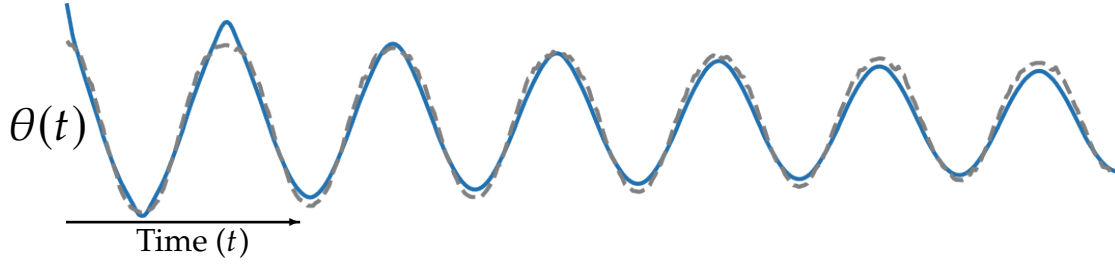


Figure 2.11: Latent dimension visualization after rescaling (blue curve) versus manually labelled moving rod angle by human (grey curve).

very beginning of future time frames, the prediction is very confident and accurate. Yet as we move to a longer time interval, the prediction becomes very uncertain, and we can observe this interesting phenomenon in Fig. 2.9 (b).

It is essential to understand the latent dimension of the autoencoder in order to know the validity of the coordinate system discovery. Therefore, we create the following analysis on the latent dimension. We first manually label all angles θ from all video frames by humans, as shown in the grey dashed curve in Fig. 2.11. The blue curve represents the rescaled latent dimension z , and we observe these two curves match mostly perfect to each other. We apply rescaling in this process due to the coordinate system for the angles θ are equivalent after arbitrary transformation in its scale. For example, the angles represented by radians and degrees are equivalent to each other under different scales.

On the discovery of the standard gravity constant g The SSGL prior accurately discovers the model with $\sin(z)$ term and reveals the standard gravity constant g . In our experimental setting, the length of the rod in our experiment is 123 cm with the initial condition at 75° . We have an estimate of g given these conditions that $\hat{g} = -9.876$. The estimation $\hat{g}$ is slightly overconfident and biased which concentrates at $(-9.889, -9.865)$. The bias is more likely to be removed with an enlarged video datasets.

Given this low-data and high-noise setting, the estimate from SSGL prior $\hat{g}$ is very close to the true value that $g \approx -9.807$. Even if the discovery of the coordinate system could be up to an arbitrary scaling, the frequency of the dynamical system could uniquely identify the coefficient of $\sin(z)$. This validates the discovery under random scaling group transformation

in the latent dimension.

2.5 Conclusion and Future works

In this paper, we propose a Bayesian SINDy autoencoder for automated coordinate and governing equation discovery from high-dimensional data. Using the Bayesian learning framework and sparsity-promoting priors, the proposed model identifies sparse dynamics in the latent dimension. Through experimental studies on synthetic Lorenz, reaction-diffusion system, pendulum, and real video of moving pendulum. We successfully perform model discovery under a learned coordinate system. In the small-data and high-noise regime for real video data, we identify the correct physical law with close estimation of the standard gravity constant g . Besides the model and physics discovery, in terms of video prediction, the Bayesian SINDy autoencoder provides uncertainty-aware future predictions with an exact understanding of its underlying physics, which enables a trustworthy alternative for deep learning-based video prediction.

In our perspective, the Bayesian SINDy autoencoder holds significant potential in advancing the concept of ‘GoPro physics’ [295]. However, the current framework has several limitations that restrict broader application across various problems. To enhance the applicability and performance of the Bayesian autoencoder, we could consider the following strategies as potential future directions.

- **Autotuning for $\lambda_1, \lambda_2, \lambda_3$.** It is essential to balance the reconstruction loss function weights in x , $\dot{x}$ and $\dot{z}$. To mitigate the risk of converging to suboptimal solutions, one can directly solve the multi-objective optimization for encoder-decoder architecture using methods like Multiple Gradient Descent Algorithm [243]. This could also practically reduce the manual effort required for hyperparameter tuning.
- **Full Bayesian specification.** The hyperparameter setting for the sparsity level (δ), spike-and-slab distribution variances (v_0, v_1), and environmental noise (σ) can be specified using a full Bayesian inference setup as described in [73, 227]. This approach lessens the need of hyperparameter tuning in the Bayesian sparse regression procedure. However, this will

exceedingly enlarge the computational requirement and could be challenging to implement.

- **Ensembling and variational autoencoders.** One may consider to utilize an ensemble of trained Bayesian SINDy autoencoder initialized from various starting points [294, 297]. This process constructs a posterior distribution of the SINDy coefficient Ξ , and could perform subset selection via stability selection or inclusion probability thresholding [90, 98, 193]. Applying variational autoencoders [151] could avoid the usage of hard-thresholding procedures. It also helps to perform better subset selection with Bayesian uncertainty estimation to the SINDy coefficient [151].

In the future work, it is important to involve new techniques to discover governing physics with multiple objects and complex chaotic systems. Bayesian SINDy autoencoder also presumes a static coordinate system, making it challenging to analyze videos captured with a moving camera. Further refinements on the autoencoder architecture are likely to resolve these challenges.

In conclusion, the Bayesian SINDy autoencoder offers significant improvements over the SINDy autoencoder, especially in scenarios with limited data and high noise. The Bayesian learning framework not only accelerates the training but also delivers superior performance across various cases. Additionally, the Bayesian SINDy autoencoder enables uncertainty estimation, which is essential for making informed decisions based on data. Our experiments demonstrate the effectiveness of the Bayesian SINDy autoencoder using synthetic and real video data. Remarkably, even with a 14 seconds video of a real moving pendulum, the Bayesian SINDy autoencoder successfully discovers the coordinate system and governing equation and estimates the fundamental constant g . In future works, we hope to extend the current methodology into more complex scenarios with real videos to explore further the power of Bayesian SINDy autoencoder in diverse applications.

Appendix: Real video discovery via the Laplace prior

We present the result from the Laplace prior. In this case, the Laplace prior identifies an incorrect term z over than the true target on $\sin(z)$ if we specify a reasonable sparsifying

constant. However, this model is worse than the null model with respect to the testing error. Therefore, Bayesian autoencoder with the Laplace prior fails to discover governing physics from real video data.

For the setting of Laplace prior, we set $\lambda_3 = 1 \times 10^{-2}$. The success discovery rate is 70% from 10 training instances while the other 30% instances will lead to zero discovery, indicating no active indices. Bayesian SINDy Autoencoder requires 1,500 epochs for training. The best test error is 0.779 for decoder loss, 0.364 for the reconstruction of $\ddot{x}$, and 0.416 for the reconstruction of $\ddot{z}$. The fraction of unexplained variance of decoder reconstruction is 0.012. For SINDy predictions, the fraction of unexplained variance of $\ddot{x}$ and $\ddot{z}$ reconstruction are 0.210 and 0.941.

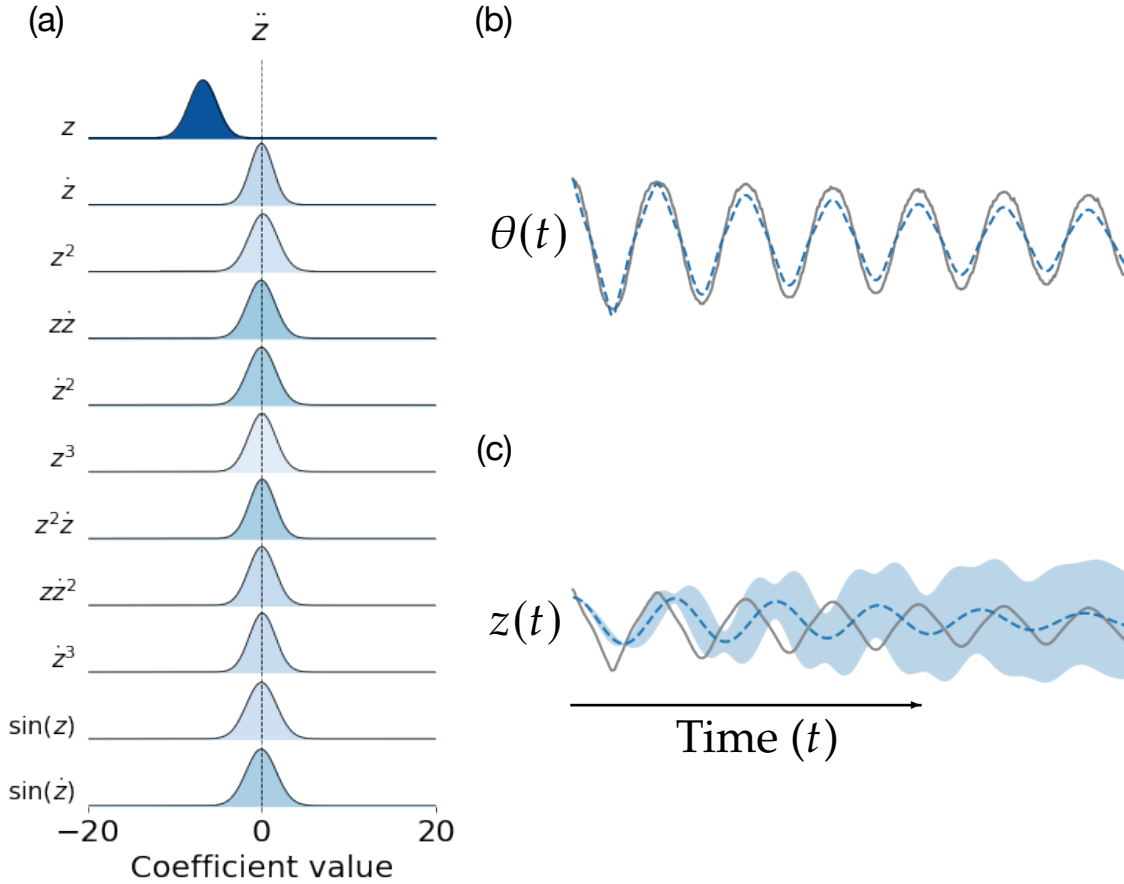


Figure 2.12: (a) Bayesian estimation and uncertainty quantification visualization of SINDy coefficient for real moving rod data under Laplace prior. The color represents the inclusion probability [0.0, 1.0] from light blue (non-active) to dark blue (active). (b) Latent dimension visualization after rescaling (blue curve) versus manually labelled moving rod angle by human (grey curve). (c) Visualization of predicted dynamics in latent space with uncertainty quantification.

The posterior samples from Bayesian SINDy are shown in Fig. 2.12 (a) and the prediction of trajectory with uncertainty quantification of latent dynamics is shown in Fig. 2.12 (c). The discovered governing equation from the Laplace prior is

$$\dot{z} = -7.10z. \quad (2.26)$$

The discovered governing equation with z slightly deviates from the true dynamics similarly as reported in [46], but still creates reasonable uncertainty estimate under model misspecification. The latent dimension in Fig. 2.12 (b) follows a similar process in Fig. 2.11. The latent dimension z still matches the human labels, but not as close as the learned result by the SSGL prior.

Regularization path and model selection Fig. 2.13 (left) shows gradual sparsification of the discovered governing equation for increasing threshold. The discovered model is null when spike variance $v_0 \geq 4.0$, which is not shown in the regularization plot. Unlike the SSGL regularization plot, the Laplace prior fails to discover $\sin(\theta)$ term uniformly through the entire regularization path. Additionally, combining with the testing loss plot in Fig. 2.13 (right), the selected model with lowest testing error is a null model. The null model (zero model) only rebuilds the background with the entire trajectory in one graph. The full regularization path study indicates that there is no active physical law in this video recording.

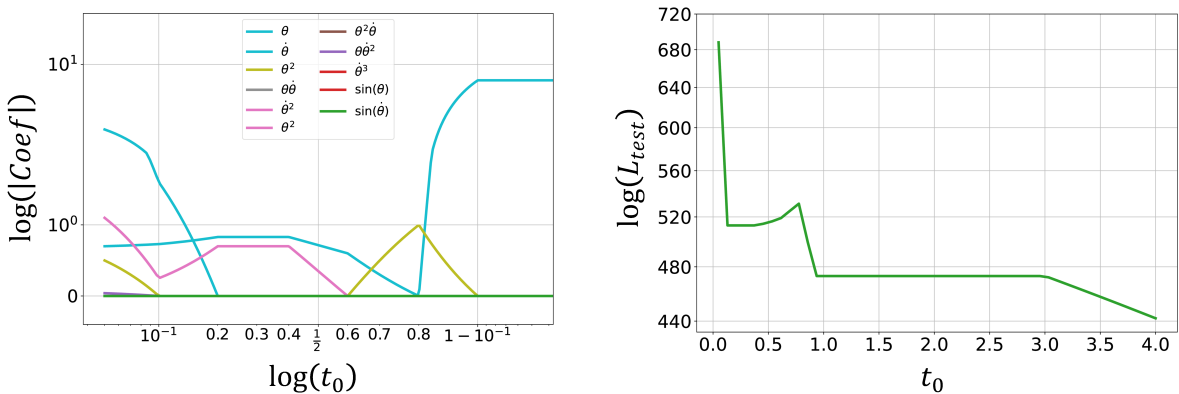


Figure 2.13: Left: Laplace prior (LASSO) regularization plot. Right: Model selection via test error.

Appendix: Pre-processing

Necessity of data preprocessing for video data learning. Video data are generally harder comparing to prior works on simulated data. The challenges come from the following three parts.

- We only have snapshots of dynamical system in the pixel space for real video data. There is no direct access to temporal derivatives in the pixel space. The temporal derivatives are required in the computation of SINDy loss, and we have to approximate it via finite difference method from video snapshots.
- Videos contain more auxiliary (unnecessary) information for dynamical system learning including background, colors, light changes, and processing noises. Deep autoencoders could remove random Gaussian noises, but are not naturally designed to deal with this type of data.
- For modern cameras, videos normally have very high resolution (e.g. $1980 \times 1080 \times 3$). It is computationally demanding to perform this level huge matrix operations w.r.t this size for neural network learning, e.g. $\nabla_{\mathbf{x}} f(\mathbf{x})$.

Suggestion for automating the video recording preprocessing. In Algorithm 1, we demonstrate a standard preprocessing procedure for governing equation learning. We first remove background via deleting the averaged frame. Then, we apply Gaussian filters to smooth the video frames. Finally, we subsample into $\tilde{X}$ and compute temporal derivatives.

Algorithm 1 Real video data preprocessing

Input: Video data frame $\{X_i\}$, smoothing variance c , subsample dimension d_1, d_2 , time between frame Δt .

Output: smoothed video frames $\{\tilde{X}_i\}$

```

1: function PREPROCESS( $X, c, d_1, d_2$ )
2:    $\bar{X} = \frac{1}{t} \sum_1^t X_i;$  ▷ compute average frame
3:    $\hat{X} = \{X_i - \bar{X}\}_{i=1}^t;$ 
4:   for  $i$  in  $1, 2, \dots, t$ : do
5:      $\hat{X}_i^{smooth} = \text{GaussianSmooth}(\hat{X}_i, c);$ 
6:   end for
7:    $\tilde{X} = \text{Subsample}(\hat{X}^{smooth}, d_1, d_2);$  ▷ subsample the image to size  $(d_1, d_2)$ 
8:   for  $i$  in  $1, 2, \dots, t - 1$ : do
9:      $\tilde{X}_i = \frac{\tilde{X}_{i+1} - \tilde{X}_i}{\Delta t};$ 
10:  end for
11:  for  $i$  in  $1, 2, \dots, t - 2$ : do
12:     $\tilde{\tilde{X}}_i = \frac{\tilde{X}_{i+1} - \tilde{X}_i}{\Delta t};$ 
13:  end for
14:  return  $\tilde{X}, \tilde{\tilde{X}}, \tilde{\tilde{X}}$ 
15: end function

```

3

Sparse Identification of Nonlinear Dynamics and Koopman Operators with Shallow Recurrent Decoder Networks

Partial differential equations (PDEs) derived from first principles or qualitative behavior remain the most ubiquitous class of models used to describe physical, spatio-temporal phenomena. However, for complex dynamics it is often the case that the simplifying assumptions necessary to construct a PDE model can render it ineffectual for real data where the physics is high-dimensional, multi-scale in nature, only partially known, or where first principles models currently do not exist. In such cases, machine learning (ML) methods offer an attractive alternative for learning both the physics and coordinates (fundamental variables) necessary to quantify the observed spatiotemporal phenomena. 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 [192]. The *Shallow REcurrent Decoder* (SHRED) network [292] is a recently introduced architecture that utilizes 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 sensing and reduced order modeling examples ranging from weather and

atmospheric forecasting [292], plasma physics [161], nuclear reactors [226], and turbulent flow reconstructions [270]. Theoretically rooted in the classic PDE method of separation of variables (see detailed derivations in [292, 270]), the decoding-only strategy of SHRED circumvents the computation of inverse pairs, i.e. an encoder and the corresponding decoder. It has been well-known for decades that the computation of the inverse of a matrix is highly unstable and ill-posed [93, 124, 10]. By decoding only, SHRED avoids this problem and learns a single embedding without the corresponding inversion.

In this paper, we introduce **S**parse **I**dentification of **N**onlinear **D**ynamics with **S**Hallow **R**ecurrent **D**ecoder networks (SINDy-SHRED). SINDy is a computational method that applies sparse regression techniques to perform nonlinear system identification. The resulting model is a set of ordinary differential equations (ODEs), well-suited for interpretable modeling. SINDy-SHRED exploits the latent space of recurrent neural networks for sparse sensor modeling, and enforces interpretability via a SINDy-class functional regularization, where the SINDy-class denotes functions formed by sparse linear combinations of candidate basis functions (e.g., polynomial or trigonometric) from the chosen SINDy library. Moreover, our theoretical analysis (see Thm. 3 and Thm. 4) rigorously demonstrates key advantages of latent space modeling with SINDy over conventional neural networks, i.e. under mild conditions it is guaranteed to converge with a bounded error. In this way, SINDy-SHRED enables a robust and sample-efficient joint discovery of governing equation and coordinate system. With the correct governing equation, SINDy-SHRED can perform an accurate long-term prediction in a learned, low-dimensional latent space, and in turn allows for long-term forecasting in the original spatiotemporal (pixel) space. In further restricting SINDy to a linear model, a Koopman approximation [31] can be constructed to produce a Koopman-SHRED architecture.

SINDy-SHRED and Koopman-SHRED are lightweight models which can perform low-rank recovery with as few as three active sensors for the high-dimensional spatio-temporal fields, which is critical for large-scale scientific data modeling and real-time control. Specifically, for D -dimensional fields, $D + 1$ sensors are required for reconstructing the spatio-temporal field from sparse observations, much like localization for cellular networks [249]. Moreover, the proposed architecture does not require large amounts of data during training, thereby

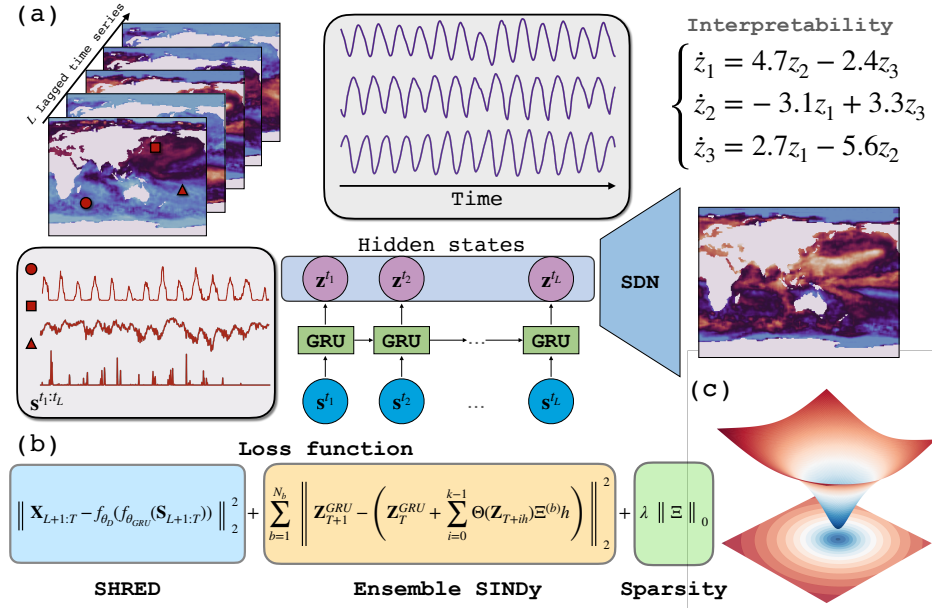


Figure 3.1: (a) Illustration of the SINDy-SHRED and Koopman-SHRED architecture. SINDy-SHRED transfers the original sparse sensor signal (red) to an interpretable latent representation (purple) that falls into the SINDy-class functional. This framework can be adapted into Koopman-SHRED by restricting the library $\Theta(\cdot)$ to be linear. The shallow decoder performs a reconstruction in the state space. We obtain an interpretable linear model for the sea-surface temperature data considered (details in Sec. 3.4.3). (b) The loss function consists of three parts: (i) the SHRED loss controls the reconstruction accuracy of the state space, (ii) the Ensemble SINDy loss helps to model the parsimonious dynamics of the latent space, and (iii) the sparsity constraint identifies the governing equation within this optimization framework. (c) We visualize the globally convex loss landscape of SINDy-SHRED as in [168].

avoiding a common pitfall in existing ML techniques for accelerating physics simulations and enabling rapid training on a single laptop. SINDy-SHRED/Koopman-SHRED are also highly reproducible with minimal effort in hyperparameter tuning due to the globally convex optimization landscape, as illustrated in the lower right panel of Fig. 3.1. The recommended network structure with the same set of hyperparameters, and training setting can generalize to many different datasets¹. In short, we demonstrate SINDy-SHRED/Koopman-SHRED to be robust and highly applicable in many modern scientific modeling problems. In what follows, we will refer generally to the proposed architecture as SINDy-SHRED, with Koopman-SHRED as a special case.

We perform a wide range of studies to demonstrate the effectiveness of SINDy-SHRED. We apply the model on the sea surface temperature data, which is a complex real-world

¹Code implementation: <https://github.com/gaoliyao/sindy-shred>.

problem. We also consider data from complex simulations of atmospheric chemistry, 2D Kolmogorov flow, and isotropic turbulence, 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,” whereby physics is learned directly from video recordings. With extremely small sample size and noisy environments, SINDy-SHRED achieves governing equation identification with stable long-term predictions, thus overcoming the critical limitations for long-term forecasts of spatio-temporal phenomena that include instabilities and massive computational requirements. To summarize, the contributions of our paper are the following:

- We propose SINDy-SHRED to learn a symbolic and interpretable generative model of a parsimonious and low-dimensional latent space (fundamental variables) of recurrent neural networks for complex spatio-temporal dynamics.
- Even for the well-known physical systems considered, we discover in each case a hitherto unknown physics model characterizing the spatio-temporal dynamics directly from measurement or video streams. These benchmark systems include turbulent flows, videos of physical systems, and PDE simulations.
- Under weak assumptions, we establish rigorous error bounds on the robust convergence of the SINDy-SHRED algorithm which allows us to guarantee model performance, a rarity in deep learning. Moreover, we systematically observe that the algorithm has a global convex loss landscape, preventing the need for any hyper-parameter tuning (except for the SINDy parameters of dimension and sparsity).
- We compare SINDy-SHRED to state-of-the-art deep learning algorithms in spatio-temporal prediction, showing that the architecture achieves superior accuracy, data efficiency, training time, and long term prediction, all with fewer model parameters.

3.2 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 knowledge of the underlying PDE, physics-informed neural networks [221] 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 [150, 173, 132, 185, 174, 182]. Alternatively, variants of SINDy [234, 194, 90] offer a data-driven approach to identify PDEs from the spatial-temporal domain. High-dimensionality and data requirements can be prohibitive for many applications.

In parallel, efforts have been directed toward the discovery of physical laws through dimensionality reduction techniques [46, 186, 191], providing yet another perspective on the modeling of scientific data. The Mori-Zwanzig formalism [62, 175, 176, 267] also provides a theoretical foundation for reduced-order modeling of spatiotemporal systems. The discovery of physics from a learned latent spaces has previously been explored by [96, 60, 88, 64, 300, 171, 211, 209, 198, 54, 183], yet none of these methods consider a regularization on the latent space with no explicit encoder. Yu et al. [311] proposed the idea of physics-guided learning 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 [91, 269, 237], computer vision [302, 298, 6, 14] and computer graphics [142, 178, 299, 200], as these fields all require better physics models.

From the deep learning side, combining the structure of differential equations into neural networks [120, 55] has been investigated in many tasks. As an example, when spatial-temporal modeling is framed as a video prediction problem, He et al. found [119] 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 [246, 287, 105, 115]. Generative models have also been found to be useful for scientific data modeling [195, 252, 36, 250, 82, 23, 164, 214, 221] and data assimilation methods similarly aim to combine imperfect forecast models with sparse sensor measurements to improve running state estimates of high-dimensional systems [58, 23, 164, 214]. Unlike many of the previous works above, SINDy-SHRED (i) provides a computationally efficient white-box approach for scientific data modeling; and (ii) discovers the governing

equations directly from data without assuming prior physics. These advantages are important for real-world scientific applications.

3.2.1 Sparse identification of nonlinear dynamics (SINDy)

SINDy is a computational method that applies sparse regression techniques to perform nonlinear system identification. Given n snapshot data of a dynamical system $\mathbf{z}(t) = [z_1(t), z_2(t), \dots, z_d(t)]^\top \in \mathbb{R}^d$, SINDy identifies the symbolic form of the following dynamical system

$$\dot{\mathbf{z}}(t) = \mathbf{f}(\mathbf{z}(t)) = \Theta(\mathbf{z}(t))\Xi, \quad (3.1)$$

where $\Theta(\mathbf{z})$ is a library of candidate nonlinear functions. The sparse coefficient matrix Ξ determines which terms in the candidate library are active, which encodes both the structure and coefficient values of the dynamical system. The candidate function library $\Theta(\mathbf{z})$ typically includes commonly used functions for ODE modeling such as polynomial, and trigonometric terms $\Theta(\mathbf{z}) = [1, \mathbf{z}, \mathbf{z}^2, \dots, \sin(\mathbf{z}), \cos(\mathbf{z})]$.

Since we do not have direct access to $\frac{d\mathbf{z}}{dt}$ in real life, $\dot{\mathbf{z}}$ is practically estimated via numerical differentiation algorithms. By solving the following sparse regression problem, the sparse coefficient Ξ identifies the governing equation from data $\dot{\mathbf{z}}(t) = \Theta(\mathbf{z})\Xi + \epsilon$ with noise ϵ . The recovery of Ξ is theoretically guaranteed with sufficient sample sizes [98].

Despite the success of SINDy in scientific discovery in a wide range of applications [180, 254, 18, 312, 131, 67, 144], it does not scale well with high-dimensional problems. Many high-dimensional systems have a compact representation in a lower-dimensional subspace. This subspace is accessible via a coordinate transformation which contains a compact representation that follows a governing law. Take a video of a moving rod as an example, the physics of the high-dimensional pixel space is governed by a single variable θ which corresponds to the angular displacement. The latent variable θ will follow the classical ODE for pendulum moving $\ddot{\theta} = -\kappa \sin(\theta)$ where κ is a constant, but the values of the pixels in the original high-dimensional state space will not follow any simple forms of an ODE. To perform

a joint discovery of the coordinate system and governing equations, prior works have been focusing on adapting autoencoders with SINDy either within training [46, 191] or after [96].

3.3 Methods

The shallow recurrent decoder network (**SHRED**) is a computational technique that utilizes recurrent neural networks to estimate high-dimensional states from limited measurements [292]. 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 $\{\mathbf{X}_i\}_{i=1}^T$ where $\mathbf{X}_i \in \mathbb{R}^{N_x \times N_y}$ that represents the evolution of a spatio-temporal dynamical system, where N_x , N_y , and T denote number of discretized spatial grid points along two spatial dimensions, and total time steps of the system, respectively. The total time steps are $\mathbf{t} = \{t_i\}_{i=1}^T$. In SHRED, N_s sensors record measurements of the evolving physical system $\{\mathbf{X}_i\}_{i=1}^T$ at discrete time steps $\mathbf{t}$. Denoting the time-series data of SHRED as $\{\mathbf{s}_i\}_{i=1}^T$, where each $\mathbf{s}_i \in \mathbb{R}^{N_s}$ contains measurements of all sensors at time t_i . To incorporate the temporal information from the sensors, SHRED employs an L -lagged temporal sensor history, such that the input-output pair is defined as $\mathcal{D} = \{(\mathbf{s}_{t_{i-L}:t_i}, \mathbf{X}_{t_i})\}_{t_i=L+1}^T$, where $t_{i-L} : t_i$ is a compact notation for discrete index set $\{t_{i-L}, t_{i-L+1}, \dots, t_i\}$.

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 sparse sensor history. In vanilla SHRED, a recurrent network 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 generalizes the separation of variables method for solving PDEs that the GRU $f_{\theta_{\text{GRU}}}$ captures temporal evolution from sparse sensor histories, and the decoder $f_{\theta_{\text{D}}}$ reconstructs the spatial field. For a detailed discussion of this connection, please see [292, 270].

SHRED enables efficient sparse sensing that is widely applicable to many scientific problems [78, 161, 232, 226, 270]. 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. Furthermore, SHRED is grounded in the classical technique of separation of variables [270].

3.3.1 Learning representations for physics discovery using SHRED

To achieve a parsimonious representation of physics, it is important to find a coordinate system that effectively captures the underlying dynamics and structure of the system. In SINDy-SHRED (shown in Fig. 3.1), we extend the advantages of SHRED, and perform a joint discovery of coordinate systems and governing equations. This is accomplished by enforcing the latent space to follow an ODE in the SINDy class of functions.

Finding better representations SHRED has a natural advantage in modeling 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 [220]. 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) [222, 46, 191]. Building upon SHRED, we further incorporate SINDy to regularize the learned recurrence with a well-characterized and simple form of governing equation. This approach is inspired by the principle that, under an ideal coordinate system, physical phenomena can be described by a parsimonious dynamics model [46, 191]. 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 from the learned dynamics.

Physics discovery and prediction By regularizing the latent dynamics to follow an ODE model, SINDy-SHRED automatically performs physics discovery straight from high-dimensional data. The resulting SINDy model captures the real physical law that governs the observed dynamics. This enables a physics discovery in the intrinsic latent coordinate system induced from the high-dimensional observation. Notably, even though the model is trained only on sensor trajectories up to a finite time horizon, the physics discovered with SINDy regularization endows it with strong extrapolative capability beyond the training window. With the discovered physics, SINDy-SHRED is able to predict future spatiotemporal evolution without requiring any further sensor inputs. After initialization with a short stream of sensor measurements, the discovered SINDy model can unroll the latent dynamics forward in time, and the decoder will map these latent trajectories back to the high-dimensional state space. Across all computational experiments, we found the discovered models robustly produce accurate long-term rollouts, which validates the discovered physics.

3.3.2 Latent space regularization via SINDy and Koopman

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 consists of a library of commonly used terms, including polynomials and Fourier series. The expressiveness of these terms allow the model to capture complex dynamics.

SINDy as a Recurrent Neural network

Incorporating SINDy within the neural network training is highly nontrivial. All prior works [46, 191, 9] require to compute temporal derivatives on the latent space $\hat{z}$ during training. However, estimating time derivatives within neural network training is inherently unstable and computationally expensive. For example in [46, 191], the derivative $\dot{z}$ is computed via the explicit solution (c.f. [46] S1.3). This operation fundamentally requires the neural network weights and the time derivative of the original state space $\dot{x}$. As a result, this estimation is highly sensitive to network fluctuations during training, causing loss spikes and training instability, while also being unstable under noise and computationally expensive when evaluating $\dot{x}$ for large-scale scientific datasets.

To resolve the problems above, we reformulate SINDy in its variational form [65] without computing the time derivatives. Inspired by ResNet [120] and Neural ODE [55], we utilize skip connections to model residual and temporal derivatives. It allows us to rewrite SINDy into a Recurrent Neural Network (RNN) which has the following form $z_{t+1} = z_t + f(z_t)$ where $f(\cdot)$ is some function of the input, and z_t denotes $z(t)$. From the Euler method, the ODE forward simulation via SINDy effectively falls into the category of Recurrent Neural Networks (RNNs) which has the form

$$z_{t+1} = z_t + f_{\Theta}(z_t, \Xi, \Delta t), \quad (3.2)$$

where $f_{\Theta}(z_t, \Xi, \Delta t) = \Theta(z_t)\Xi\Delta t$ is a nonlinear function. Notice that this $f_{\Theta}(\cdot)$ follows the SINDy formulation of dynamical systems [32] rewritten in a variational form [65]. We demonstrate a visual form of this design in the SI Appendix. The application of function libraries with sparsity constraints is a manner of automatic neural architecture search (NAS) [325]. Compared to all prior works [46, 96, 64], the usage of the variational form of SINDy is similar to a RNN training, which fits better as a neural network model with gradient-based training. Additionally, applying the variational form in the latent space, we avoid the need to estimate vector fields in the original high-dimensional state space, which are both computationally expensive and numerically unstable, and were required

in prior works [46, 191]. We utilize trajectory data $\mathbf{Z}_T = \{\mathbf{z}_i\}_{i=1}^{T-1}$ and forward simulate the SINDy-based ODE using a trainable parameter Ξ . 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 Ξ with the following:

$$\Xi = \arg \min \left\| \mathbf{Z}_{T+1} - \left(\mathbf{Z}_T + \sum_{i=0}^{k-1} \Theta(\mathbf{Z}_{T+ih}) \Xi h \right) \right\|_2^2, \min \|\Xi\|_0,$$

where $\mathbf{Z}_{T+ih} = \mathbf{Z}_{T+(i-1)h} + \Theta(\mathbf{Z}_{T+(i-1)h}) \Xi h$.

To achieve ℓ_0 optimization, we perform pruning with ℓ_2 regularization which is known to approximate ℓ_0 regularization under regularity conditions [321, 98, 26]. 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 and propagates only a single hidden state. Further details are provided in the SI Appendix.

Our goal is to find a SHRED model with a latent state that is within the SINDy-class functional. However, there is no guarantee that the initial latent representation from SHRED will follow any ODE structure. 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 the spatial domain reconstruction. Therefore, it is important to let the two latent spaces align smoothly and progressively.

In SI Appendix Algorithm 1, we describe our training procedure that allows the two

trajectories to progressively align with each other. To further ensure a smooth alignment and avoid over-regularization, we introduce ensemble SINDy units with varying levels of sparsity constraints, ranging from a full model (all terms in the library are active) to a null model (where no dynamics are represented). From the initial latent representation $\mathbf{Z}^{\text{iter } 0}$ from SHRED, the SINDy model first provides an initial estimate of an ensemble of SINDy coefficients $\{\hat{\Xi}^b\}_{b=1}^{N_b}$ with N_b members. Then, the parameters of SHRED will be updated towards the dynamics simulated by $\{\hat{\Xi}^b\}_{b=1}^{N_b}$ learned from the first iteration, which generates another latent representation trajectory $\mathbf{Z}^{\text{iter } 1}$. We iterate this procedure and jointly optimize the following loss function to let the SHRED latent representation trajectory approximate the SINDy generated trajectory:

$$\begin{aligned} \mathcal{L} = & \left\| \mathbf{X}_{L+1:T} - f_{\theta_D}(f_{\theta_{\text{GRU}}}(\mathbf{S}_{L+1:T})) \right\|_2^2 + \lambda \|\Xi\|_0 \\ & + \sum_{b=1}^{N_b} \left\| \mathbf{Z}_{T+1}^{\text{GRU}} - \left(\mathbf{Z}_T^{\text{GRU}} + \sum_{i=0}^{k-1} \Theta(\mathbf{Z}_{T+ih}) \Xi^{(b)} h \right) \right\|_2^2, \end{aligned} \quad (3.3)$$

where $\mathbf{X}_{L+1:T} = \{\mathbf{X}_i\}_{i=L+1}^T$ and $\mathbf{S}_{L+1:T} = \{\mathbf{s}_{t_i-L:t_i}\}_{t_i=L+1}^T$. Here, $\mathbf{Z}_{T+ih} = \mathbf{Z}_T + \Theta(\mathbf{Z}_{T+(i-1)h})\Xi h$, $h = \frac{\Delta t}{k}$, and $\mathbf{Z}^{\text{GRU}}$ denotes the GRU-inferred latent space computed from the real observation data. We note here that during neural network training, the total loss is computed as the summation of $\mathcal{L}$ over the entire dataset $\{\mathbf{X}_i\}_{i=1}^T$.

Latent space linearization via the Koopman operator

Koopman operator theory [155, 156] provides an alternative approach to solving these problems by linearizing the underlying dynamics. The linearized embedding is theoretically grounded to be able to represent nonlinear dynamics in a linear framework, which is desirable for many applications in science and engineering, including control, robotics, weather modeling, and so on. From the transformed measurements $\mathbf{z} = g(\mathbf{X})$ from the true system $\mathbf{X}$, the Koopman operator $\mathcal{K}$ is an infinite-dimensional linear operator given by $\mathcal{K}g := g \circ \mathbf{F}$ where $\mathbf{F}(\cdot)$ describes the law of the dynamical system in its original space that $\mathbf{X}_{t+1} = \mathbf{F}(\mathbf{X}_t)$. The Koopman operator enables a coordinate transformation from $\mathbf{X}$ to a linearized latent space with dynamics $\mathcal{K}g(\mathbf{X}_t) = g(\mathbf{X}_{t+1})$.

However, it is generally impossible to obtain the exact form of this infinite-dimensional operator, so a typical strategy is to find a finite-dimensional approximation of the Koopman operator by means of data-driven approaches [31]. In Koopman-SHRED, we utilize the GRU unit to approximate the eigenfunctions. In the latent space, we enforce and learn the linear dynamics, represented by a matrix $\mathbf{K}$, which corresponds to the latent space evolution derived from the eigenfunctions. We keep the interpretability of the model via a parsimonious latent space. In implementation, a simple strategy is to adapt the SINDy-unit with only the linear terms. This models a continuous version of Koopman generator that $\frac{d}{dt}z(x(t)) = \mathcal{G}_t z(x(t))$ where the corresponding Koopman operator $\mathcal{K}_t = e^{t\mathcal{G}}$. Then, we follow a similar practice in SINDy-SHRED, updating the Koopman loss function:

$$\mathcal{L} = \|\mathbf{X}_{L+1:T} - f_{\theta_D}(f_{\theta_{\text{GRU}}}(\mathbf{S}_{L+1:T}))\|_2^2 + \|\mathbf{Z}_{T+1}^{\text{GRU}} - \mathbf{K}\mathbf{Z}_T^{\text{GRU}}\|_2^2.$$

To enable continuous spectrum for better approximation for the Koopman operator, one could adapt an additional neural network to learn the eigenvalues λ_i 's and form the linear dynamics $\mathbf{K}$ from the learned eigenvalues [186]. We also provide detailed implementation and experimental results in the SI Appendix. This flexibility allows the system to learn dynamical systems in a more general setting, accommodating various initial conditions and experimental setups. In our experimental study, we observe that this strategy may perform comparably to learning a fixed $\mathbf{K}$ for linear dynamics under fixed experimental environments.

To summarize, the principal advantage of Koopman-SHRED over SINDy-SHRED lies in its enforcement that the latent dynamics of the recurrent unit are approximately linear. This linearity enables stability analysis and simulation-free predictions. However, this constraint also limits the expressive capacity of the learned model, and in some cases, the flexibility of a nonlinear representation remains essential for accurate modeling. As a practical guideline, Koopman-SHRED should be employed when its performance is adequate, with SINDy-SHRED serving as a more expressive alternative when additional modeling complexity is required.

3.3.3 Theoretical analysis on dynamical system learning

Suppose that the dynamical system has the form $\dot{z} = f(z)$, and we have measurements of $z = \{z_1, z_2, \dots, z_t, z_{t+1}, \dots, z_T\}$ with time gap Δt . The empirical loss function $L_n(\cdot)$ of a function f given the empirical distribution P_n which consists of data samples $\mathcal{D} = \{(z_i, \dot{z}_i)\}_{i=1}^n$ is $L_n(f) = \frac{1}{n} \sum_{i=1}^n \ell(f(z_i), \dot{z}_i)$. The true loss is $L(f) = \mathbb{E}[\ell(f(z), \dot{z})]$.

From empirical process theory, there is a generalization gap between $L_n(f)$ and $L(f)$, which is correlated with the expressive power of the functional class [278, 274]. When the expressive power of the estimator is excessive, we expect that local perturbations can cause significant shifts in performance, leading to a larger generalization error. Therefore, although neural networks are universal approximators [134], they can perform poorly in extrapolation. SINDy effectively avoids this issue because the functional class is moderately sized, which will help to control the generalization error. Additionally, the continuous formulation of SINDy also enables prediction at arbitrary temporal resolutions. It allows physically meaningful forecasts even beyond the sampling rate of the training data.

The following theorems (assumptions in the SI Appendix) present the error bounds for dynamical system learning with the SINDy-class and the neural network class respectively. We consider Gaussian residuals on z to reflect independent measurement noise commonly arising from sensing devices.

Theorem 1. *For a SINDy-class model with a library of functions $\Theta(z)$, and we estimate the coefficient vector ξ by $\hat{\xi}$ using least squares. Let ξ^* be the true coefficient vector. Under the assumptions in the SI Appendix, we have the expected error in predicting the dynamical system after time T is bounded by $E\|\hat{z}(T) - z(T)\| \leq O\left(e^{LT} s \sqrt{\frac{p}{n}}\right)$ where L is the Lipschitz constant of the system, s is the level of noise, p is the number of functions in the library, and n is the number of samples.*

Furthermore, with probability $1 - \delta$, the error in predicting the dynamical system after time T is bounded by

$$\|\hat{z}(T) - z(T)\|_2 \leq O\left(e^{LT} s \sqrt{\frac{p}{n} \log\left(\frac{1}{\delta}\right)}\right). \quad (3.4)$$

Theorem 2. *For a neural network model with k layers of ReLU activation functions and parameters $\theta = (\mathbf{W}_1, \dots, \mathbf{W}_k)$, which computes functions $f(\mathbf{x}; \theta) = \sigma_k(\mathbf{W}_k \sigma_{k-1}(\mathbf{W}_{k-1} \cdots \sigma_1(\mathbf{W}_1 \mathbf{x})))$.*

Under the assumptions in the SI Appendix, with probability at least $1 - \delta$, the error in predicting the dynamical system after $H = T/\Delta t$ steps is bounded by

$$\mathbb{E} [\|\hat{z}(T) - z(T)\|_2^2] \leq O \left((\log n)^4 B^{k(H+1)} \sqrt{\frac{k}{n}} + \frac{\log(1/\delta)}{n} \right),$$

where n is the number of samples.

From the error bounds above, we notice that compared to traditional learning problems (e.g. classification), dynamical system learning is more challenging as the error grows exponentially with increasing time horizon even in linear settings. From the error bound, it becomes evident why neural networks may encounter challenges. First, for the constant term B^{kH} , the norm of weights B and the number of layers k are expected to increase significantly for complex systems. Second, the rate of convergence of neural network learning is generally slower than that of linear least-squares methods ($n^{-\frac{1}{4}}$ versus $n^{-\frac{1}{2}}$), which requires substantially more training samples. Moreover, the theoretical bounds on neural networks can primarily control the expected trajectory error, but has a limited route to bound the supremum error (as in Thm. 3), which can quickly amplify and lead to divergence in dynamical system predictions. The usage of popular transformer networks tend to memorize instead of learning the mechanism, which can produce poor extrapolations from unseen initial conditions [308, 313, 323].

3.4 Computational Experiments

In the following, we perform case studies across a range of scientific and engineering problems. We begin with building models directly from video data, which includes flow around a cylinder and a pendulum. We then consider real-world sea-surface temperature data and conclude with turbulent flow examples. The experiments are all executed on a Linux system with a single NVIDIA GeForce RTX 2080 Ti GPU.

3.4.1 GoPro physics video 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 [3]. The Reynolds number is 171 in the experiment. The dataset contains 11 seconds of video taken at 30 frames per second (FPS). We transfer the original RGB 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 the dataset and SINDy-SHRED in the SI Appendix (Page 17). **SINDy-SHRED discovery** We define the representation of the latent state space as (z_1, z_2, z_3, z_4) . We discover the following dynamical system:

$$\begin{cases} \dot{z}_1 &= -0.69z_2 + 0.98z_3 - 0.40z_4, \\ \dot{z}_2 &= 1.00z_1 - 0.78z_3 - 0.31z_2z_3^2, \\ \dot{z}_3 &= -1.029z_1 + 0.59z_2 + 0.41z_4. \\ \dot{z}_4 &= -0.26z_1^2 - 0.29z_2^2z_3 - 0.39z_3^3. \end{cases} \quad (3.5)$$

The identified nonlinear system has two fixed points, $\mathbf{z} = 0$ and $\mathbf{z} = \begin{pmatrix} -0.11 & -0.23 & -0.14 & 0.05 \end{pmatrix}^T$ which are unstable.

Koopman-SHRED discovery We define the latent states as $(z_1, z_2, z_3, z_4, z_5, z_6)$. We discover

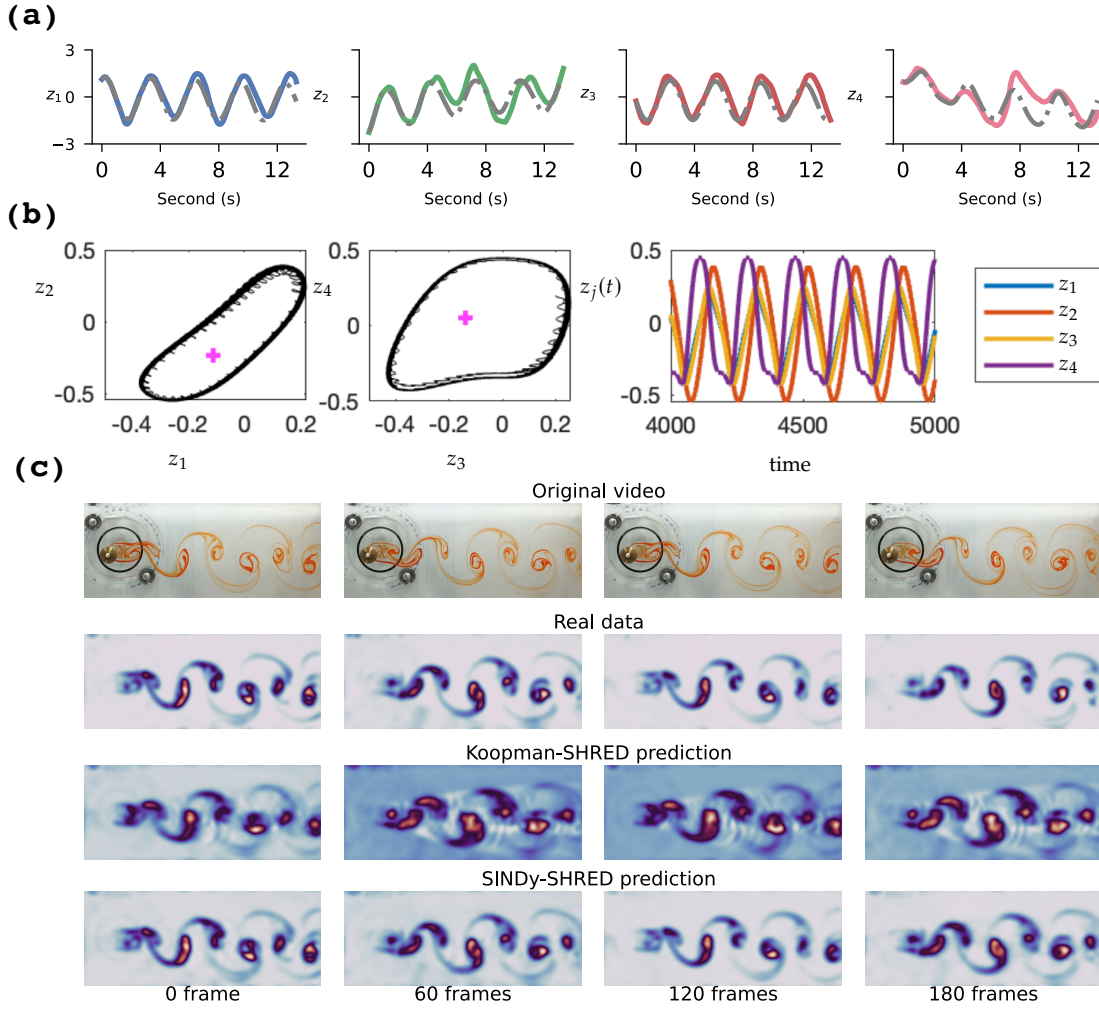


Figure 3.2: (a) Extrapolation of latent representation in SINDy-SHRED from the discovered dynamical system for flow over a cylinder data. Colored: latent representation. Grey: SINDy extrapolation from derived latent dynamical system. (b) Evolution of dynamical system (3.5). The left two panels are the phase plane z_1 vs z_2 and z_3 vs z_4 respectively. The magenta plus symbols are the fixed points. The nonlinear limit cycle behavior is shown in the right panel for $z_j(t)$. (c) Long-term pixel space video prediction via SINDy-SHRED. We demonstrate the forward prediction outcome up to 180 frames.

the following dynamics:

$$\begin{cases} \dot{z}_1 &= -0.920z_4 + 0.620z_5, \\ \dot{z}_2 &= -0.163z_1 + 0.817z_4 + 0.778z_6, \\ \dot{z}_3 &= -0.462z_4 - 1.026z_6, \\ \dot{z}_4 &= 1.791z_1 + 0.307z_6, \\ \dot{z}_5 &= -0.969z_1 + 0.548z_6, \\ \dot{z}_6 &= -0.800z_2 - 0.915z_5. \end{cases} \quad (3.6)$$

The analytic solution has the form

$$\begin{aligned} \mathbf{z}(t) = & c_1 \mathbf{v}_1 \cos(\omega_1 t) e^{-\lambda_1 t} + c_2 \mathbf{v}_2 \sin(\omega_1 t) e^{-\lambda_1 t} \\ & + c_3 \mathbf{v}_3 \cos(\omega_3 t) e^{\lambda_3 t} + c_4 \mathbf{v}_4 \sin(\omega_3 t) e^{\lambda_3 t} + c_5 \mathbf{v}_5 e^{-\lambda_5 t} \end{aligned}$$

where $\omega_1 = 1.52$, $\omega_3 = 1.05$, $\lambda_1 = 0.01$, $\lambda_3 = 0.11$, and $\lambda_5 = -0.20$. The explicit solution is given in the SI.

The flow over a cylinder model contains both linear and nonlinear interactions. In Eqn. 3.5, 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 further present a linear model derived from Koopman-SHRED in Eqn. 3.6 with its latent space evolution demonstrated in SI Appendix Fig. S22. We show 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. 3.2 (a), the learned ODE closely follows the dynamics of z_1 and z_3 up to around 7 seconds, where the relative mean-squared error (rMSE) remains below 0.1%; z_2 and z_4 also have close extrapolation up to 4 seconds with rMSE remains below 0.1%. Fig. 3.2 (b) shows that the learned nonlinear dynamical system. The discovered latent ODE exhibits a closed limit cycle as demonstrated in SI Appendix Fig. S22. The Koopman-SHRED approximation aligns with the overall trend for each latent coordinate but exhibits phase and amplitude drift.

This learned representation nicely predicts the future frames in pixel space. In SINDy-SHRED, the shallow decoder prediction has an averaged MSE error of 0.030 over the entire available trajectory. Koopman-SHRED has relatively worse prediction error with an averaged MSE error of 0.053. In Fig. 3.2 (c), we observe that the autoregressively generated prediction frames follow the same temporal progression and vortex patterns as the true data. Furthermore in SI Appendix Fig. S24, we find that the long-term predictions can preserve the vortex pattern without collapsing to an averaged static frame after 1,000 steps, which is out of the size of the original dataset. The sensor-level prediction in SI Appendix Fig. S23 further demonstrates the accuracy of reconstruction in detail.

3.4.2 Prediction and baseline study of single shot pendulum video

In this subsection, we compare the performance of SINDy-SHRED to existing state-of-the-art learning algorithms. In the following, we demonstrate the result of video prediction on the pendulum data using ResNet [120], convolutional LSTM (convLSTM) [246], and PredRNN [287], and SimVP [105]. 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. We include the details of the experimental settings of the dataset and SINDy-SHRED in the SI Appendix (Page 17). 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 (shown in Fig. 3.3) is given by:

$$\ddot{z} = 0.17\dot{z}^2 - 0.06\dot{z}^3 - 10.87 \sin(z) + 0.48 \sin(\dot{z}), \quad (3.7)$$

which includes complex damping effects via $\sin(\dot{z})$, $\dot{z}^2$, $\dot{z}^3$, in contrast to the commonly assumed linear damping.

As shown in Table 3.1, 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 accumulates for longer-term predictions. This is also observable from the prediction in the pixel space as shown in Fig. 3.3 (b). SINDy-SHRED is the only method that does not produce collapsed longer-term predictions. Interestingly, we observe that incorporating nonlinear terms in the SINDy library is crucial in this case, as Koopman-SHRED struggles to reconstruct the finer details. We attribute this to the fact that strong linear regularization may over-regularize and oversimplify the complex dynamics, leading to a reduction in predictive accuracy. Similar behaviors for Koopman-SHRED are observed when encoding the continuous spectrum using an additional shallow network. In SI Appendix Fig. S27, 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

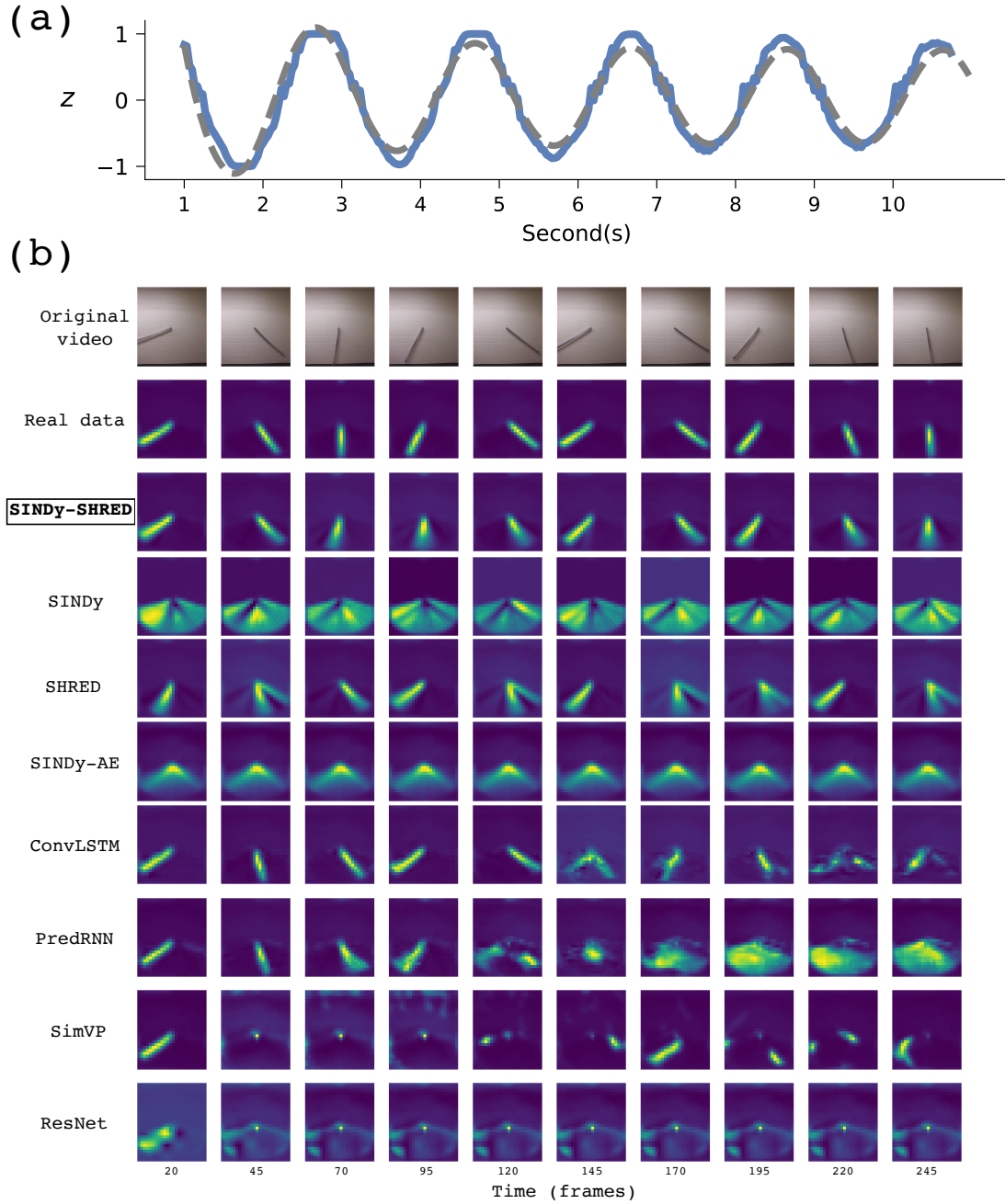


Figure 3.3: (a) Extrapolation of latent representation in SINDy-SHRED from the discovered dynamical system for the pendulum moving data. Blue: true latent representation. Grey: SINDy extrapolation. (b) The pendulum video generation outcome from ResNet, SimVP, ConvLSTM, PredRNN, SINDy, SHRED, SINDy-autoencoder, and SINDy-SHRED from frame 20 to frame 245.

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, it should be noted that 2D

Models	Params	Training time	$T = [0, 100]$	$T = [100, 200]$	$T = [200, 275]$	Total
SINDy (linear) [32]	420K	8 mins	4.04×10^{-2}	4.20×10^{-2}	4.21×10^{-2}	4.14×10^{-2}
SHRED [292]	240K	8 mins	1.24×10^{-2}	2.08×10^{-2}	2.83×10^{-2}	1.66×10^{-2}
SINDy-AE [46]	200K	6 mins	1.69×10^{-2}	1.51×10^{-2}	1.52×10^{-2}	1.58×10^{-2}
ResNet [120]	2.7M	24 mins	2.08×10^{-2}	1.88×10^{-2}	2.05×10^{-2}	2.00×10^{-2}
SimVP [105]	460K	30 mins	2.29×10^{-2}	2.47×10^{-2}	2.83×10^{-2}	2.53×10^{-2}
PredRNN [287]	444K	178 mins	1.02×10^{-2}	1.79×10^{-2}	1.69×10^{-2}	1.48×10^{-2}
ConvLSTM [246]	260K	100 mins	9.24×10^{-3}	1.86×10^{-2}	1.99×10^{-2}	1.55×10^{-2}
SINDy-SHRED*	44K	17 mins	1.70×10^{-2}	9.36×10^{-3}	5.31×10^{-3}	1.05×10^{-2}

Table 3.1: Comparison table of SINDy-SHRED to baseline methods for parameter size, training time, and error over different time horizons.

convolution is much more computationally expensive. For larger spatiotemporal domains (e.g. the SST example and 3D ozone data), the direct combination of 3D convolution and recurrent layers become computationally demanding [197], as both memory and runtime scale cubically with spatial resolution and linearly with temporal history [63]. Similar computational issues will occur for diffusion models as a denoising diffusion probabilistic model can take 20 hours to sample 50K images of size 32×32 [127]. For scientific datasets that easily reach terabyte scale, these methods can become computationally intensive and challenging to scale to long-horizon, high-resolution scientific applications [86, 305, 205, 133, 229, 235]. By constraining the library to be linear, a SINDy model could predict the high-dimensional system forward. However, it is inaccurate for short- and long-term forecasts, and loses interpretability due to the required large number of parameters. SHRED also struggles with long-term rollouts, as it is primarily focused on sparse sensing problems rather than full spatiotemporal modeling. SINDy-AE often converges to an incorrect latent dynamical model, leading to degraded rollout performance. The detailed training and model configurations are provided in the SI Appendix. In summary, we observe that SINDy-SHRED is not only producing long-term predictions with smaller mean squared error, but is also fast to execute and smaller in size.

3.4.3 NOAA sea-surface temperature data

The third 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 [225]. The reanalysis dataset combines in situ and satellite data through optimal interpolation, which is represented by a 180×360 grid. Within these grids, 44,219 of its 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. 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. We include the further details of the dataset and the experimental settings of SINDy-SHRED in the SI Appendix (Page 17). From the discovered coordinate system, we define the latent representation of physics to be (z_1, z_2, z_3) . The dynamics evolves via the following set of equations:

$$\begin{cases} \dot{z}_1 &= 4.68z_2 - 2.37z_3, \\ \dot{z}_2 &= -3.10z_1 + 3.25z_3, \\ \dot{z}_3 &= 2.72z_1 - 5.55z_2, \end{cases} \quad (3.8)$$

where the analytic solution to this system of ODEs is

$$\mathbf{z}(t) = c_1 \mathbf{v}_1 \cos(\omega_1 t) e^{-\lambda_1 t} + c_2 \mathbf{v}_2 \sin(\omega_1 t) e^{-\lambda_1 t} + c_3 \mathbf{v}_3 e^{\lambda_3 t}.$$

The value of $\omega_1 = 1.99\pi$, approximately corresponding to the expected period of 1 year. $\lambda_1 = 0.00763$ indicating a slow decay of the oscillatory mode with half-life 90.84 years. $\lambda_3 = 0.01527$ indicating global increases in temperature with doubling time 45.39 years. The explicit solution is given in SI Appendix (Page 25).

The discovery of a linear system describing the evolution of the latent state is in line with prior work on SST data [68] in which it was assumed that the underlying physics is an advection-diffusion PDE. In Fig. 3.4 (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 in SI Appendix Fig. S17. It is observable how the discovered law is close to the true evolution

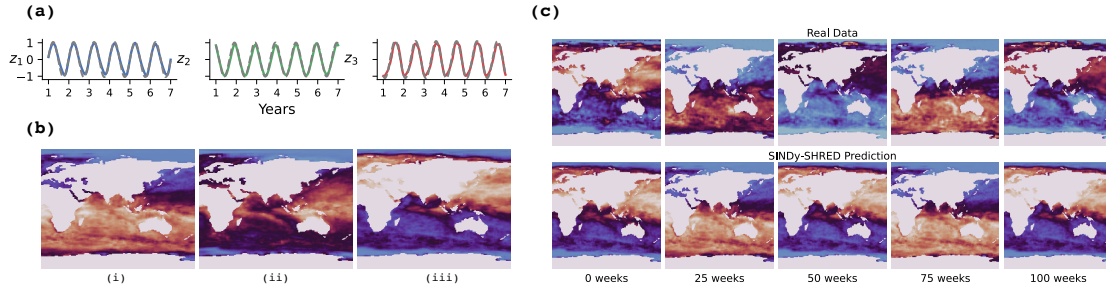


Figure 3.4: (a) Extrapolation of latent representation in SINDy-SHRED from the discovered dynamical system for SST. Colored: true latent representation. Grey: SINDy extrapolation. (b) Decoder reconstruction of three independent directions z_1, z_2, z_3 in the latent space. (c) Long-term global sea-surface temperature prediction via SINDy-SHRED from week 0 to week 100. We crop the global map for better visualization.

of latent states and, critically, there appears to be minimal phase slipping. By extrapolating the latent state space via forward integration, we can apply the shallow decoder to return forecasts of the future spatial domain. We denormalize the temperature values during evaluation with temperature ranges from $(-1.8, 35.62)$. Doing so, we find an averaged MSE error of 0.57 ± 0.10 for 318 weeks in the test dataset. This corresponds to a mean absolute error of 0.75°C for each sensor points with relative error 2.01%. The eigenmodes of the SST data is visualized by feeding unit vectors along the independent latent directions z_1, z_2, z_3 into the decoder in Fig. 3.4 (b). In Fig. 3.4 (c), we show SINDy-SHRED produces stable long-term predictions for SST data. We further include SI Appendix Fig. S18 to demonstrate the extrapolation from randomly sampled sensor points and SI Appendix Fig. S9 to show the error evolution with time. The sensor level prediction is based on the global prediction of the future frame, and we visualize the signal trajectory of randomly sampled testing sensor locations. We find SINDy-SHRED is robust for out-of-distribution sensors, with reasonable extrapolations even in the presence of anomalous events.

3.4.4 Predicting isotropic turbulence flow

The isotropic turbulence flow data is a data simulation from the pressure field of a forced isotropic turbulent flow. We obtain 2D snapshots of the original pressure field data in [170, 309]

$$\nabla^2(p/\rho) = (\Omega - \epsilon/\nu)/2, \quad (3.9)$$

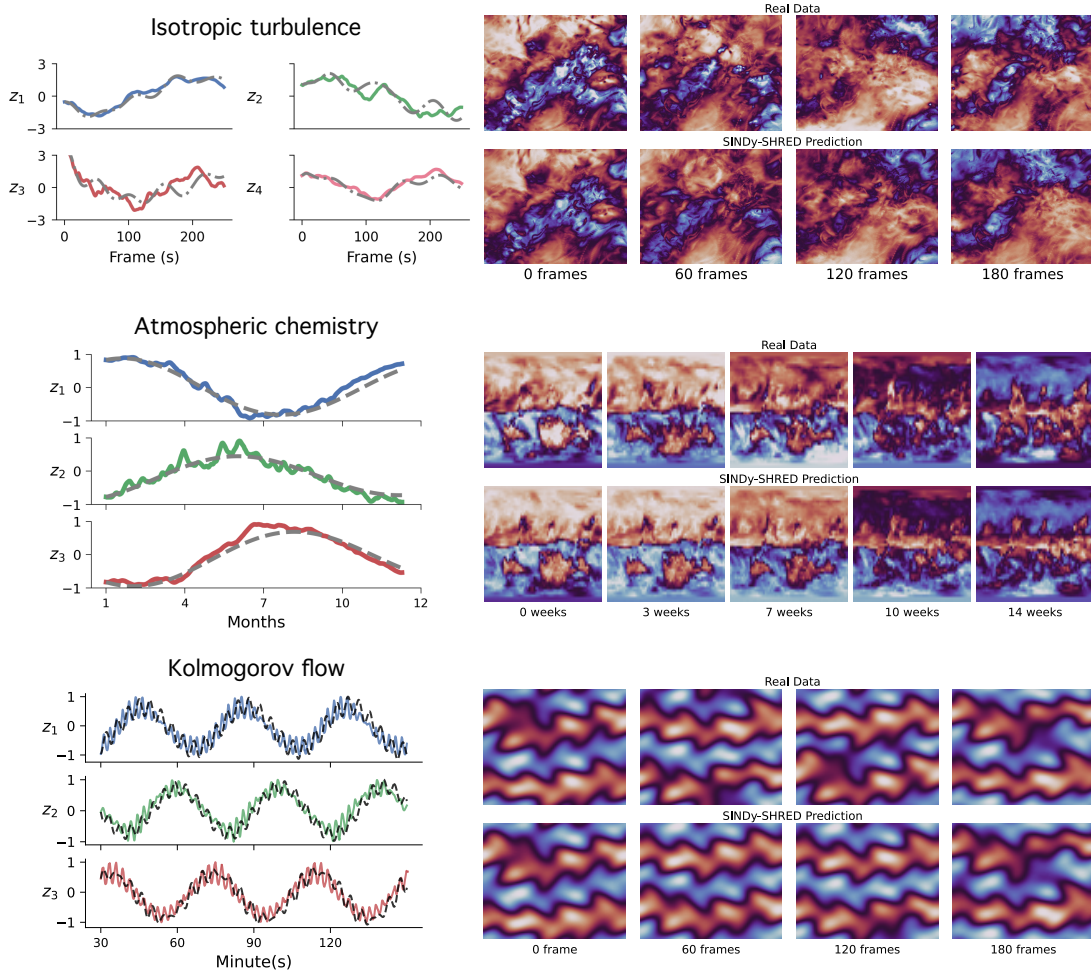


Figure 3.5: Visualizations of latent spaces and predictions for SINDy-SHRED applied to PDE simulations of isotropic turbulence, atmospheric chemistry, and 2D Kolmogorov flow datasets. **Left:** Extrapolation of latent representations using SINDy-SHRED from the discovered dynamical system for the ozone data, showing the true latent representation (colored lines) and SINDy extrapolation (grey lines). **Right:** Long-term spatiotemporal predictions from SINDy-SHRED compared to ground truth. The details of the 2D Kolmogorov experiment can be found in the SI Appendix.

where p is the pressure, ρ is the density of the fluid, Ω is the vorticity, ϵ is the dissipation rate, and ν is the kinematic viscosity. The flow was generated with 1024^3 nodes with the Taylor-scale Reynolds number fluctuating around 433. The pressure field provides a scalar representation that captures the essential multi-scale dynamics while simplifying analysis compared to vector velocity fields. We process the data to shrink the spatial resolution to 350×350 , while keeping the time step to be 0.002. We standardize the data within the range of $(0, 1)$ with 50 sensors out of 122,500 grid points (0.4%). We include details of the dataset

and SINDy-SHRED setup in the SI Appendix (Page 17).

Our approximation of the latent representation has 8 oscillatory modes, and permits the following analytic solution:

$$\begin{aligned} \mathbf{z}(t) = & \sum_{i=1}^7 (c_{2i-1} \mathbf{v}_{2i-1} \cos(\omega_i t) e^{\lambda_i t} + c_{2i} \mathbf{v}_{2i} \sin(\omega_i t) e^{\lambda_i t}) \\ & + c_{15} \mathbf{v}_{15} e^{\lambda_8 t} + c_{16} \mathbf{v}_{16} e^{\lambda_9 t}, \end{aligned} \quad (3.10)$$

where $\omega_1 = 9.39$, $\lambda_1 = 0.47$, $\omega_2 = 11.90$, $\lambda_2 = 0.05$, $\omega_3 = 13.42$, $\lambda_3 = 0.03$, $\omega_4 = 3.46$, $\lambda_4 = -0.04$, $\omega_5 = 5.44$, $\lambda_5 = -0.27$, $\omega_6 = 8.30$, $\lambda_6 = -0.75$, $\omega_7 = 18.72$, $\lambda_7 = 1.37$, $\lambda_8 = -0.26$, $\lambda_9 = -3.39$, with coefficients c_i ' and eigenvectors $\mathbf{v}_i$ (see SI Appendix). Since the model is learned from a local 2D snapshot of a larger flow, positive eigenvalues reflect local amplification and do not imply instability of the full statistically stationary system.

The isotropic turbulent flow exhibits chaotic behavior which makes the prediction challenging. However, a linear model still captures the governing dynamics nicely and provides a stable prediction for approximately 200 frames. In Eqn. 3.10, we find three increasing oscillatory modes and five decreasing oscillatory modes. For pixel space prediction, SINDy-SHRED prediction has an averaged MSE error of 0.03 over the entire dataset. In Fig. 3.5, we find that the predictions have small MSE up to 180 frames, which closely follows the governing trend of turbulent flow. The sensor-level prediction in SI Appendix Fig. S26 further demonstrates the details of the signal prediction. Three of the oscillatory modes demonstrate growing trends. Four of the oscillatory modes decay, as well as the exponential modes. We note here that due to the complex and chaotic nature of the data, accurate sensor-level prediction is particularly challenging. Despite this complexity, the SINDy-SHRED model still provides a stable extrapolation of the overall trends.

3.4.5 3D Atmospheric ozone concentration

The atmospheric ozone concentration dataset [21] 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 the dataset and SINDy-SHRED in the SI Appendix (Page 17). The converged latent representation gives:

$$\begin{cases} \dot{z}_1 &= -0.002 - 0.013z_2 + 0.007z_3, \\ \dot{z}_2 &= -0.001z_1 + 0.004z_2 - 0.008z_3, \\ \dot{z}_3 &= 0.002 + 0.012z_2 - 0.005z_3. \end{cases} \quad (3.11)$$

The analytic solution to this system of ODEs has the form

$$\begin{aligned} \mathbf{z}(t) &= c_1 \mathbf{v}_1 \cos(\omega_1 t) e^{-\lambda_1 t} + c_2 \mathbf{v}_2 \sin(\omega_1 t) e^{-\lambda_1 t} + c_3 \mathbf{v}_3 e^{-\lambda_3 t} \\ &+ \int_0^t (c_4 \mathbf{v}_1 \cos(\omega_1 \tau) e^{-\lambda_1 \tau} + c_5 \mathbf{v}_2 \sin(\omega_1 \tau) e^{-\lambda_1 \tau} + c_6 \mathbf{v}_3 e^{-\lambda_3 \tau}) d\tau \end{aligned}$$

where $\omega_1 = 0.0079$, $\lambda_1 = 0.003$, and $\lambda_3 = 0.003$. The complete solution is given in SI Appendix Eqn. 64.

Unlike traditional architectures for similar problems, which includes expensive operations such as the 3D convolution, SINDy-SHRED provides an efficient way of training. 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 SI Appendix Fig. S19). From the extrapolation of the latent state space, the shallow decoder prediction has an averaged MSE error of $1.5e^{-2}$. In Fig. 3.5, we visualize the shallow decoder prediction up to 14 weeks. In Fig. 3.5, we reconstruct the sensor-level predictions. The observations are much noisier than the SST data, but SINDy-SHRED provides a smoothed extrapolation.

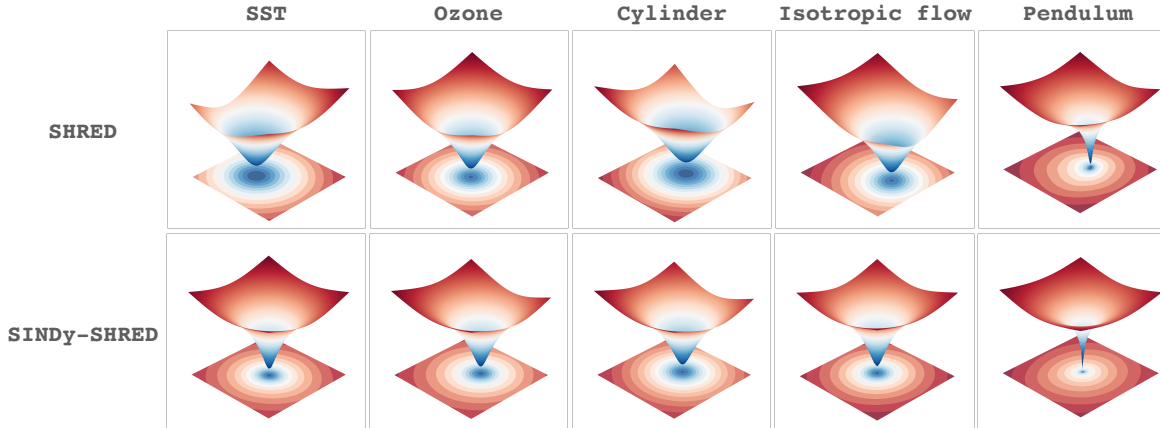


Figure 3.6: Loss landscape plot of SHRED (top row) and SINDy-SHRED (bottom row) for all datasets presented: sea-surface temperature, Ozone, flow over a cylinder, isotropic flow, and pendulum.

3.4.6 Visualizing the convex loss landscape

We visualize the loss landscape for each experiment in Fig. 3.6. Following the methodology of [168], this visualization demonstrates the loss landscape along two random directions. By perturbing the model weights along these directions, we obtain a 3D visualization of the loss surface. We observe that, for both SHRED and SINDy-SHRED, the loss landscape is surprisingly smooth and exhibits perturbative convexity, featuring a clear global minima. The visualizations across all datasets consistently demonstrate these favorable convex loss landscapes. In the Appendix, we further visualize different settings of SHRED under various random directions and scales in SI Appendix Fig. S2, S3. From these optimization landscapes, we observe that the SHRED architecture encourages flat minimizers and reduces the potential for chaotic training behavior. This property also provides SHRED with properties including hyperparameter stability, reliable statistical inference, and guaranteed data recovery.

3.5 Discussion and conclusion

In this paper, we present the SINDy-SHRED algorithm, which jointly solves the sensing and model identification problem with simple implementation, efficient computation, and robust performance. Specifically, SINDy-SHRED (Koopman-SHRED) learns a latent coordinate system for which parsimonious, low-dimensional governing equations, either linear or

nonlinear, can be constructed. The SINDy-SHRED model has strong predictive capabilities and stable roll outs by identifying the underlying physics. Through numerous experiments on challenging real-world data, we show that our method can produce robust and accurate long-term predictions for a variety of complex problems, including global sea-surface temperature, turbulent flows, and videos of physics, which includes recordings of flow over a cylinder and a moving pendulum. The simplicity and robustness of the architecture, along with its performance on real videos and measurements, make it highly compelling as a paradigm for learning physics models directly from data.

Learning physics and obtaining insights from well-established physics problems is challenging, especially given the extensive efforts and long-standing interest from the broader community. Despite this, SINDy-SHRED has been demonstrated to derive reliable physics models for every single example considered in this paper. For instance, directly from video, SINDy-SHRED identified two potential models to capture the flow around a cylinder: (i) a four-dimensional nonlinear ODE system containing a limit cycle modeling the nonlinear oscillations induced by the von Karman vortex shedding, and (ii) a linear six-dimensional Koopman model which can be solved in closed form. Both models provide alternative formulations of the vortex shedding dynamics, despite decades of research devoted to understanding the bifurcation underlying this behavior. Moreover, the physics discovery is done solely from pixel space, a remarkable achievement that stands in contrast to previous efforts requiring access to the full PDE simulation fields. In the pendulum example, SINDy-SHRED learns a nonlinear damping term that captures the complex motion of the system, improving on the typical linear damping assumption. Beyond video data, SINDy-SHRED also demonstrates exceptional performance in modeling turbulent flows. Remarkably, the latent space dynamics reveal that eight modes suffice to accurately capture the dynamics of isotropic turbulence. Even global sea-surface temperature is captured with a three-dimensional linear ODE system that one might solve in a first course on ODEs. These results are astounding given the complex data, highlighting the ability of SINDy-SHRED to extract governing equations from well-known and important scientific problems.

In addition to learning interpretable physics models, SINDy-SHRED is simple and robust

to train. As shown above and in the supplement, this largely stems from the observed globally convex loss landscape, which is rare among all existing deep learning models as noted in [168]. In general, a major goal in designing deep learning models is to engineer the loss landscape to align with the inherent characteristics of the data, thereby facilitating stable training, robust hyper-parameter tuning, and generalizable performance [284, 74]. SINDy-SHRED naturally exhibits an ideal loss landscape that is globally convex, which is what the deep learning community strives to engineer. Although exploring the underlying mechanisms is beyond the scope of the current study, future work will investigate how the SINDy-SHRED architecture induces this ideal loss landscape. Moreover, SINDy-SHRED is grounded by rigorous convergence guarantees for the SINDy-class of models, offering theoretical advantages over competing neural network-based approaches. These theoretical results offer precise error bounds and the performance for long-term prediction. As a result, SINDy-SHRED is well-grounded in theory, providing a solid foundation for further development and widespread practical applications.

Most importantly, in practice, SINDy-SHRED achieves state-of-the-art performance in long-term autoregressive video prediction, outperforming ConvLSTM, PredRNN, ResNet, and SimVP while maintaining the lowest computational cost, smallest model size and minimal training time simultaneously. Notably, SINDy-SHRED outperforms the leading methods on all metrics of interest without any hyper-parameter tuning. Moreover, this performance is achieved while producing an interpretable model of the underlying dynamics, effectively capturing the actual governing physics, a contribution that competing methods do not offer. This enables stable, long-time rollouts of video prediction. Such performance advances the goal of GoPro Physics, which aims to discover governing equations directly from few-shot video data streams.

SINDy-SHRED yields interpretable latent dynamics, while using a data-driven surrogate to model the coordinate transformation. The GRU network approximates the mapping from sensor measurements to latent states with limited interpretability on its learned weights. We also note that SINDy-SHRED explicitly enforces the latent dynamics to follow a single ODE uniformly. If the true system exhibits switching dynamics, this assumption is violated unless

appropriate regime separation or preprocessing is applied.

To conclude, the SHRED architecture and its variants have proven to be highly effective algorithms for sensing [292, 78], parametric model reduction [270, 161, 226] and now, as demonstrated in this work, physics discovery. The SINDy-SHRED and Koopman-SHRED architectures are robust, stable, and computationally efficient. This allows for laptop level computing along with afternoon-level tuning-even on exceptionally challenging datasets. Its ease of use and reliability position it as a compelling algorithm to be the paradigm for general-purpose physics model discovery.

3.6 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 a difficult task for neural networks. The dynamical system x is generated via the following equation:

$$\ddot{z} = -\sin(z). \quad (3.12)$$

We implement a GRU network in the following, which contains 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. The input sequences are normalized to $[0, 1]$. The SINDy library consists of 12 candidate functions and is denoted by $\Theta(x) = [1, z, \dot{z}, z^2, z\dot{z}, \dot{z}^2, z^3, z^2\dot{z}, z\dot{z}^2, \dot{z}^3, \sin(z), \sin(\dot{z})]$. We apply a sequentially thresholded least-squares algorithm to perform sparsity promoting regression with threshold 0.5 and ℓ_2 regularization 0.05.

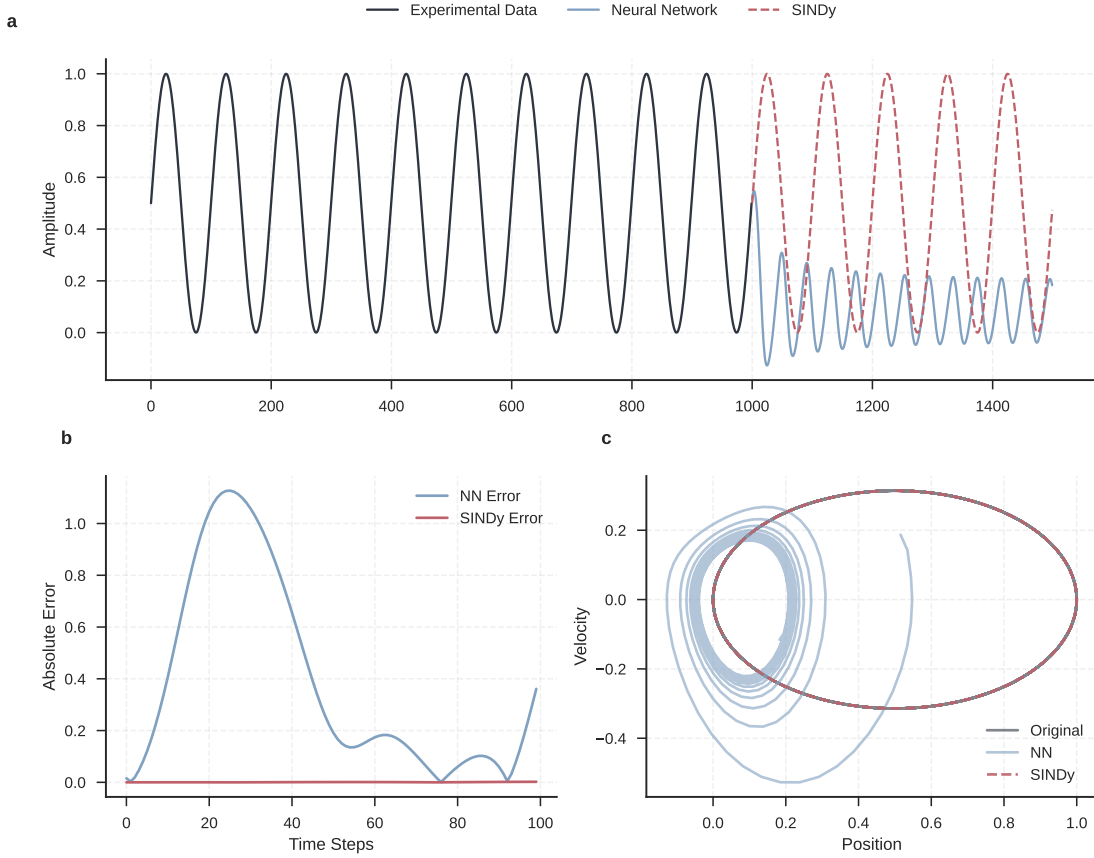


Figure S1: Fitting a simple sine function using SINDy and the GRU network.

From Fig. S1, we observe that SINDy approximation largely outperforms the GRU network. The extrapolation of GRU is poor due to the overly strong expressive power of this type of functional. We locate this to be a critical reason that most neural networks fail to capture physics nicely. Additionally, it is not clear for neural networks to estimate the dynamical system from the discrete setup $(t_0, t_1, \dots, t_T)$ to a continuous one, while SINDy provides a natural transition into a continuous-time formulation. The SINDy prediction could also remarkably improve generalizability, which could enhance the performance in long-term prediction.

3.7 Theoretical justification on the generalization error between SINDy and the neural networks

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.

3.7.1 Theoretical setup

The system we wish to study has the form that

$$\dot{z}_t = f^*(z), \quad (3.13)$$

which is an ODE describing the trajectory of a dynamical system in a learned latent space and $f^*(\cdot)$ is the target function.

Suppose we have measurements of $\mathbf{Z} = \{z_1, z_2, \dots, z_t, z_{t+1}, \dots, z_T\}$ with time gap Δt , and we wish to predict $z_{t+1}, z_{t+2}, \dots, z_T$.

We first define the SINDy-class functional based on a library of functions $\Theta(\cdot)$. Mathematically, $\mathcal{F}_{\text{SINDy}}$ defines as:

$$\mathcal{F}_{\text{SINDy}} := \{\Theta\xi : \xi \in \mathbb{R}^p\}, \quad (3.14)$$

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 $\mathcal{L}^2(P)$. The space $\mathcal{L}^2(P)$ refers to the set of square-integrable functions with respect to the probability measure P . An example of $\Theta(z)$ is $[z, z^2, z^3, z^4, \sin(z), \cos(z)]$, and this could represent a dynamical system with the following

form

$$\dot{z}_t = a_1 z + a_2 z^2 + a_3 z^3 + a_4 z^4 + a_5 \sin(z) + a_6 \cos(z), \quad (3.15)$$

where $a_1, a_2, \dots, a_6$ are constants. Therefore, the feature matrix $\Theta(\mathbf{Z}) \in \mathbb{R}^{n \times p}$.

We note that $\mathcal{F}_{\text{SINDy}}$ 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 neural network way of learning dynamical systems. Unlike SINDy learning, the neural network collects discrete samples $z_{t-L}, z_2, \dots, z_t$, and to predict z_{t+1} . We consider the k -layer ReLU functional class $\mathcal{F}_{\text{ReLU}}$ to be

$$\mathcal{F}_{\text{ReLU}} := \{z \mapsto \sigma_k(W_k \sigma_{k-1}(W_{k-1} \cdots \sigma_1(W_1 z)))\}, \quad (3.16)$$

where $\theta = (W_1, \dots, W_k)$ with $W_1 \in \mathbb{R}^{p \times d}$, $W_i \in \mathbb{R}^{d \times d}$, $W_k \in \mathbb{R}^{d \times p}$, and $\sigma_i(x) = \max(0, x)$ represents the ReLU function. Instead of modeling the temporal derivatives, this training strategy believes that the following dynamics are strongly tied with the history samples, and autoregressively predicts the future dynamics. This is commonly used in neural networks training for dynamical systems predictions.

For both cases, we define the error of dynamical system simulation as

$$\mathcal{E}(z, t) = z(t) - \hat{z}(t). \quad (3.17)$$

Stability It is important to note that, in the following analysis, we set aside stability issues due to the lack of direct theoretical tools. In practice, we observe that the dynamical systems approximated by neural networks are often exhibit instability. Longer rollouts of a learned dynamical system typically lead to poor performance. This is also reflected in the theory that it is hard to obtain guarantees without both a bounded input domain and bounded target values. There is no clear assurance on how neural networks might perform when getting out of the training input domain [245]. To facilitate the current analysis, we assume a stable

forward simulation scheme by always projecting the $\hat{z}$ back into the input domain. Fully addressing the stability issue for neural networks will require more extensive study in future work.

3.7.2 The error of SINDy-class

In SINDy setting, the optimal functional $\theta^*(\cdot) \in \mathcal{F}_{\text{SINDy}}$ can be parameterized from the following:

$$\theta^*(z) = \Theta(z)\xi^*, \quad (3.18)$$

where ξ^* is a vector in $\mathbb{R}^p$. The functional learning problem shrinks down to a linear parameter estimation problem, and the empirical risk minimizer is the least-square solution that

$$\hat{\xi} = (\Theta(\mathbf{Z})^T \Theta(\mathbf{Z}))^{-1} \Theta(\mathbf{Z})^T \hat{\mathbf{Z}}_t. \quad (3.19)$$

Here, we estimate the real $\dot{z}_t$ from the data using numerical differentiation methods via $\hat{\dot{z}}_t$. We assume a linear model for $\hat{\dot{z}}_t$ that

$$\hat{\dot{z}}_t = \Theta(\mathbf{Z})\xi^* + \epsilon, \quad (3.20)$$

where $\epsilon \sim P$ and P is a noise distribution.

Assumption 1 (Lowest eigenvalue). The lowest eigenvalue of $\Theta(\mathbf{Z})^T \Theta(\mathbf{Z})$ is bounded from below by cn , where $c > 0$, and n is the sample size.

Assumption 2 (Gaussian white noise). The noise vectors $\{\epsilon_i\}_{i=1}^n$ are independent and

$$\epsilon_i \sim \mathcal{N}(0, s^2 I).$$

Assumption 3 (Regularity SINDy function library). There exists a compact set $\mathcal{K} \subset \mathbb{R}^d$ such that both trajectories $\{z(t), \hat{z}(t)\}_{t \in [0, T]}$ remain in $\mathcal{K}$. Moreover, the true vector field

$f^*(z) := \Theta(z)\xi^*$ is L -Lipschitz on $\mathcal{K}$:

$$\|f^*(z) - f^*(z')\|_2 \leq L\|z - z'\|_2, \quad \forall z, z' \in \mathcal{K},$$

and the library is uniformly bounded on $\mathcal{K}$:

$$R := \sup_{z \in \mathcal{K}} \|\Theta(z)\|_{\text{op}} < \infty.$$

Assumption 1 is an identifiability requirement on the design matrix, which prevents nearly collinear libraries and ensures least-squares will have effective information. Assumption 2 models random error from sensor collection as independent isotropic Gaussian. Assumption 3 is a finite-time stability condition. For example, the true system cannot exhibit finite-time blow-up, and, additionally, the learned rollout must be smooth, and remain in a bounded region over the time horizon. This guarantees that the chosen library functions are reasonable, and the model class is not severely misspecified.

Theorem 3. *Suppose we have a SINDy-class functional with a library of functions $\Theta(z)$, and we estimate the coefficient vector ξ by $\hat{\xi}$ using least squares. Let ξ^* be the true coefficient vector. Then, under suitable regularity conditions on $\Theta(z)$ and the noise in the data, we have*

$$E_\epsilon \|\hat{\xi} - \xi^*\|_2 = \mathcal{O} \left(s \sqrt{\frac{p}{n}} \right) \quad (3.21)$$

where C is a constant depending on the properties of $\Theta(z)$, s is the standard deviation of the noise, p is the number of functions in the library $\Theta(z)$, and n is the number of data points.

Given Assumption 1 and Assumption 2, we have the error in predicting the dynamical system after time T is bounded by

$$E_\epsilon \|\hat{z}(T) - z(T)\|_2 \leq \mathcal{O} \left(e^{LT} E \|\hat{\xi} - \xi^*\| \right) = \mathcal{O} \left(e^{LT} s \sqrt{\frac{p}{n}} \right), \quad (3.22)$$

where L is the local Lipschitz constant of the system within the region traversed by the trajectories.

Furthermore, with probability $1 - \delta$, the error in predicting the dynamical system after time T is

bounded by

$$\|\hat{x}(T) - x(T)\|_2 \leq \mathcal{O} \left(e^{LT} s \sqrt{\frac{p}{n} \log \left(\frac{1}{\delta} \right)} \right). \quad (3.23)$$

Proof.

$$\mathbb{E}_\epsilon \left[\|\hat{\xi} - \xi^*\|_2^2 \right] = \mathbb{E}_\epsilon \left[\left\| (\Theta(\mathbf{Z})^T \Theta(\mathbf{Z}))^{-1} \Theta(\mathbf{Z})^T \hat{\mathbf{z}}_t - \xi^* \right\|_2^2 \right] \quad (3.24)$$

$$= \mathbb{E}_\epsilon \left[\left\| (\Theta(\mathbf{Z})^T \Theta(\mathbf{Z}))^{-1} \Theta(\mathbf{Z})^T (\Theta(\mathbf{Z}) \xi^* + \epsilon) - \xi^* \right\|_2^2 \right] \quad (3.25)$$

$$= \mathbb{E}_\epsilon \left[\left\| (\Theta(\mathbf{Z})^T \Theta(\mathbf{Z}))^{-1} \Theta(\mathbf{Z})^T \epsilon \right\|_2^2 \right] \quad (3.26)$$

$$= \mathbb{E}_\epsilon \left[\text{Tr} \left[(\Theta(\mathbf{Z})^T \Theta(\mathbf{Z}))^{-1} \Theta(\mathbf{Z})^T \epsilon \epsilon^T \Theta(\mathbf{Z}) (\Theta(\mathbf{Z})^T \Theta(\mathbf{Z}))^{-1} \right] \right] \quad (3.27)$$

$$= \text{Tr} \left[(\Theta(\mathbf{Z})^T \Theta(\mathbf{Z}))^{-1} \Theta(\mathbf{Z})^T \mathbb{E}_\epsilon [\epsilon \epsilon^T] \Theta(\mathbf{Z}) (\Theta(\mathbf{Z})^T \Theta(\mathbf{Z}))^{-1} \right] \quad (3.28)$$

$$\text{from Assumption 2} = s^2 \text{Tr} \left[(\Theta(\mathbf{Z})^T \Theta(\mathbf{Z}))^{-1} \Theta(\mathbf{Z})^T \Theta(\mathbf{Z}) (\Theta(\mathbf{Z})^T \Theta(\mathbf{Z}))^{-1} \right] \quad (3.29)$$

$$= s^2 \text{Tr} \left[(\Theta(\mathbf{Z})^T \Theta(\mathbf{Z}))^{-1} \right] \quad (3.30)$$

Under Assumption 1 on $\Theta(z)$, the smallest eigenvalues of $\Theta(\mathbf{Z})^T \Theta(\mathbf{Z})$ is lower bounded by cn . Then the largest eigenvalue of $(\Theta(\mathbf{Z})^T \Theta(\mathbf{Z}))^{-1}$ is bounded above by $\frac{1}{cn}$. Therefore,

$$\text{Tr} \left[(\Theta(\mathbf{Z})^T \Theta(\mathbf{Z}))^{-1} \right] \leq \frac{p}{cn}, \quad (3.31)$$

where p is the number of functions in $\Theta(z)$. Therefore,

$$\mathbb{E}_\epsilon \left[\|\hat{\xi} - \xi^*\|_2^2 \right] \leq \frac{s^2 p}{cn} \quad (3.32)$$

Taking the square root, we have

$$\mathbb{E}_\epsilon \left[\|\hat{\xi} - \xi^*\|_2 \right] \leq \frac{s \sqrt{p}}{\sqrt{cn}}. \quad (3.33)$$

The error is $\mathcal{E}(z, t) = z(t) - \hat{z}(t)$. Taking the derivative with respect to t , we get

$$\dot{\mathcal{E}}(t) = \dot{z}(t) - \dot{\hat{z}}(t) \quad (3.34)$$

$$= \Theta(z(t))\xi^* - \Theta(\hat{z}(t))\hat{\xi} \quad (3.35)$$

$$= \underbrace{(\Theta(z(t))\xi^* - \Theta(\hat{z}(t))\xi^*)}_{= f^*(z(t)) - f^*(\hat{z}(t))} + \underbrace{\Theta(\hat{z}(t))(\xi^* - \hat{\xi})}_{\text{parameter error}}. \quad (3.36)$$

By Assumption 3,

$$\|\dot{\mathcal{E}}(t)\|_2 \leq L\|\mathcal{E}(t)\|_2 + B\|\hat{\xi} - \xi^*\|_2. \quad (3.37)$$

Applying Grönwall's inequality yields, for all $T \in [0, T]$,

$$\|\mathcal{E}(T)\|_2 \leq e^{LT}\|\mathcal{E}(0)\|_2 + \int_0^T e^{L(T-t)}R\|\hat{\xi} - \xi^*\|_2 dt \quad (3.38)$$

$$= e^{LT}\|\mathcal{E}(0)\|_2 + \frac{R}{L}(e^{LT} - 1)\|\hat{\xi} - \xi^*\|_2 \quad (3.39)$$

where the last line absorbs $(e^{LT} - 1)/L \leq e^{LT}$ into constants.

If $\mathcal{E}(0) = 0$, then

$$\|\hat{z}(T) - z(T)\|_2 = \frac{R}{L}(e^{LT} - 1)\|\hat{\xi} - \xi^*\|_2. \quad (3.40)$$

Therefore,

$$E_\epsilon\|\hat{z}(T) - z(T)\|_2 \leq \mathcal{O}\left(e^{LT}E\|\hat{\xi} - \xi^*\|_2\right) = \mathcal{O}\left(e^{LT}s\sqrt{\frac{p}{n}}\right), \quad (3.41)$$

from Eqn. 3.33 and R is a constant that does not scale with n, p, s, L and T .

To get a high-probability bound, we wish to find a bound on $\|(\Theta(\mathbf{Z})^\top \Theta(\mathbf{Z}))^{-1} \Theta(\mathbf{Z})^\top \epsilon\|_2$. Notice from Assumption 2 that this is a sub-Gaussian random vector. Therefore, for any $\delta > 0$, with probability $1 - \delta$, we have

$$\left\| (\Theta(\mathbf{Z})^\top \Theta(\mathbf{Z}))^{-1} \Theta(\mathbf{Z})^\top \epsilon \right\|_2 \leq \sqrt{2s^2 p \lambda_{\max}((\Theta(\mathbf{Z})^\top \Theta(\mathbf{Z}))^{-1}) \log(1/\delta)} \quad (3.42)$$

From Assumption 1, the smallest eigenvalue of $\Theta(\mathbf{Z})^\top \Theta(\mathbf{Z})$ is bounded below by cn . Therefore, the largest eigenvalue of $(\Theta(\mathbf{Z})^\top \Theta(\mathbf{Z}))^{-1}$ is bounded above by $\frac{1}{cn}$.

Substituting this into the bound, we get that with probability at least $1 - \delta$,

$$\|\hat{\xi} - \xi^*\|_2 \leq \sqrt{\frac{2s^2 p}{cn} \log(1/\delta)} = s \sqrt{\frac{2p \log(1/\delta)}{cn}}.$$

Therefore, we have that with probability at least $1 - \delta$,

$$\|\hat{z}(T) - z(T)\|_2 \leq O\left(e^{LT} s \sqrt{\frac{p \log(\frac{1}{\delta})}{n}}\right). \quad (3.43)$$

□

Although the error bound grows exponentially with e^{LT} , empirical results from SINDy modeling show that the error grows almost linearly within finite-time regime. This behavior can be consistent with the exponential error bound above as L is small for a correctly specified model. The normalized input and smooth basis functions that will make $\Theta(\mathbf{z})$ and their derivatives tend to vary smoothly and remain small in magnitude within the input domain. Effectively, $\Theta(\mathbf{z})$ will have a very small local Lipschitz constant. Consequently, we frequently observe the error will behave linearly on the order of $O\left(LTs \sqrt{\frac{p}{n}}\right)$ in finite-time. However, over sufficiently long time horizons, exponential divergence will dominate and is unavoidable, particularly for chaotic systems.

3.7.3 The error of neural networks

In a feed-forward network setting, the optimal functional $\theta^*(\cdot) \in \mathcal{F}_{\text{ReLU}}$ can be parameterized from the following:

$$\theta^*(z) = \sigma_k \left(W_L^* \sigma_{k-1} \left(W_{k-1}^* \cdots \sigma_1 \left(W_1^* z \right) \right) \right). \quad (3.44)$$

This functional learning problem shrinks down to a parametric learning problem as well. The empirical risk minimizer is defined as

$$\hat{\theta} =_{\theta} \frac{1}{n} \sum_{i=1}^n \ell(f(z_i), \theta(z_i)), \quad (3.45)$$

where $\ell(\cdot)$ is the loss function, and $f(z_i)$ is the target value.

Apart from the SINDy-class, the empirical risk minimizer has to be solved via numerical optimization procedures. Additionally, as the loss landscape is non-convex, two-layer ReLU networks are prone to overfitting, and could perform poorly in extrapolation. Fig. S1 shows how the extrapolation of neural networks can be suboptimal for learning dynamical systems. In the following, we present the error analysis for neural networks in dynamical system learning.

We require the following assumptions for theoretical analysis on neural networks.

Assumption 4 (True dynamics). There exists a compact set $\mathcal{K} \subset \mathbb{R}^d$ such that both trajectories $\{z(t), \hat{z}(t)\}_{t \in [0, T]}$ remain in $\mathcal{K}$. Moreover, the true vector field $f^* : \mathcal{K} \rightarrow \mathbb{R}^d$ is L -Lipschitz on $\mathcal{K}$:

$$\|f^*(z) - f^*(z')\|_2 \leq L \|z - z'\|_2, \quad \forall z, z' \in \mathcal{K}.$$

Finally, the state is uniformly bounded on $\mathcal{K}$:

$$R := \sup_{z \in \mathcal{K}} \|z\|_2 < \infty.$$

Assumption 5 (Neural network model class). Consider the class $\mathcal{F}_{\text{ReLU}}$ of k -layer ReLU networks $f_{\theta}(\cdot)$ with parameters $\theta = (W_1, \dots, W_k)$. Assume the weights are uniformly

bounded:

$$B := \sup_{f_\theta \in \mathcal{F}_{\text{ReLU}}} \max_{\ell \in \{1, \dots, k\}} \|W_\ell\|_F < \infty.$$

Moreover, the induced one-step predictor is stable on $\mathcal{K}$ in the sense that

$$f_\theta(\mathcal{K}) \subseteq \mathcal{K}, \quad \forall f_\theta \in \mathcal{F}_{\text{ReLU}}.$$

Assumption 6 (Bounded loss). The training data consist of i.i.d. pairs $\{(z_0^{(i)}, z_H^{(i)})\}_{i=1}^n$ supported on $\mathcal{K} \times \mathcal{K}$, where $H = T/\Delta t$. Define the H -step rollout predictor

$$f_\theta^{(H)} := \underbrace{f_\theta \circ \dots \circ f_\theta}_{H \text{ times}},$$

and the normalized squared loss

$$\ell(f_\theta^{(H)}(z_0), z_H) := \frac{1}{4R^2} \|f_\theta^{(H)}(z_0) - z_H\|_2^2.$$

Under Assumption 4, we have $\ell \in [0, 1]$.

Assumption 4 is similar to Theorem 1 which describes the statistical model. Assumption 5 is a technical assumption that ensures the neural network weights are bounded to avoid unexpected extrapolation. Assumption 6 is used to simplify the constant bound for Big-O analysis.

Theorem 4. Suppose we have a neural network functional with k layers of ReLU activation functions and parameters $\theta = (\mathbf{W}_1, \dots, \mathbf{W}_k)$, which computes functions

$$f(\mathbf{z}; \theta) = \sigma_k(\mathbf{W}_k \sigma_{k-1}(\mathbf{W}_{k-1} \dots \sigma_1(\mathbf{W}_1 \mathbf{z}))). \quad (3.46)$$

Suppose we have Assumption 4-7. Let $\hat{f}$ denote the empirical risk minimizer that achieves zero empirical loss and define the H -step rollout $\hat{z}(T) = \hat{f}^{(H)}(z_0)$ and $z(T) = z_H$. Then, with probability at

least $1 - \delta$, the prediction error after $H = T/\Delta t$ steps satisfies

$$\mathbb{E} \left[\|\hat{z}(T) - z(T)\|_2^2 \right] \leq O \left((\log n)^4 B^{k(H+1)} \sqrt{\frac{k}{n}} + \frac{\log(1/\delta)}{n} \right). \quad (3.47)$$

Proof. From the normalized squared loss from Assumption 7:

$$\ell(f^{(H)}(z_0), z_H) := \frac{1}{4R^2} \|f^{(H)}(z_0) - z_H\|_2^2 \in [0, 1]. \quad (3.48)$$

Let $L(f)$ and $\hat{L}(f)$ denote the population and empirical risks for this loss:

$$L(f) := \mathbb{E} \ell(f^{(H)}(Z_0), Z_H), \quad \hat{L}(f) := \frac{1}{n} \sum_{i=1}^n \ell(f^{(H)}(z_0^{(i)}), z_H^{(i)}). \quad (3.49)$$

From Lemma 2, with probability at least $1 - \delta$, we have

$$L(f) \leq (1 + c)\hat{L}(f) + c(\log n)^4 \mathcal{R}_n(\mathcal{F}_{\text{ReLU}}^{(H)}) + \frac{c \log(1/\delta)}{n}, \quad (3.50)$$

where $c > 0$ is a universal constant.

We aim to control the generalization error after $H = T/\Delta t$ steps. From Lemma 5, the Rademacher complexity after H compositions is bounded by

$$\mathcal{R}_n(\mathcal{F}_{\text{ReLU}} \circ \dots \circ \mathcal{F}_{\text{ReLU}}) \lesssim B^{kH} \frac{\sqrt{k} B^k}{\sqrt{n}} = B^{k(H+1)} \sqrt{\frac{k}{n}}. \quad (3.51)$$

Substituting this bound into Lemma 2 yields, with probability at least $1 - \delta$,

$$L(f) \leq (1 + c)\hat{L}(f) + c(\log n)^4 B^{k(H+1)} \sqrt{\frac{k}{n}} + \frac{c \log(1/\delta)}{n}. \quad (3.52)$$

Finally, by the definition of ℓ we have

$$L(f) = \mathbb{E} \left[\frac{1}{4R^2} \|f^{(H)}(Z_0) - Z_H\|_2^2 \right], \quad (3.53)$$

and therefore, multiplying both sides by $4R^2$ (and absorbing constants into $O(\cdot)$), we obtain

the desired bound

$$\mathbb{E} [\hat{z}(T) - z(T)]_2^2 \leq O \left((\log n)^4 B^{k(H+1)} \sqrt{\frac{k}{n}} + \frac{\log(1/\delta)}{n} \right). \quad (3.54)$$

□

Technical Lemmas for the Proof of Thm. 4

We introduce the following lemma from [16] and rewrite it within our context.

Lemma 1 (Thm. 2.11 in [16]). *Let $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ be the ReLU activation function. Consider a feed-forward network with k layers of these nonlinearities and parameters $\theta = (\mathbf{W}_1, \dots, \mathbf{W}_L)$, which computes functions*

$$f(\mathbf{z}; \theta) = \sigma_k(\mathbf{W}_k \sigma_{k-1}(\mathbf{W}_{k-1} \cdots \sigma_1(\mathbf{W}_1 \mathbf{z}))). \quad (3.55)$$

Define the class of functions on the unit Euclidean ball in $\mathbb{R}^p$,

$$\mathcal{F}_B = \{f(\cdot; \theta) : \|\mathbf{W}_i\|_F \leq B\}, \quad (3.56)$$

where $\|\mathbf{W}_i\|_F$ is the Frobenius norm of $\mathbf{W}_i$. Then, we have the Rademacher complexity of $\mathcal{F}$ is bounded by

$$\mathcal{R}_n(\mathcal{F}_B) \lesssim \frac{\sqrt{k} B^k}{\sqrt{n}}. \quad (3.57)$$

This result is from [112] and [16]. We then introduce the following connection between the Rademacher complexity and the generalization error.

Lemma 2 (Thm. 2.2 in [16]). *Let $\mathcal{F}_B$ be a class of neural networks with weights bounded by B , and let $f \in \mathcal{F}_B$. For the mean-squared loss that $\ell(\hat{y}, y) = (\hat{y} - y)^2$, and for any distribution on $X \times [-1, 1]$ where the output is bounded. With probability at least $1 - \delta$, there exists a constant $c > 0$ such that*

$\forall f \in \mathcal{F}_B$:

$$L(f) \leq (1 + c)\hat{L}(f) + c(\log n)^4 \mathcal{R}_n(\mathcal{F}_B) + \frac{c \log(1/\delta)}{n}, \quad (3.58)$$

where $L(f)$ is the true risk, $\hat{L}(f)$ is the empirical risk, n is the number of data points, and B is a bound on the weights of the neural network.

An important observation here is that, even with a fixed set of parameters, for autoregressive neural networks, the rademacher complexity grows massively with longer time horizons. In other words, if we want to predict T steps ahead, the network size will grow exponentially with T . We use the following lemma called the contraction principle to analyze the expected network growth in this scenario.

Lemma 3. *Let $\mathcal{F}_1$ be a class of functions from $\mathbb{R}^p$ to $\mathbb{R}^q$ with Lipschitz constant α . Let $\mathcal{F}_2$ be a class of functions from $\mathbb{R}^q$ to $\mathbb{R}^r$. Then, the contraction principle describes that the Rademacher complexity of $\mathcal{F}_1 \circ \mathcal{F}_2$ is*

$$R_n(\mathcal{F}_1 \circ \mathcal{F}_2) \leq \alpha \mathcal{R}_n(\mathcal{F}_2). \quad (3.59)$$

The proof of this lemma can be found in [15, 165]. From this Lemma, we further control the Lipschitz constant for neural network in the following:

Lemma 4 (Proposition 2 in [276]). *Suppose we have a neural network functional with k layers of ReLU activation functions and parameters $\theta = (\mathbf{W}_1, \dots, \mathbf{W}_L)$, which computes functions*

$$f(\mathbf{z}; \theta) = \sigma_k(\mathbf{W}_k \sigma_{k-1}(\mathbf{W}_{k-1} \cdots \sigma_1(\mathbf{W}_1 \mathbf{z}))). \quad (3.60)$$

The Lipschitz constant of the network is bounded by

$$\prod_{i=1}^k \|\mathbf{W}_i\|_2 \leq B^k, \quad (3.61)$$

where $\|\mathbf{W}_i\|_2$ is the spectral norm of $\mathbf{W}_i$.

Proof. Since the ReLU activation function has a Lipschitz constant of 1, the Lipschitz constant of the i -th layer is bounded by $\|\mathbf{W}_i\|_2$. Therefore, from Lemma 3, for a k -layer neural network with weights $\theta = (\mathbf{W}_1, \dots, \mathbf{W}_k)$, the Lipschitz constant of the entire network is bounded by the product of the spectral norms of the weight matrices:

$$\text{Lip}(f(z; \theta)) \leq \prod_{i=1}^k \|\mathbf{W}_i\|_2. \quad (3.62)$$

Given that $\|\mathbf{W}_i\|_F \leq B$ for all i , and $\|\mathbf{W}_i\|_2 \leq \|\mathbf{W}_i\|_F$, we have

$$\prod_{i=1}^k \|\mathbf{W}_i\|_2 \leq \prod_{i=1}^k \|\mathbf{W}_i\|_F \leq B^k. \quad (3.63)$$

This means that the Lipschitz constant of the neural network is bounded by B^k . $\square$

From the Lemma above, we could obtain the Rademacher complexity of neural network prediction up to $H = \frac{T}{\Delta t}$ steps forward.

Lemma 5. *The Rademacher complexity of the neural network functional class after H forward unroll operations for $\mathcal{F}_{\text{ReLU}}$ is bounded by*

$$\mathcal{R}_n(\mathcal{F}_{\text{ReLU}} \circ \dots \circ \mathcal{F}_{\text{ReLU}}) \lesssim B^{\frac{kT}{\Delta t}} \frac{\sqrt{k} B^k}{\sqrt{n}}. \quad (3.64)$$

Proof. From Lemma 4, we know that the Lipschitz constant of the neural network is bounded by B^k . Therefore, the Rademacher complexity of the neural network functional class is bounded by

$$\mathcal{R}_n(\mathcal{F}_{\text{ReLU}}) \lesssim \frac{\sqrt{k} B^k}{\sqrt{n}}. \quad (3.65)$$

From Lemma 3, we know that the Lipschitz constant of the network is bounded by B^k . Therefore, the Rademacher complexity of the neural network functional class after H forward

unroll operations is bounded by

$$\mathcal{R}_n(\mathcal{F}_{\text{ReLU}} \circ \cdots \circ \mathcal{F}_{\text{ReLU}}) \lesssim B^{\frac{kT}{\Delta t}} \frac{\sqrt{k}B^k}{\sqrt{n}}. \quad (3.66)$$

□

3.8 Observation on the convex landscape

We demonstrate our observations on the convexity of the loss landscape of SHRED. To visualize the loss landscape in a high-dimensional space, we utilize a popular method [168] which represents the landscape in the following way. Suppose the neural network is parameterized by θ , which includes the weights and biases of all layers. We perturb θ_0 with two random directions $\mathbf{r}_x, \mathbf{r}_y$ via

$$\theta' = \theta_0 + t \cdot \alpha \mathbf{r}_x + t \cdot \alpha \mathbf{r}_y, \quad (3.67)$$

where $\mathbf{r}_x, \mathbf{r}_y$ are both normalized i.i.d. Gaussian samples. α is the scale of changes and $t \in [0, 1]$ moves from θ_0 towards the linear combination of the two random directions.

In Fig. S2, we visualize the loss landscape of SHRED from different random directions using the methodology described above under various network depths and scales α . Across all subplots, we consistently observe a clear global minimum with a convex landscape. Furthermore, Fig. S3 presents an empirical convexity analysis, showing loss changes along several paths from different initializations. We evaluate convexity using randomly sampled points and find that all of them satisfy the convexity condition. We use a tolerance of 10^{-7} to counter floating-point imprecision.

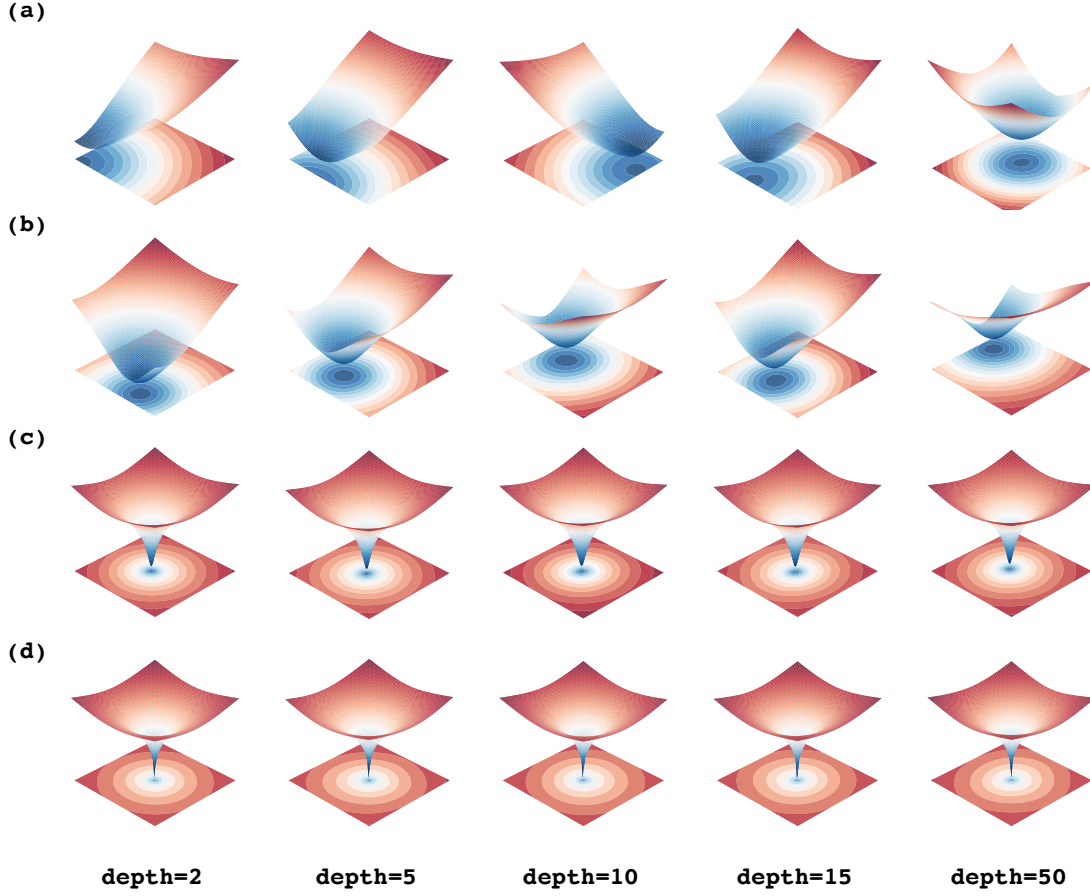


Figure S2: (a) Visualization of loss landscape from parameters ranging from $[-5, 5]$. (b) Visualization of loss landscape from parameters ranging from $[-10, 10]$. (c) Visualization of loss landscape from parameters ranging from $[-100, 100]$. (d) Visualization of loss landscape from parameters ranging from $[-1000, 1000]$.

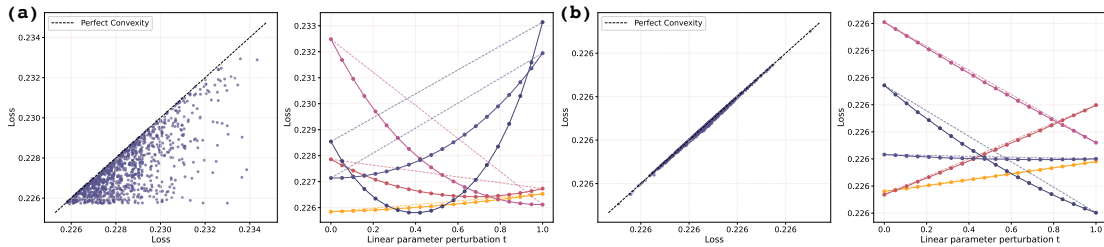


Figure S3: Visualization of loss landscapes for SHRED parameters along random directions, showing how loss values change when moving from the origin toward randomly generated points in parameter space. The figure provides a detailed view of loss behavior at different initial points. (a) shows the parameter range $[-100, 100]$ and (b) shows the parameter range $[-1, 1]$.

3.9 Ablation study

3.9.1 Effect on the recurrent units

In this study, we investigate the effect of different recurrent unit choices on the ability of SINDy-SHRED to learn parsimonious latent dynamics. Although in theory many recurrent architectures could be effective, subtle differences in their modeling strategies could induce different latent spaces.

Recurrent neural network (RNN). The Elman network [84] (often referred to generically as RNN) has the following mathematical formulation,

$$z_t = \tanh(x_t W_{iz}^\top + b_{iz} + z_{t-1} W_{zz}^\top + b_{zz}), \quad (3.68)$$

where z_t is the hidden state at time t , x_t is the input at time t , and z_{t-1} is the hidden state at time $t - 1$. The initial hidden state z_0 is set to be 0.

Long short-term memory (LSTM) network. The LSTM network is a recurrent unit that utilizes gating mechanisms to help ameliorate issues with vanishing and exploding gradients that often arise in the training of RNNs. The mathematical formulation of the LSTM is

$$\begin{aligned} i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{zi}z_{t-1} + b_{zi}), \\ f_t &= \sigma(W_{if}x_t + b_{if} + W_{zf}z_{t-1} + b_{zf}), \\ o_t &= \sigma(W_{io}x_t + b_{io} + W_{zo}z_{t-1} + b_{zo}), \\ g_t &= \tanh(W_{ig}x_t + b_{ig} + W_{zg}z_{t-1} + b_{hg}), \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t, \\ z_t &= o_t \odot \tanh(c_t), \end{aligned} \quad (3.69)$$

where i_t, f_t, o_t are the input, forget, and output gates respectively. Here, z_t is the hidden state and c_t is the cell state at time t . Both z_0 and c_0 are set to be 0.

Gated recurrent unit (GRU). The GRU network simplifies LSTM network with the following mathematical formulation:

$$\begin{aligned}
 r_t &= \sigma(W_{ir}x_t + b_{ir} + W_{zr}z_{t-1} + b_{zr}), \\
 h_t &= \sigma(W_{ih}x_t + b_{ih} + W_{zh}z_{t-1} + b_{zh}), \\
 n_t &= \tanh(W_{in}x_t + b_{in} + r_t \odot (W_{zn}z_{t-1} + b_{zn})), \\
 z_t &= (1 - h_t) \odot n_t + h_t \odot z_{t-1},
 \end{aligned} \tag{3.70}$$

where r_t is the reset gate and h_t is the update gate. Here, z_t is the hidden state at time t , and z_{t-1} represents the hidden state at time $t - 1$. The initial hidden state z_0 is set to be 0.

Empirically, we are interested in choosing the network architecture that permits the smoothest representation of latent trajectories. Here, we define smoothness as latent trajectories that have small state to state fluctuations. Among these three architectures, the RNN network tends to have more spikes, as the change from z_{t-1} to z_t is directly tied to a nonlinear $\tanh(\cdot)$ function. Both LSTM and GRU networks introduce extra gates, which ensure that the hidden states will evolve more smoothly. The GRU network is preferred because only a single latent state is propagated, in contrast to the LSTM which propagates a hidden state and a cell state.

In Fig. S4, we visualize the difference between the three recurrent network architectures using the sea-surface temperature dataset. We train all three networks with the same optimizer parameters and latent state dimension and plot their latent space trajectories on the test set. We see that visually, the GRU produces the smoothest latent trajectory. Across all of our computational examples, we found the best performance using GRUs. Nevertheless, it is possible that other architectures could prove superior for certain dynamical systems.

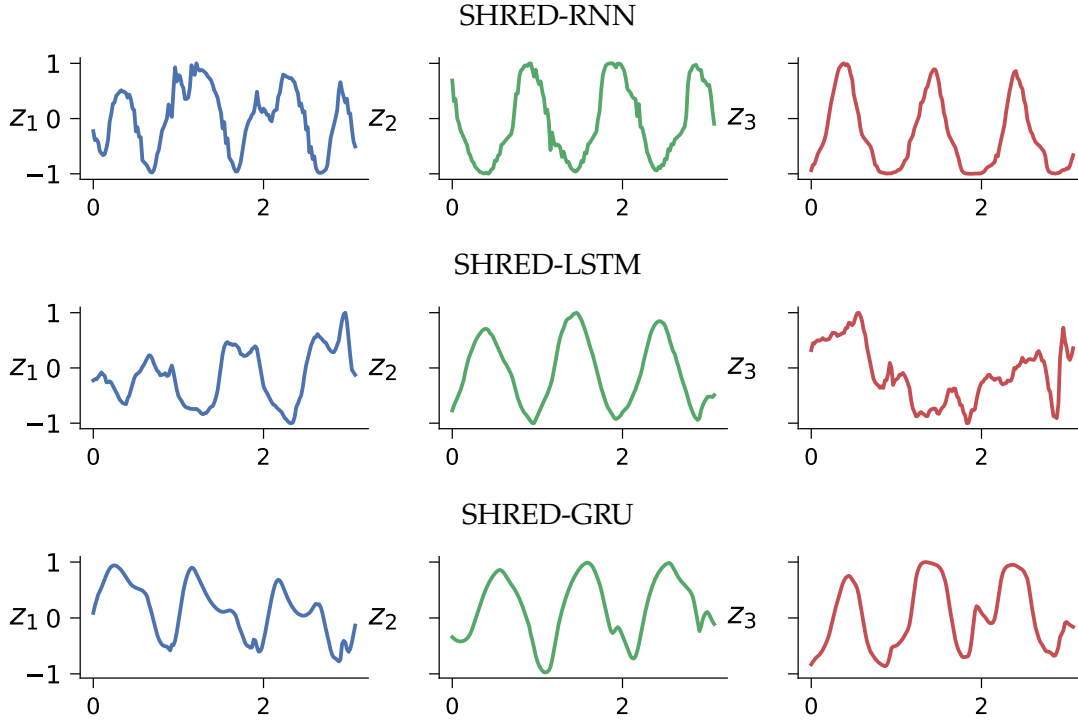


Figure S4: Comparison of latent spaces learned by SHRED (no SINDy regularization) with different recurrent units on the test set. The x-axis represents the number of years in the sea-surface temperature dataset.

3.9.2 A comparison to other baseline methods

	SINDy-SHRED	SINDy	SINDy-AE	CNN-SINDy	SHRED
Model discovery	✓	✓	✗	✗	✗
High dimensionality	✓	✗	✓	✓	✓
Interpretable	✓	✓	✓	✓	✓
Short-term prediction	✓	✗	✓	✓	✓
Long-term rollouts	✓	✗	✗	✗	✗
Reproducibility	✓	✓	✗	✓	✓

Table 3.2: Comparison of different SINDy-based and SHRED-based methods in model discovery, spatiotempora learning, and reproducibility.

In Table 3.2, SINDy-SHRED is compared to all baseline methods. SINDy algorithm [32] is not designed for high-dimensional problem, which requires a pre-specification of a good coordinate transform. Therefore, it works poorly for both short- and long-term predictions for high-dimensional datasets. With autoencoders, SINDy-autoencoder [46] can work on high-dimensional data. However, because autoencoders impose no constraints on the derived

latent space, SINDy-AE generally struggles to find a stable physical system in the latent space that does not diverge during rollout. CNN-SINDy [96] does not jointly solve SINDy and the latent dynamics, which typically results in an unnecessarily complex dynamical system. SHRED [292] is primarily designed as a sensing method, which does not have built-in capability for model discovery and long-term predictions.

3.9.3 Quantitative comparison by training SHRED with and without SINDy loss terms

In what follows, we present a comparison study between SHRED and SINDy-SHRED.

Equation discovery. We visualize the latent spaces of SHRED and SINDy-SHRED in Fig. S5 for the sea-surface temperature dataset. Starting from the same weight initialization, we observe that the derived latent spaces of SHRED and SINDy-SHRED are visually different. For every 50 epochs, we plot the induced latent space by the two models from epoch 50 (light blue) to epoch 1000 (dark green). The latent space of SINDy-SHRED converges to a smoother dynamical system without frequent spiking behavior, demonstrating that SINDy-SHRED does act as a regularizer on the dynamics of the latent space.

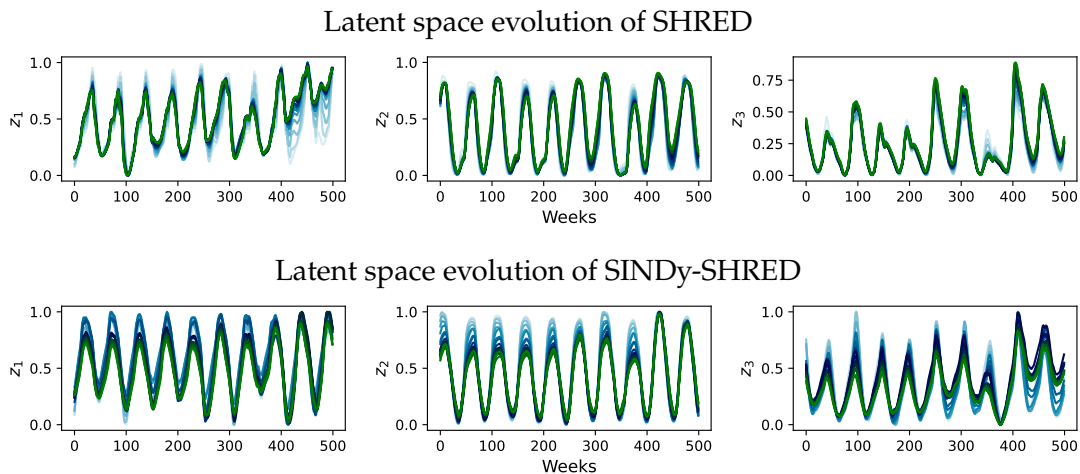


Figure S5: Comparison of latent spaces learned by SHRED (top row) and SINDy-SHRED (top row). The color represents the induced latent space evolution from early time (blue) to the end of training (green).

Additionally, the strength of regularization plays an important role in equation discovery, which is shown in Fig. S6. The latent space trajectories are gradually regularized into an ODE model with an explicit formulation.

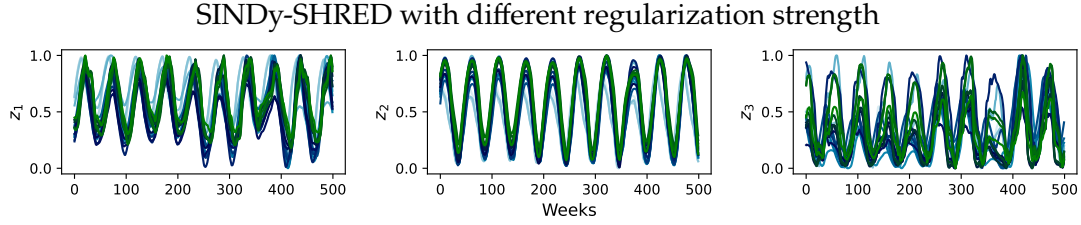


Figure S6: Top row: the latent space evolution with increasingly stronger SINDy regularization parameter, ranging from 0.1 to 10.

In Fig. S7, we visualize the latent space of a SHRED model without SINDy regularization. The result is latent dynamics which fail to be well-described by SINDy model.

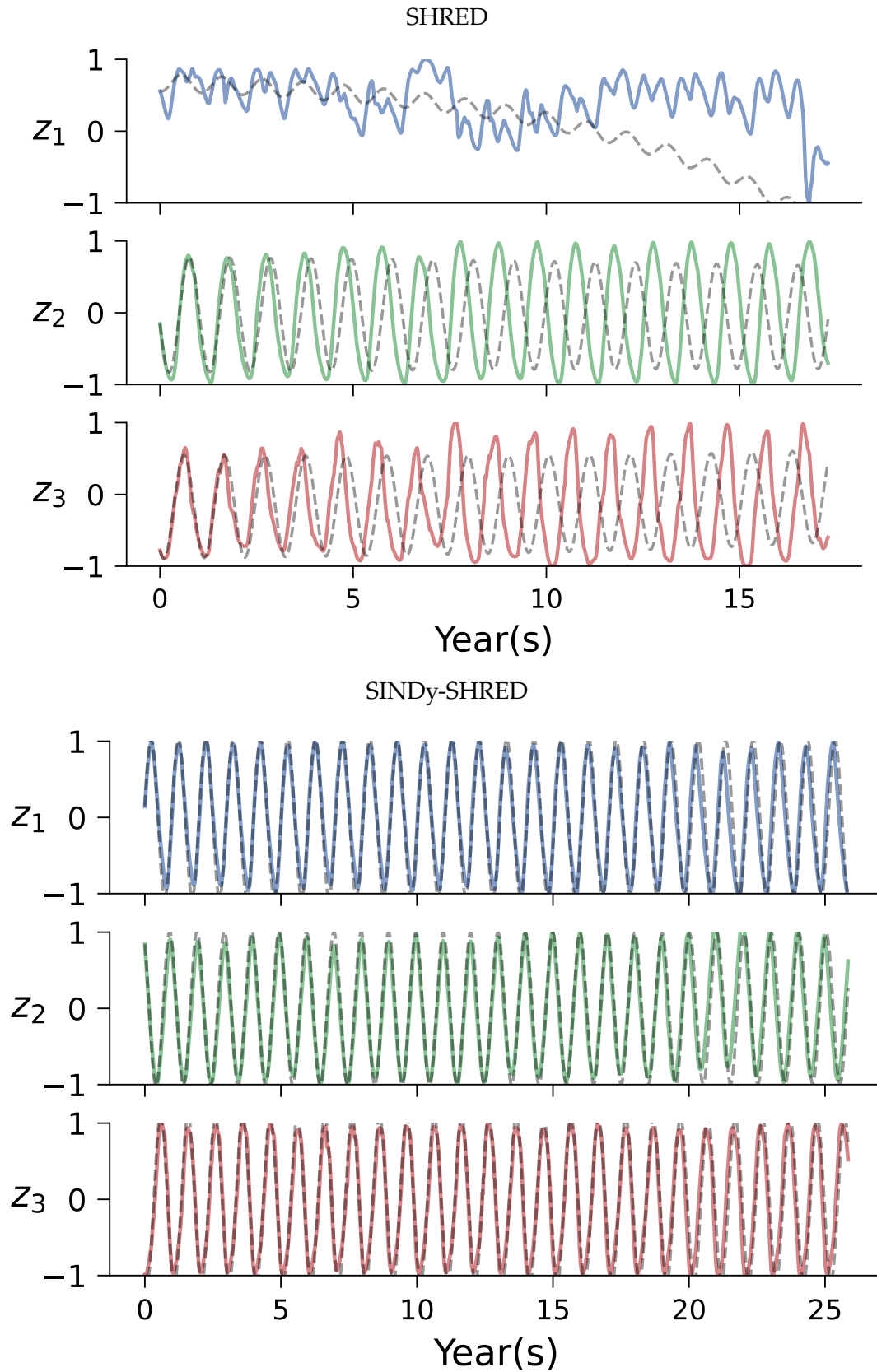


Figure S7: Comparison of the latent space discovery from SHRED and SINDy-SHRED. Without SINDy regularization, SHRED has no guarantee to compress the raw observation into a physically interpretable latent space. SINDy-SHRED finds an accurate ODE model that can be used to achieve long term forecasts.

Predictive power. An additional benefit of the SINDy-SHRED method is increased forecasting performance in comparison to SHRED. To enable standard SHRED to perform spatio-temporal forecasting, one strategy mentioned in [292] is to first train a separate GRU network to predict the sparse input sensor measurements. Then, from this trained predictor for the sparse sensors, one can autoregressively forecast the sparse sensor sequence from available data (e.g. in Fig. S8). By treating the predicted sparse sensors as inputs, standard SHRED can produce a prediction of the spatial field.

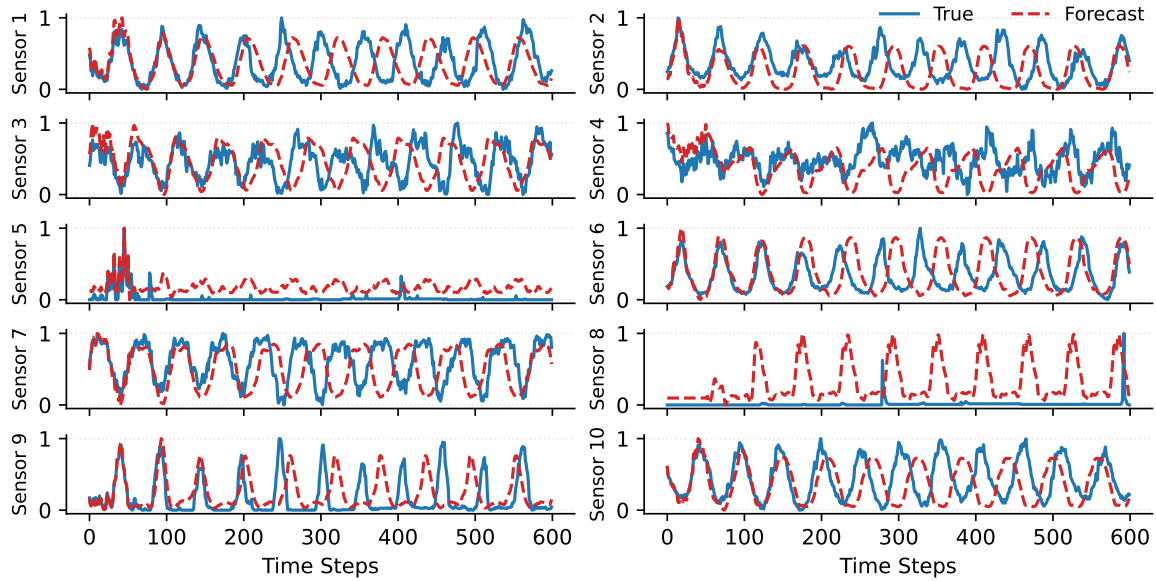


Figure S8: Comparison of long-term prediction between SHRED and SINDy-SHRED. The frame error is evaluated by each frame (left) and the cumulative error is calculated by summing up all previous frame errors (right).

As shown in Fig. S9, we find SINDy-SHRED greatly outperforms standard SHRED in such predictions. This behavior is in agreement with that in Fig. S8, which shows that the long-term rollout prediction from GRU network tends to be inaccurate. Utilizing an explicit ODE model via SINDy-SHRED can stabilize its long-term behavior and reduce the prediction error in general.

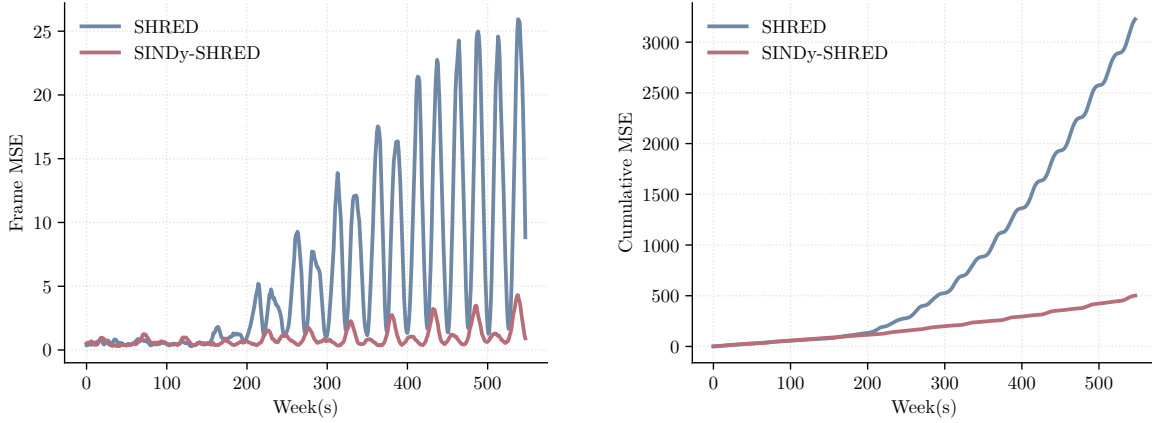


Figure S9: Comparison of long-term prediction between SHRED and SINDy-SHRED. The frame error is evaluated by each frame (left) and the cumulative error is calculated by summing up all previous frame errors (right).

3.9.4 Latent space discoveries from different sensor placements are equivalent under linear transformations

Different sensor placements can lead SINDy-SHRED to recover the same underlying physical dynamics, but expressed in distinct latent coordinate systems related by a linear transformation. This occurs because the governing dynamics are invariant under linear transformations such as rotations and scalings, so the same dynamics may appear with different symbolic representations in different coordinate frames. In practice, varying the sensor placement introduces an implicit prior that biases SINDy-SHRED toward learning the system from a particular geometric perspective.

To illustrate this equivalence, we use sea-surface temperature data as an example. We train SINDy-SHRED with a latent dimension of two (chosen for ease of visualization) while keeping all other training settings fixed. We then compare the raw latent spaces identified from two different sensor configurations to examine how sensor placement affects the learned coordinate representation. As shown in Fig. S10, the raw latent spaces obtained from the two sensor configurations appear different but are equivalent under a linear transformation. We identify this transformation using linear regression to map the latent variables from the first run (latent 1) onto those from the second (latent 2). This result aligns with previous findings

in autoencoder-based SINDy [46] (see Supplemental Figure S1), where models trained under different conditions were likewise equivalent up to an affine transformation.

Both discoveries provide equally valid representations of the underlying dynamics. In principle, one could identify the most parsimonious representative within the equivalence class defined by the full group of linear transformations, but this procedure is computationally expensive [210].



Figure S10: Latent space discovery from the sea-surface temperature data under two different sensor placements. The raw discoveries appear to be different but the two discoveries are close under a linear transformation.

3.9.5 Hyperparameter robustness

The following study supports the empirical observation that the SINDy-SHRED architecture is robust to hyperparameter choices (e.g. learning rate and batch sizes). As shown in Fig. S11, we retrain SINDy-SHRED on the sea-surface temperature dataset using the same set of sensors with different hyperparameter settings. We keep all other settings to be the same in the main

SST experiment. Across these ablation trails, we observe that all configurations converge at similar rates and achieve similar final errors.

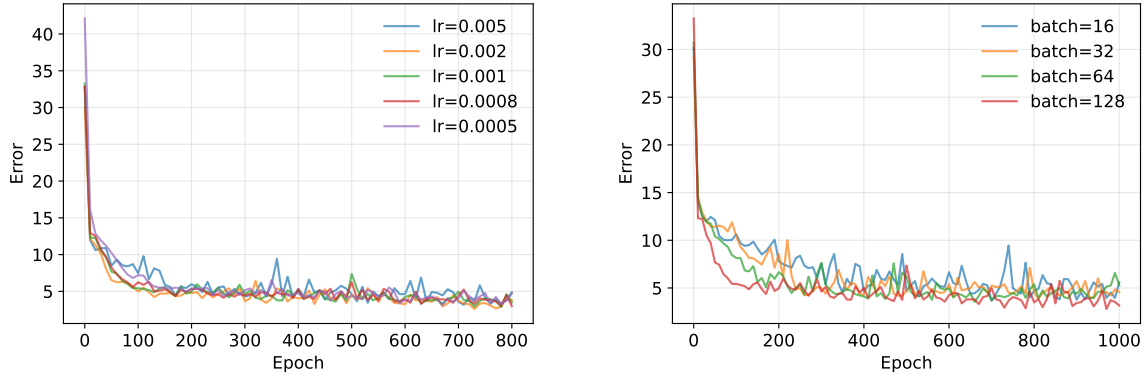


Figure S11: Error decay with training epochs using different hyperparameter settings: learning rates (left) batch sizes (right). All configurations converge to a similar level of training error.

3.9.6 Energy spectrum analysis on the isotropic turbulent flow

Fig. S12 provides a detailed evaluation of the physical quantities predicted by SINDy-SHRED. The energy spectrum is computed on available ground-truth 2D pressure field snapshots and SINDy-SHRED predictions using 2D FFT, summarizing the distribution of energy across spatial scales. SINDy-SHRED effectively preserves the spectral characteristics of the real data. The probability distribution histogram further visualizes the empirical distributions of the predicted and real data. The two distributions largely overlap with each other, showing minor or no distributional shift. We see that the predicted data has a smaller tail than compared to the ground truth, indicating that the learned model smooths out high-frequency content. For reference, in 3D incompressible isotropic turbulence, the pressure spectrum is theoretically expected to scale as $k^{-7/3}$ in the inertial range [219]. The presented spectrum is computed from 2D slices which is not directly comparable to the full 3D data ($k^{-7/3}$), but we observe SINDy-SHRED a similar energy spectrum compared to the ground truth.

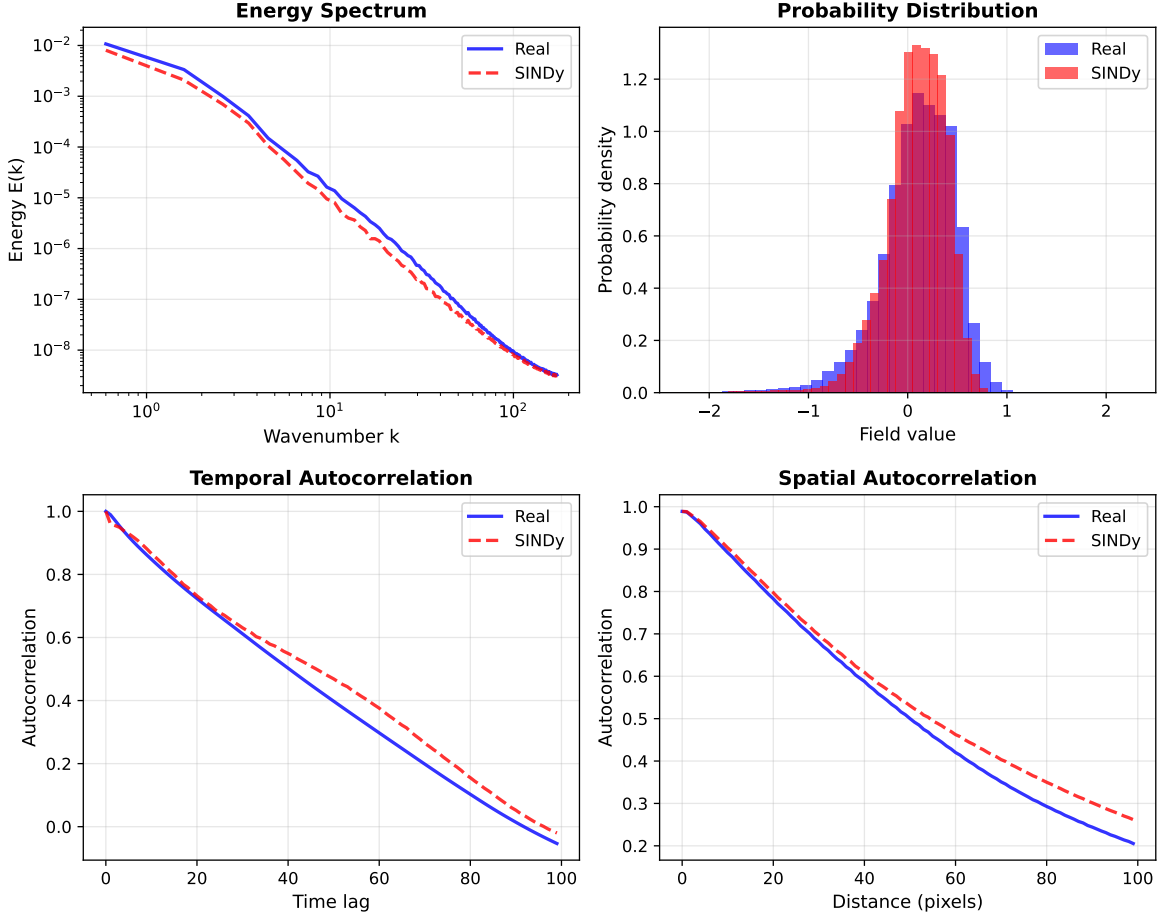


Figure S12: SINDy-SHRED prediction captures important statistical properties for long-term predictions.

3.10 Experimental details

3.10.1 The design of SINDy unit

In Fig. S13, we show the design of SINDy unit. Notice how this design is similar to the skip-connection mechanism in ResNet. Here, it models the forward simulation process of ODE integration by having $x_{t+1} = x_i + \Theta(x)\Xi\Delta t$.

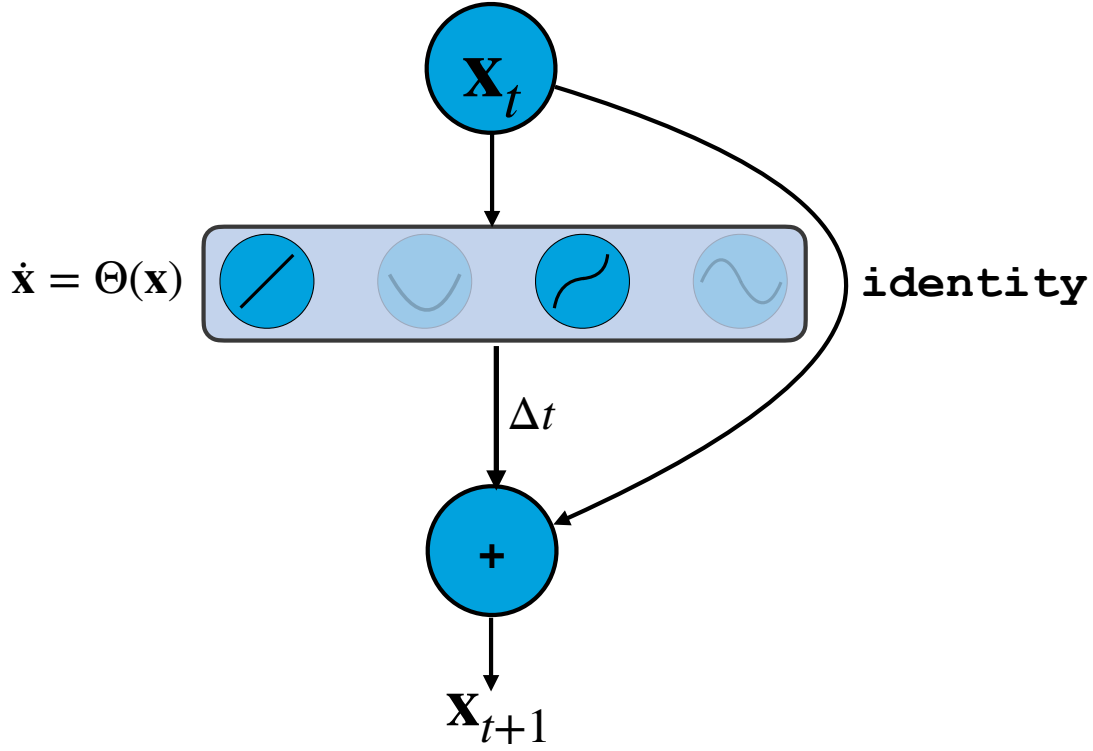


Figure S13: Diagram of the RNN form of SINDy.

3.10.2 Evaluation metrics

We define the following evaluation metrics that are commonly used to assess model's performance.

The **mean squared error** is defined as the L_2 distance between the real target data $\mathbf{X}$ to the predicted target $\hat{\mathbf{X}}$ that

$$\text{MSE} = \frac{1}{n} \sum_i (\mathbf{X}_i - \hat{\mathbf{X}}_i)^2, \quad (3.71)$$

where $\mathbf{X}$ has n samples.

The **mean absolute error** is defined as the L_1 distance between the real target data $\mathbf{X}$ to the predicted target $\hat{\mathbf{X}}$ that

$$\text{MAE} = \frac{1}{n} \sum_i |\mathbf{X}_i - \hat{\mathbf{X}}_i|, \quad (3.72)$$

where $\mathbf{X}$ has n samples.

Additionally, we define the **relative mean-squared error** for both spatiotemporal data $\mathbf{X}$ and latent space trajectory $\mathbf{Z}$ that

$$\text{rMSE}(\mathbf{X}, \hat{\mathbf{X}}) = \frac{\|\mathbf{X} - \hat{\mathbf{X}}\|_2^2}{\|\mathbf{X}\|_2^2} \quad (3.73)$$

$$\text{rMSE}(\mathbf{Z}, \hat{\mathbf{Z}}) = \frac{\|\mathbf{Z} - \hat{\mathbf{Z}}\|_2^2}{\|\mathbf{Z}\|_2^2}, \quad (3.74)$$

3.10.3 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 dataset is preprocessed from a single video recording of vortex shedding. 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 $(1e^{-4}, 1e^{-3})$. From this extreme low-data limit, we manually perform data augmentation by reusing the latter part of the video once in the training loop to increase the number of available samples.

3.10.4 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 (080×960), so we follow the same preprocessing procedure as in [191] 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 $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_L\}$ with lag L , the model autoregressively predicts the next frame $\hat{\mathbf{X}}_{L+1}$ and use it as a new input $\{\mathbf{X}_2, \mathbf{X}_3, \dots, \hat{\mathbf{X}}_{L+1}\}$. This step will be repeated L times to obtain $\{\hat{\mathbf{X}}_{L+1}, \hat{\mathbf{X}}_{L+2}, \dots, \hat{\mathbf{X}}_{2L}\}$. We treat this as the prediction and optimize the loss from this quantity. In the following baseline models, we uniformly set $L = 20$.

SINDy. We adapt the dataset and treat each the original pixel as sensors using SINDy [32]. The execution of the sparse regression program is slow on large symbolic matrices, so we only include linear libraries and constants. All tested sparse models from SINDy diverge numerically in forward simulation, so we used a model with threshold 0.0 and ℓ_2 regularization 0.05 which effectively fits the dynamical system with ridge regression solution. Given the initial sensor conditions, the algorithm unrolls the values forward with the discovered ODE.

SHRED. To enable prediction capability of SHRED, we train an additional GRU network from available sparse sensor trajectory to serve as the inputs for SHRED [292]. The vanilla SHRED network is set up to have the same latent space, decoder size, and optimizer to SINDy-SHRED.

SINDy-AE. We revise the original SINDy autoencoder framework [46] under the same variational form in SINDy-SHRED. For each input data frame, the encoder compresses the high-dimensional image into a single variable $z_{AE} \in \mathbb{R}$, and the decoder further reconstructs the original data using the latent variable z_{AE} . The SINDy loss is computed via enforcing a SINDy model on the temporal evolution of the latent variable z_{AE} . We set the learning rate, batch size and training epochs to be the same as SINDy-SHRED training.

ResNet. We use the residual neural network (ResNet) [120] 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 [105] 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 [246] 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 [287] 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.

3.10.5 Sea-surface temperature data

The SST dataset is an observational reanalysis produce provided by NOAA [225]. Every snapshot includes a 180×360 grid and the data are preprocessed through interpolation combining in-situ buoy, ship, and satellite observations. The SST dataset from NOAA is directly adapted into the training pipeline of SINDy-SHRED. We set the latent dimension to 3 because we observe only minor impacts on the reconstruction accuracy when the latent dimension is ≥ 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 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.

3.10.6 3D atmospheric ozone concentration

The ozone dataset is synthetic and derived from the GEOS-Chem chemical transport model, which numerically integrates coupled advection-diffusion-reaction equation [21]. In SINDy-SHRED training, 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 $1e^{-3}$.

3.10.7 Isotropic turbulent flow

The isotropic turbulent flow dataset is synthetic and obtained from the Johns Hopkins Turbulence Database (JHTDB) [170]. It is generated from the incompressible Navier-Stokes equations using pseudo-spectral method with periodic boundary condition. The internal DNS step is 0.0002. The generation is based upon a 1024^3 grid, and we obtain a 350×350 spatial slice of the original pressure field. In the isotropic flow experiment, due to its complex nature, we select the latent dimension to be 16 with lag 100. The forward integration time step is set to $dt = 0.0002$. We set the batch size to 128 and the learning rate to $5e^{-4}$. The thresholding procedure is executed every 100 epochs with thresholds ranging from (0.15, 1.5). The total training epoch is 300.

3.11 Experiment on the 2D Kolmogorov flow

The 2D Kolmogorov flow data is a chaotic turbulent flow generated from the pseudospectral Kolmogorov flow solver [43]. The solver numerically solves the divergence-free Navier-Stokes equation:

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 \\ \partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nu \Delta \mathbf{u} + f, \end{cases} \quad (3.75)$$

where $\mathbf{u}$ stands for the velocity field, p stands for the pressure, ν is the kinematic viscosity and f describes an external forcing term. In this case, we consider doubly periodic domain $\Omega = [0, 2\pi)^2$ with periodic boundary conditions in both spatial directions. We specify the external force to be $f(x, y) = \sin(n_f y) \mathbf{e}_1$ where $n_f = 4$ and $\mathbf{e}_1 = [1, 0]^\top$. Setting the Reynolds number to 30, which is within the weakly chaotic regime [48], and the spatial field has resolution 80×80 . 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 [57]. 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 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.

$$\left\{ \begin{array}{l} \dot{z}_1 = -0.007z_3 + 0.009z_5, \\ \dot{z}_2 = -0.207z_4, \\ \dot{z}_3 = -0.011z_1 - 0.008z_5, \\ \dot{z}_4 = 0.103z_2, \\ \dot{z}_5 = -0.012z_1 + 0.006z_3. \\ \dot{z}_6 = 0.151z_1z_2. \end{array} \right. \quad (3.76)$$

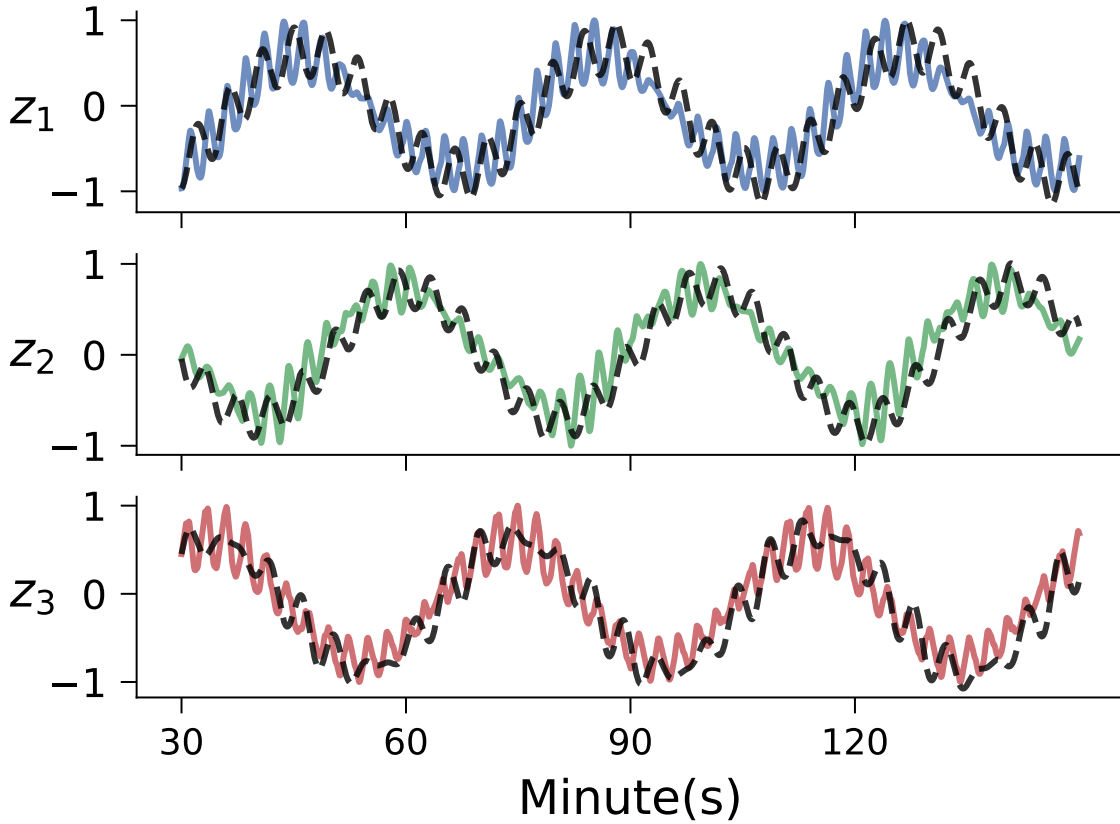


Figure S14: Extrapolation of latent representation in SINDy-SHRED from the discovered dynamical system for the 2D Kolmogorov flow data. Colored: true latent representation. Black: SINDy extrapolation.

In Eqn. 3.76, 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. S14. As we increase the Reynolds number, the discovery fails to produce robust predictions.

This representation also demonstrates nice predictions for future frames. In Fig. S15, the future prediction has an averaged MSE error of 0.035 for all available data samples. The sensor-level prediction in Fig. S28 further demonstrates the details of the reconstruction.

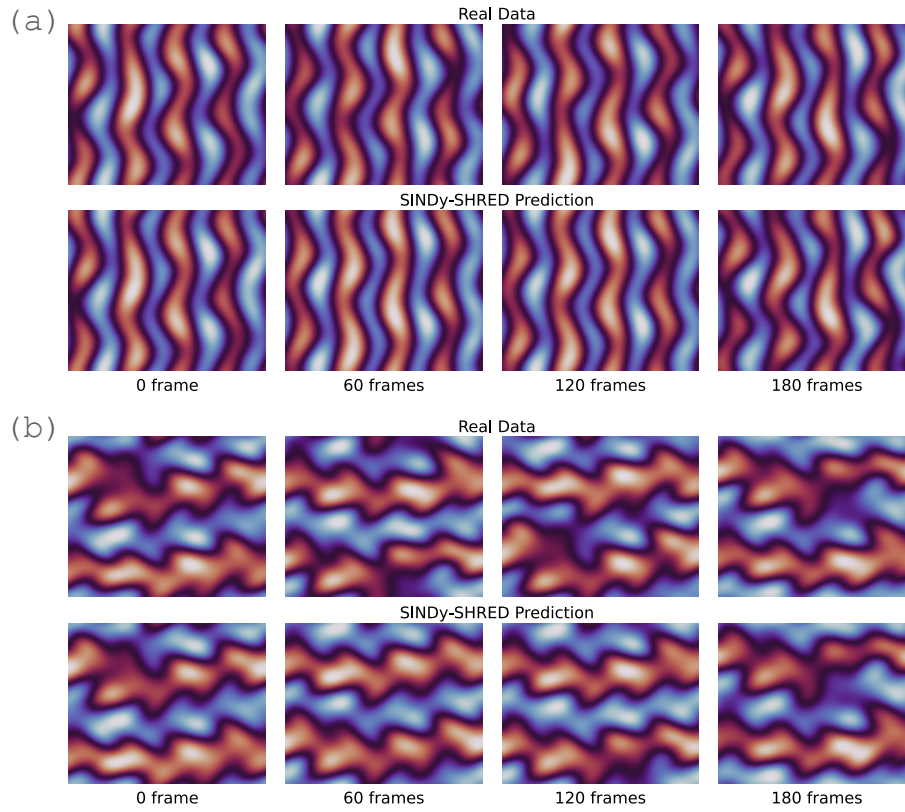


Figure S15: Long-term prediction via SINDy-SHRED for 2D Kolmogorov flow data.

3.12 Sensor level plots of experiments

3.12.1 Sea surface temperature

3D visualization of SINDy-SHRED

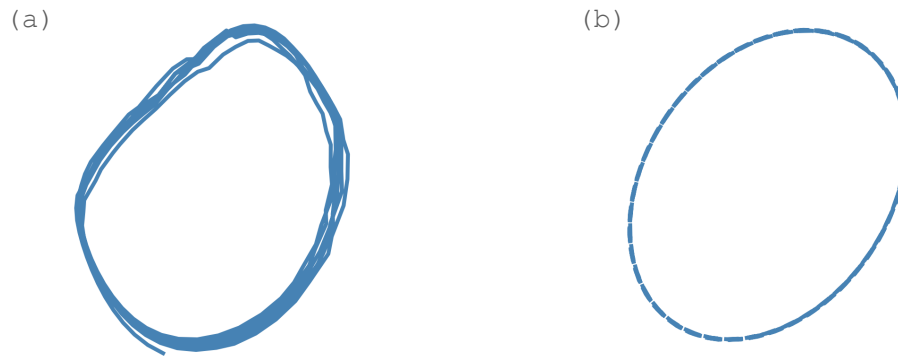


Figure S16: 3D reconstruction of the original latent space and SINDy simulated latent space.

Long-term extrapolation of SINDy-SHRED.

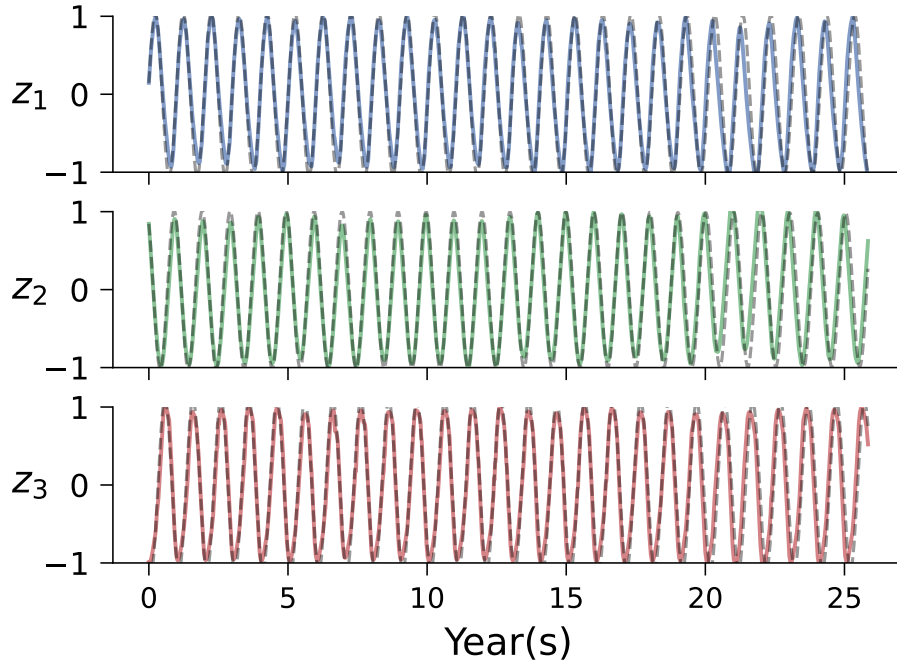


Figure S17: Extrapolation of latent representation in SINDy-SHRED from the discovered dynamical system for SST over the entire 27 years. Colored: true latent representation. Grey: SINDy extrapolation.

Sensor-level prediction on the SST dataset.

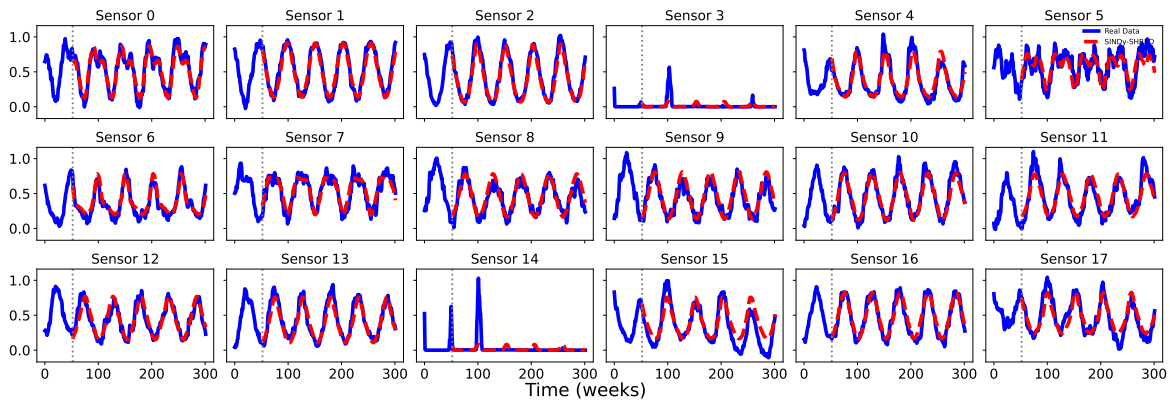


Figure S18: Extrapolation of SINDy-SHRED for sensor-level predictions on the SST data. We randomly picked 18 sensors from spatial locations that are not in the sparse sensor training. The extrapolation shows the SINDy-SHRED prediction for the following 300 weeks.

3.12.2 Ozone data

Convergence behavior of SINDy-SHRED on the Ozone dataset.

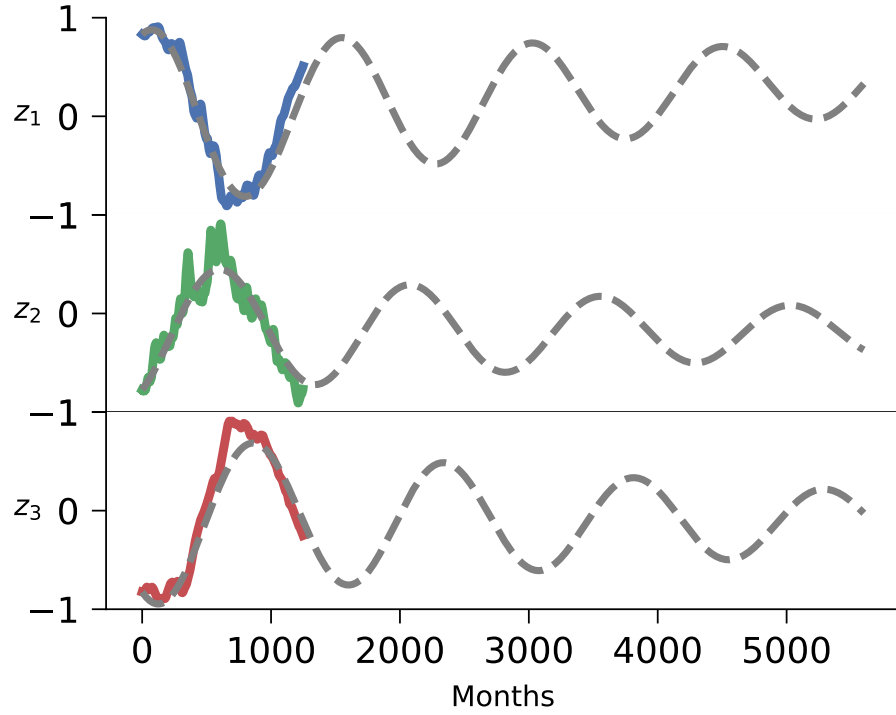


Figure S19: Long term extrapolation of Ozone data. The latent SINDy model presents a convergence behavior towards the mean-field solution.

Sensor-level prediction on the Ozone dataset.

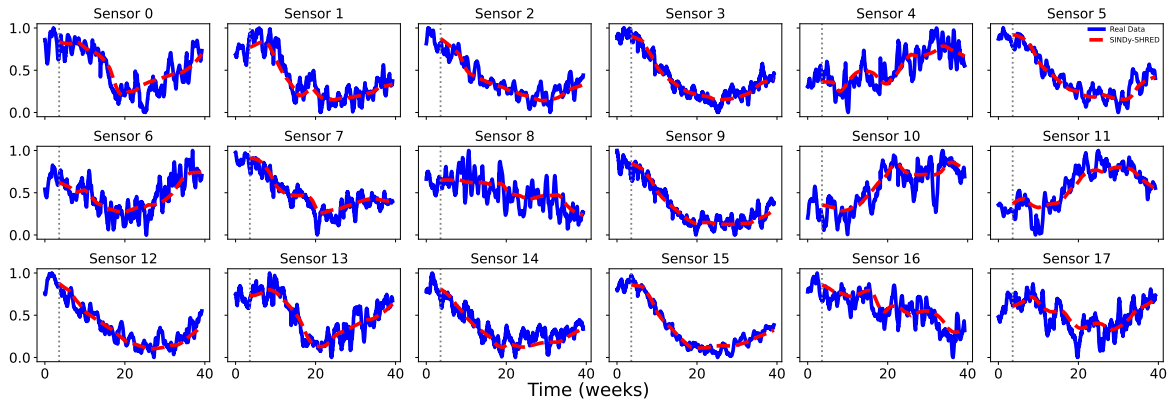


Figure S20: Extrapolation of SINDy-SHRED for sensor-level predictions on the Ozone data. We randomly picked 18 sensors from spatial locations that are not in the sparse sensor training. The extrapolation shows the SINDy-SHRED prediction for the following 40 weeks.

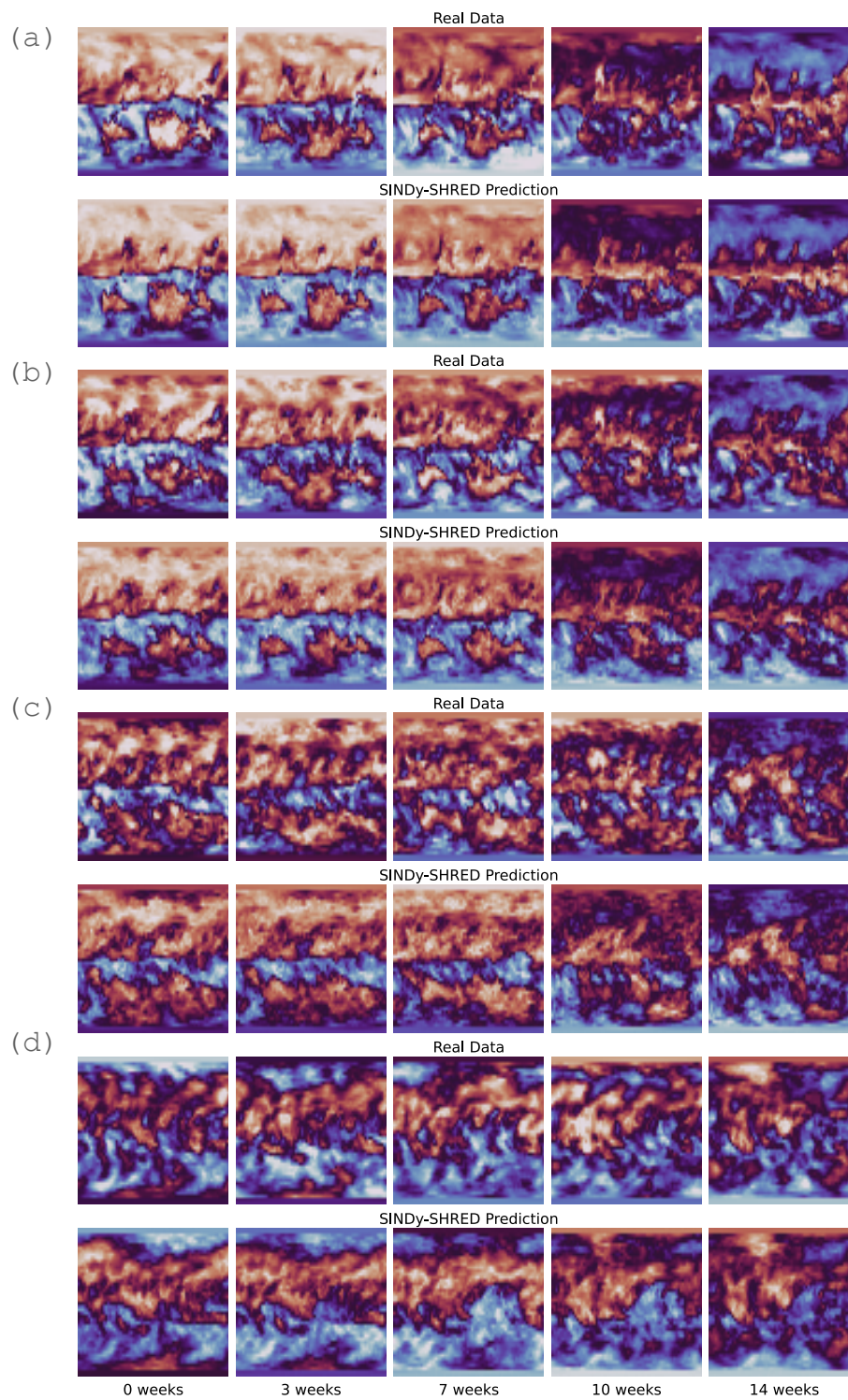


Figure S21: Reconstruction of atmospheric ozone concentration data for different elevation (a) 0 km (b) 4 km (c) 8 km (d) 12 km.

3.12.3 Flow over a cylinder

Latent space of Koopman-SHRED

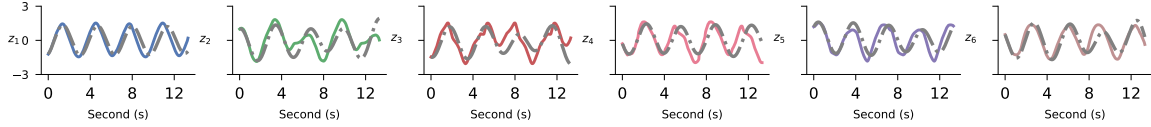


Figure S22: Extrapolation of latent representation in Koopman-SHRED from the discovered dynamical system for flow over a cylinder data. Colored: true latent representation. Grey: SINDy extrapolation.

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

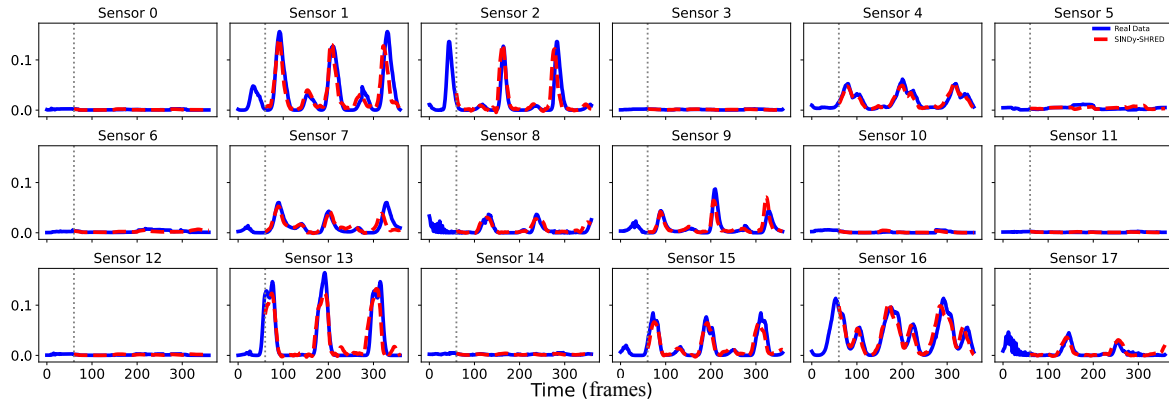


Figure S23: Extrapolation of SINDy-SHRED for sensor-level predictions on the flow over a cylinder data. We randomly picked 18 sensors from spatial locations that are not in the sparse sensor training. The extrapolation shows the SINDy-SHRED prediction for the following 400 frames.

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

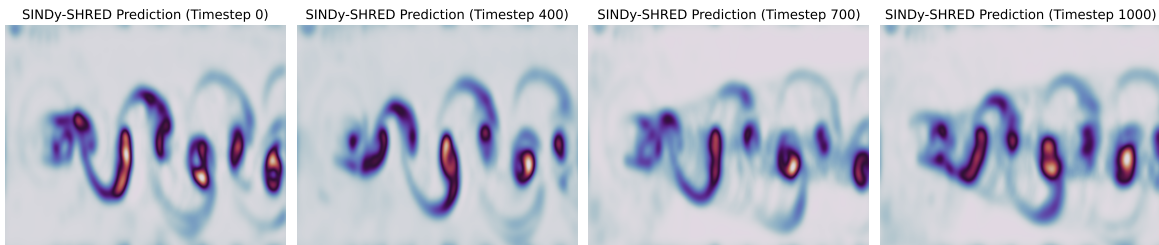


Figure S24: Prediction of the flow over a cylinder data from time step 0 (reconstruction) to 1000 frames. We note this extrapolation is completely out of the dataset. The real data for testing is only available up to 500 frames.

3.12.4 Isotropic turbulent flow

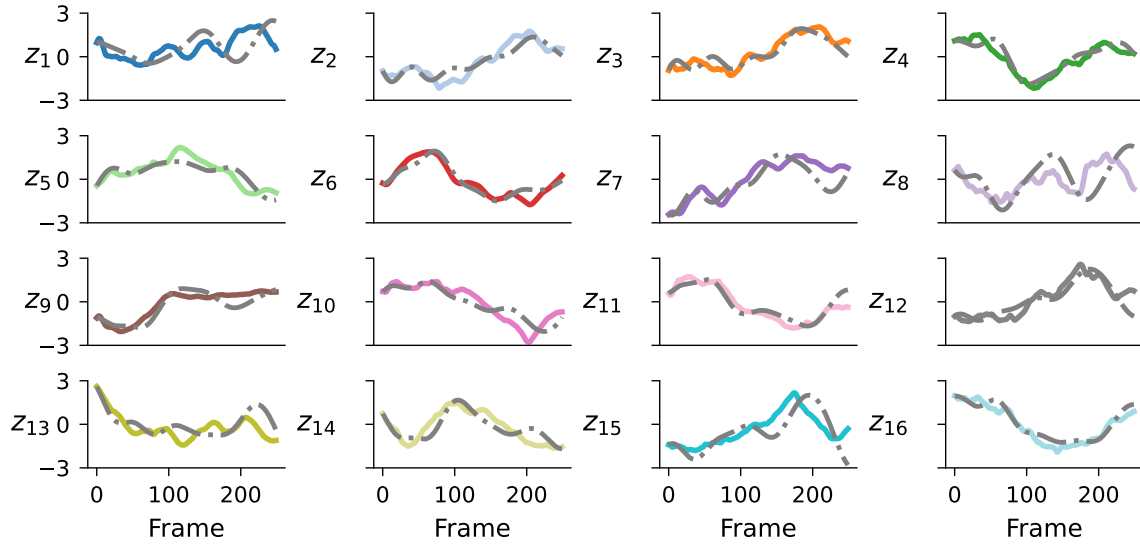


Figure S25: Extrapolation of all 16 latent representation in SINDy-SHRED from the discovered dynamical system for isotropic turbulent flow data for 250 frames. Colored: true latent representation. Grey: SINDy extrapolation.

Sensor-level prediction on the isotropic turbulent flow dataset.

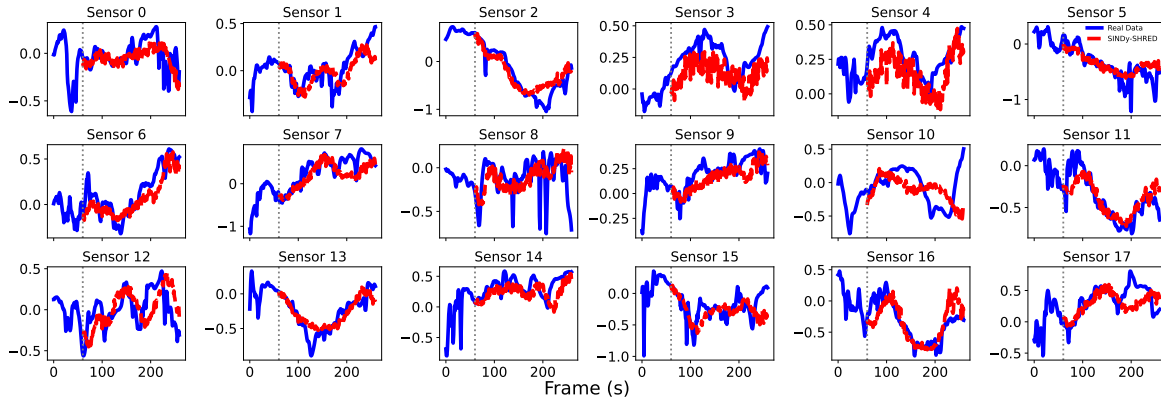


Figure S26: Extrapolation of SINDy-SHRED for sensor-level predictions on the isotropic turbulent flow data. We randomly picked 18 sensors from spatial locations that are not in the sparse sensor training. The extrapolation shows the SINDy-SHRED prediction for the following 250 frames.

3.12.5 Pendulum

Sensor-level prediction on the moving pendulum dataset.

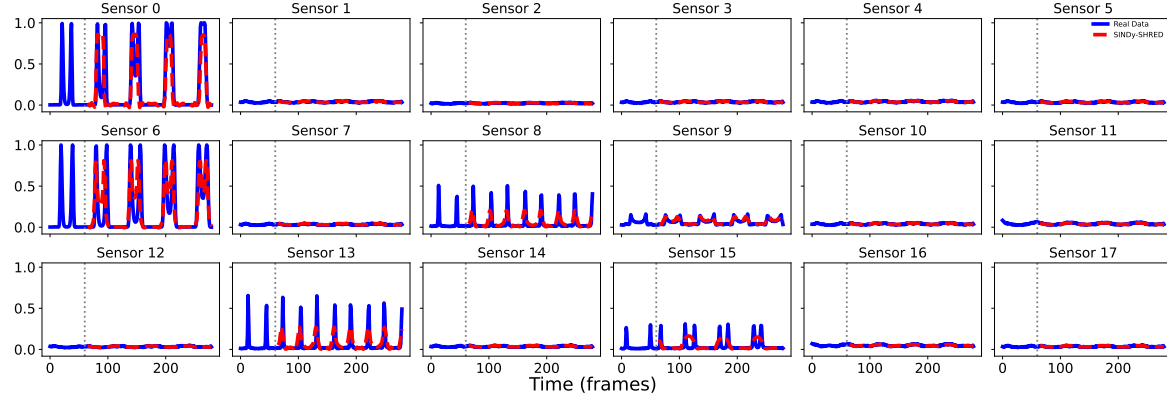


Figure S27: Extrapolation of SINDy-SHRED for sensor-level predictions on the moving pendulum data. We randomly picked 18 sensors from spatial locations that are not in the sparse sensor training. The extrapolation shows the SINDy-SHRED prediction for the following 382 frames.

3.12.6 Kolmogorov flow

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

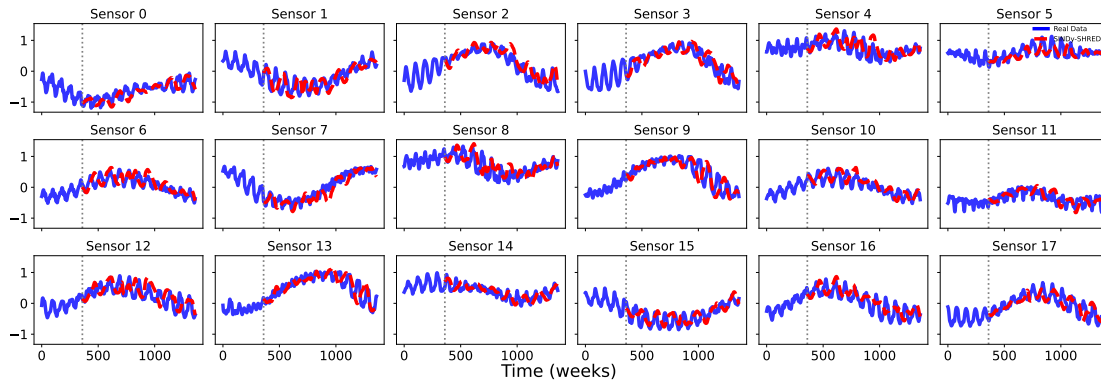


Figure S28: Extrapolation of SINDy-SHRED for sensor-level predictions on the 2D Kolmogorov flow data. We randomly picked 18 sensors from spatial locations that are not in the sparse sensor training. The extrapolation shows the SINDy-SHRED prediction for the following 1500 frames.

3.13 Analysis of learned ODEs

3.13.1 Sea-surface temperature

The analytic solution of the dynamical system discovered from the SST data

$$\mathbf{z}(t) = c_1 \mathbf{v}_1 e^{(-0.01+6.24i)t} + c_2 \mathbf{v}_2 e^{(-0.01-6.24i)t} + c_3 \mathbf{v}_3 e^{0.02t}, \quad (3.77)$$

where $v_1 = \begin{pmatrix} -0.39 + 0.38i \\ -0.19 - 0.52i \\ 0.63 \end{pmatrix}$, $v_2 = \begin{pmatrix} -0.39 - 0.38i \\ -0.19 + 0.52i \\ 0.63 \end{pmatrix}$, $v_3 = \begin{pmatrix} 0.68 \\ 0.33 \\ 0.65 \end{pmatrix}$, and c_1, c_2 , and c_3 depend on the initial condition.

3.13.2 Ozone data

The complete closed form solution to the discovered ODE from Ozone data is given by:

$$\mathbf{z}(t) = T \text{diag} \begin{pmatrix} e^{(-0.003+0.0079i)t} \\ e^{(-0.003+0.0079i)t} \\ -0.003 \end{pmatrix} T^{-1} \mathbf{z}_0 + T \left(\text{diag} \begin{pmatrix} (-42 - 111i)e^{(-0.003+0.0079i)t} \\ (-42 + 111i)e^{(-0.003-0.0079i)t} \\ -333e^{-0.003t} \end{pmatrix} \right) \Big|_0^t T^{-1} \begin{pmatrix} -0.002 \\ 0 \\ 0.002 \end{pmatrix}, \quad (3.78)$$

where $\mathbf{z}_0$ is the state at $t = 0$ and

$$T = \begin{pmatrix} 0.66 & 0.66 & -0.99 \\ -0.30 - 0.35i & -0.30 + 0.35i & 0.061 \\ -0.59 + 0.11i & -0.59 - 0.11i & 0.16 \end{pmatrix}.$$

3.13.3 Flow over a cylinder

The complete closed form solution to the discovered dynamical system from the flow over a cylinder data is given by:

$$\mathbf{z}(t) = T \text{diag} \begin{pmatrix} e^{(-0.01+1.52i)t} \\ e^{(-0.01-1.52i)t} \\ e^{(0.11+1.05i)t} \\ e^{(0.11-1.05i)t} \\ e^{-0.20t} \\ 0 \end{pmatrix} T^{-1} \mathbf{z}_0, \quad (3.79)$$

where $\mathbf{z}_0$ is the state at $t = 0$ and

$$T = \begin{pmatrix} -0.35 + 0.20i & -0.35 - 0.20i & 0.028 - 0.006i & 0.028 + 0.006i & 0.005 & 0 \\ 0.15 - 0.39i & 0.15 + 0.39i & -0.12 - 0.45i & -0.12 + 0.45i & 0.68 & 0 \\ -0.10 + 0.37i & -0.10 - 0.37i & 0.034 + 0.57i & 0.034 + 0.57i & 0.23 & 1 \\ 0.24 + 0.33i & 0.24 - 0.33i & 0.013 - 0.21i & 0.013 + 0.21i & -0.37 & 0 \\ -0.13 - 0.38i & -0.13 + 0.38i & 0.034 - 0.27i & 0.034 + 0.27i & -0.55 & 0 \\ 0.44 & 0.44 & 0.58 & 0.58 & 0.21 & 0 \end{pmatrix}.$$

3.13.4 Isotropic turbulence flow

The complete closed-form solution of the discovered latent dynamical system for the isotropic turbulent flow data is the following:

$$\mathbf{z}(t) = T \text{diag} \begin{pmatrix} e^{(0.47+9.39i)t} \\ e^{(0.47+9.39i)t} \\ e^{(0.05+11.90i)t} \\ e^{(0.05+11.90i)t} \\ e^{(0.03+13.42i)t} \\ e^{(0.03+13.42i)t} \\ e^{(-0.04+3.46i)t} \\ e^{(-0.04+3.46i)t} \\ e^{-0.26t} \\ e^{(-0.27+5.44i)t} \\ e^{(-0.27+5.44i)t} \\ e^{(-0.75+8.30i)t} \\ e^{(-0.75+8.30i)t} \\ e^{(-1.37+18.72i)t} \\ e^{(-1.37+18.72i)t} \\ e^{-3.39t} \end{pmatrix} T^{-1} \mathbf{z}_0, \quad (3.80)$$

where $\mathbf{z}_0$ is the state at $t = 0$ and T contains all eigenvectors.

3.14 Algorithm

Algorithm 2 Latent state space regularization via SINDy

Input: sensor $S_{L+1:T}$, $X_{L+1:T}$, SINDy library $\Theta(\cdot)$, timestep Δt , regularization λ , threshold frequency δ_{thres} , Euler mini-step number k , temporal lag parameter L , learning rate η , total epoch T_{epoch} .

```

1: function LATENTSPACESINDY( $S_{L+1:T}, X_{L+1:T}, \Delta t, \lambda, \delta_{\text{thres}}, k, \eta, L$ ) :
2:   For  $i$  in  $0, 1, \dots, T_{\text{epoch}}$ :
3:      $Z_T, Z_{T+1} = f_{\theta_{\text{GRU}}}(S_{L+1:T-1}), f_{\theta_{\text{GRU}}}(S_{L+2:T});$ 
4:     For  $j$  in  $(0, k - 1)$ :  $\triangleright$  SINDy forward simulation
5:        $Z_{T+\frac{j+1}{k}\Delta t}^{\text{SINDy}} = Z_{T+\frac{j}{k}\Delta t}^{\text{SINDy}} + \Theta(Z_{T+\frac{j}{k}\Delta t}^{\text{SINDy}})\Xi\Delta t$ 
6:        $\hat{X}_{L+2:T} = f_{\theta_D}(Z_{T+1})$   $\triangleright$  SHRED reconstruction
7:        $(\theta_{\text{GRU}}, \Xi, \theta_D) = (\theta_{\text{GRU}}, \Xi, \theta_D) - \eta \nabla \mathcal{L}$   $\triangleright$  with  $\mathcal{L}$  defined in equation 4
8:       If  $i \bmod \delta_{\text{thres}} = 0$ 
9:          $\Xi[|\Xi| < \text{threshold}] = 0$   $\triangleright$  Train until converges

```

Output: GRU network weights θ_{GRU} , SINDy coefficients Ξ , decoder network weights θ_D .

4

Convergence of Uncertainty Estimates in Ensemble and Bayesian Sparse Model Discovery

4.1 Introduction

Data-driven model discovery methods based on sparse linear regression have been demonstrably successful, enabling the discovery of interpretable and generalizable models that balance accuracy and efficiency. The sparse identification of nonlinear dynamics (SINDy) [32] model discovery framework can leverage a variety of sparsity promoting algorithms, including the *sequential thresholding least-squares* (STLS) estimator introduced in the original paper. Recently, SINDy was extended to incorporate ensembling techniques, leveraging **bootstrap aggregating** (bagging) methods [30] to produce the robust and computationally efficient probabilistic model discovery method *ensemble SINDy* (E-SINDy) [90]. With the bootstrapping procedure, E-SINDy empirically improves the robustness of variable selection, especially for limited and noisy data, and enables statistically valid uncertainty quantification.

Variable selection and sparse regression have a rich history in data science and statistics. In classical theory, the likelihood ratio test is one of the earliest variable selection methods [291]. More recently, the least absolute shrinkage and selection operator (LASSO) uses an ℓ_1 shrinkage estimator for sparse inference [268]. The statistical properties of LASSO-based methods have been systematically studied, including the asymptotic behavior [95], consistency [319], the oracle property in classic regression settings [87, 326], and in high-dimensional settings [277, 278, 135]. From the Bayesian perspective, a hierarchical Bayesian model with spike-and-

slab [196, 107, 137], Laplace [213], or regularized horseshoe prior [44, 45] have been thoroughly studied for Bayesian sparse inference.

Many recent efforts discover the suboptimality of LASSO in the design-dependent setting and promote thresholding-based techniques to improve this problem [322, 215, 282]. The design-dependent setting encompasses not only the dynamical system discovery problem but also a variety of modern statistical learning challenges including multitask learning, transfer learning, neural network pruning, and so on [266]. Alternatively, thresholding-based estimators offer huge potential in these problems and have been widely applied in problems like wavelet regression [77], denoising [76], robust high-dimensional regression [22], and sparse identification of physical systems [32]. However, establishing the statistical properties of thresholding-based estimators is still an ongoing research. An important prior work on the statistical guarantees for STLS-based estimators investigated their oracle property in low- and high-dimensional settings [109]. The oracle property is defined as having both (i) ℓ_0 variable selection consistency and (ii) the same limiting distribution of non-zero coefficients if correct active indices are known.

Characterizing the effect of bootstrapping applied to thresholding-based sparse estimators adds another layer of complexity. Bootstrapping algorithms are computational methods to improve small sample inference that use the idea of resampling [79, 118, 80], which can help improve variable selection performance for LASSO [7, 193, 290]. Yet the combination requires careful analysis, for example, the residual-bootstrap LASSO is inconsistent for sparse models [51], but the residual-bootstrap distribution of Adaptive LASSO [326] is consistent. To the best of our knowledge, only a few studies have analyzed the effect of bootstrapping and STLS jointly. The detailed interaction between bootstrap and STLS still remains unknown.

In this paper, we study sparse linear regression using bootstrap aggregating inclusion probability STLS (or bagging-based STLS) estimation, a generalization of the sparse regression algorithm used in E-SINDy. We prove that the E-SINDy algorithm can perform correct sparse model identification with exponentially fast control of false discovery probability (FDP) and exponentially fast increase of the true discovery probability (TDP). Importantly, we achieve our proof within the context of *dynamical system discovery*, where the design matrix may have

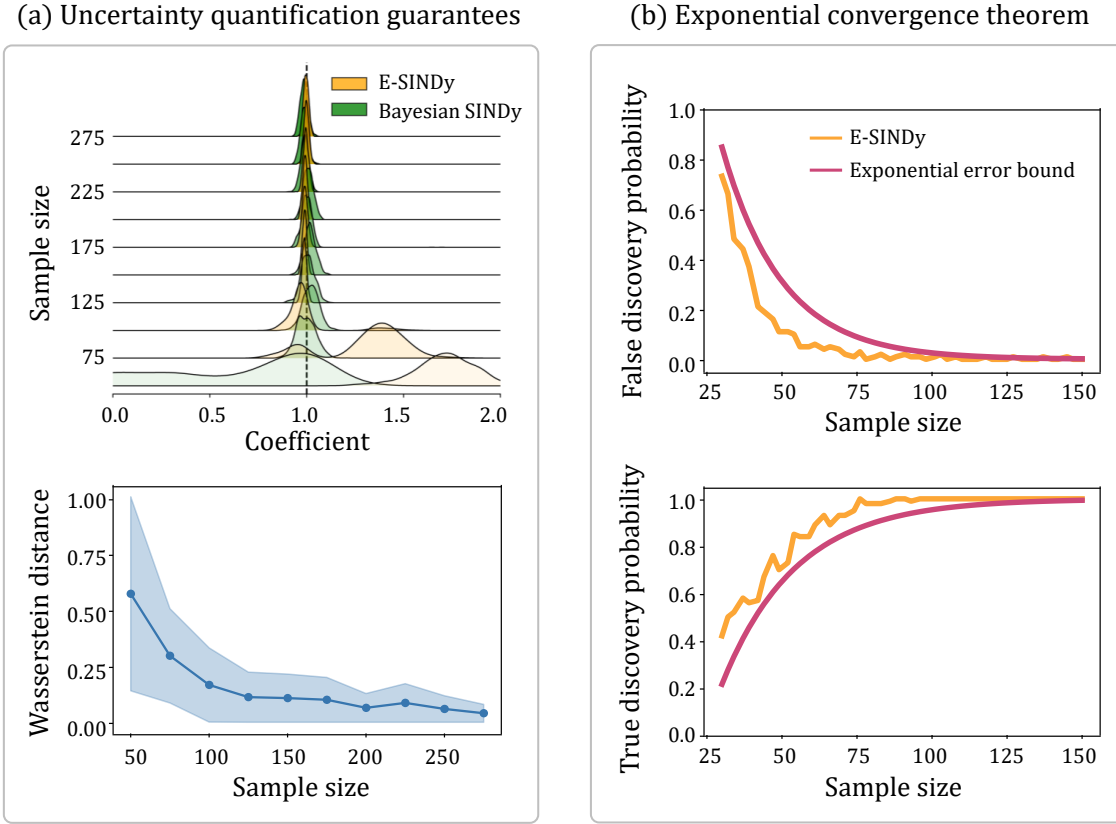


Figure 4.1: (a) Top: convergence of distributional estimation of the Bayesian SINDy (green) and E-SINDy algorithm (orange) over different sample sizes from the Lotka-Volterra equation coefficient α (as defined in Eqn. 4.26). Bottom: Wasserstein-2 distance between E-SINDy and Bayesian SINDy with Spike-and-slab prior. The shaded area indicates the 95% confidence band. (b) Exponential convergence for E-SINDy in terms of False Discovery Probability and True Discovery Probability for the synthetic Lotka-Volterra dataset. We numerically approximate the constant as our bounds only captures the rate of convergence.

dependent features and observations. This can introduce an ill-conditioned design matrix that presents a challenge to many classical methods for sparse signal recovery [278, 24, 277, 319], making sparse dynamical system discovery a distinct problem. Beyond variable selection consistency, our theoretical results for E-SINDy demonstrate a statistical guarantee for uncertainty quantification, requiring only zero mean and a bounded second moment for the noise distribution. These theoretical results highlight the effectiveness of E-SINDy for sparse model discovery and uncertainty quantification within the challenging context inherent to sparse identification of nonlinear dynamics.

With highly correlated, ill-conditioned data, the lowest eigenvalue of the Gram matrix

$\frac{1}{n}\mathbf{X}^\top\mathbf{X}$ typically shrinks quickly towards zero with increasing sample sizes. This scenario challenges many sparse regression algorithms. For example, the sample complexity of LASSO under Gaussian random design is $O\left(\frac{\rho_2}{\rho_1}\right)$, where ρ_1, ρ_2 is the smallest/largest eigenvalue of the Gram matrix [148]. In our work, under a shrinking lowest eigenvalue condition, we show the oracle property in Thm. 5 of bagging inclusion probability STLS with an exponential convergence rate under milder sample complexity assumptions compared to STLS [109]. This theoretical result proves that bagging-STLS guarantees identification of the correct model given sufficiently large sample sizes. Fig. 4.1 (b) shows the exponential convergence of error bounds and the empirical convergence using E-SINDy for the FDP and TDP.

In addition to the guarantee on variable selection, we show in Theorem 6 that the E-SINDy algorithm is statistically valid for uncertainty quantification (UQ). We demonstrate the similarity between the bagging-based STLS and Bayesian UQ methods with spike-and-slab prior [137, 126] from (i) asymptotic convergence in Wasserstein distance [204, 137] and (ii) Bayesian bootstrap Spike-and-Slab LASSO [207]. Compared to Bayesian UQ methods, bootstrapping-based methods can provide computationally efficient distributional estimates. For example, even in low-dimensional population dynamics model discovery [126], Bayesian SINDy requires a computational execution time of more than 12 hours, while E-SINDy takes less than 20 seconds. We also discuss the difference in computational complexity between Bayesian Markov Chain Monte Carlo (MCMC) and the bagging-based methods in Remark 4.5.2.

The effectiveness of E-SINDy for distributional convergence and uncertainty quantification is shown in Fig. 4.1 (a) top. The E-SINDy algorithm quickly converges to the true mean with an increasing number of samples, and generates an uncertainty estimate around the true value. In Fig. 4.1 (a) bottom, the solid blue line represents the average Wasserstein-2 distance over ten experimental trials. The shaded area shows the confidence interval estimate using the 80% percentile over the ten trials. The distributional estimation of E-SINDy converges to the estimation of Bayesian SINDy with large sample sizes. We examine this convergence behavior under Wasserstein distance in Thm. 7 from the asymptotic perspective.

We further apply bagging-based STLS to broader scenarios, from synthetic sparse linear

regression to physics model discovery. From the simulation of sparse linear regression under different settings, we observe an accelerated effect of bagging-based variable selection in terms of the sample size. Additionally, we perform a study on the robustness of hyperparameters in varying experimental settings for our bagging-based STLS and other baseline methods. The bagging-based STLS shows stable performance with respect to changes in sparsity and noise levels. For governing equation discovery, we observe that the E-SINDy with bagging-based STLS can accurately perform variable selection with uncertainty quantification. The estimation bias quickly reduces with more observation points and the uncertainty estimation is valid under noisy settings. The experimental study, as the supplement of Thm. 6, shows that E-SINDy provides valid uncertainty quantification that is asymptotically equivalent to Bayesian MCMC with sparsifying priors [126].

The contributions of this paper are threefold:

1. We study a theoretical extension of the E-SINDy algorithm [90] with bagging inclusion probability STLS and analyze its oracle property under regularity conditions with an exponential convergence rate.
2. We show that bagging inclusion probability STLS is asymptotically converging to a Gaussian limit which is equivalent to the Bayesian method with a lower computational cost. The bagging-based algorithm can also provide valid uncertainty quantification statistically.
3. We implement and verify the effect of bagging inclusion probability STLS in both synthetic sparse linear regression settings and for physical equation discovery. We verify the effect of accelerated variable selection and the validity in uncertainty quantification.

In what follows, we first introduce the SINDy algorithm and variable selection methods in Sec. 4.2. We then review the analysis on sequential thresholding least-squares estimation in Sec. 4.3, and propose a theoretical analysis for E-SINDy with bagging inclusion probability-based STLS in Sec. 4.4. We utilize Sec. 4.5 to provide the uncertainty quantification guarantees and connections to the Bayesian method. Finally, we evaluate our work by conducting various experiments on sparse linear regression and sparse physics model discovery in Sec. 4.6. Our code implementation is available in <https://github.com/gaoliyao/BootstrapTLSE>.

4.2 Background

4.2.1 Sparse identification of nonlinear dynamics

The *sparse identification of nonlinear dynamics* (SINDy) [32] algorithm is a thresholding least-square sparse regression technique to identify the underlying dynamical system from observational data snapshots $\mathbf{x}(t) \in \mathbb{R}^n$ where

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)). \quad (4.1)$$

The data snapshots $\mathbf{x}(t)$ are measured at time t , and the function $\mathbf{f}$ represents the underlying dynamical system that is to be discovered. By calculating the temporal derivatives from the snapshot data, SINDy forms the following covariates and targets:

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}(t_1) \\ \mathbf{x}(t_2) \\ \vdots \\ \mathbf{x}(t_n) \end{pmatrix}, \quad \dot{\mathbf{X}} = \begin{pmatrix} \dot{\mathbf{x}}(t_1) \\ \dot{\mathbf{x}}(t_2) \\ \vdots \\ \dot{\mathbf{x}}(t_n) \end{pmatrix}.$$

A function library is constructed with p candidate model terms so that $\boldsymbol{\Theta}(\mathbf{X}) = [\theta_1(\mathbf{X}) \cdots \theta_p(\mathbf{X})] \in \mathbb{R}^{n \times p}$. The candidate functions can be polynomials, sinusoids, exponentials, and so on. In summary, we hope to derive a sparse model between $\mathbf{X}$ and $\dot{\mathbf{X}}$ such that

$$\dot{\mathbf{X}} = \boldsymbol{\Theta}(\mathbf{X})\boldsymbol{\Xi}, \quad (4.2)$$

where the unknown sparse matrix $\boldsymbol{\Xi} = (\xi_1 \ \xi_2 \ \cdots \ \xi_n) \in \mathbb{R}^{p \times n}$ enables the correct identification of the dynamics.

Difference to classical sparse regression settings

The statistical model of sparse identification of dynamical system is different from many prior classical settings in sparse recovery [40, 42]. In this model, we have (i) an ill-conditioned

design matrix, (ii) mutually coherent columns, and (iii) irregular noises. These differences could suggest and promote very different learning algorithms for sparse subset recovery.

We first notice that $\mathbf{x}(t) = [\mathbf{x}(t_1), \mathbf{x}(t_2), \dots, \mathbf{x}(t_m)]$ is a time-series with strong temporal dependency. For dynamical system discovery, having larger sample size n typically represents finer temporal sampling rates (or equivalently smaller timesteps $\Delta t = t_{i+1} - t_i$) [152]. For very small Δt , sample pairs $(\mathbf{x}(t_i), \mathbf{x}(t_{i+1}))$ can be near identical. Therefore, in the context of dynamical systems discovery, the effective sample size is generally very small. Additionally, due to practical constraints, many experiments are difficult to replicate so the measurements could only come from a single trajectory, e.g. the predator-prey data [256].

Therefore, the design matrix in this setting will be ill-conditioned from potentially similar linear copies, and the lowest eigenvalue of the sample covariance matrix could go to zero with a rate of $\mathcal{O}(\frac{1}{n})$. Ill-conditioned input matrices challenge the commonly used lower eigenvalue and mutual incoherence conditions in many literature for variable selection consistency (c.f. [278] 7.43).

Additionally, in the context of dynamical system discovery, the library of functions are mutually coherent. For example, a commonly used function in the Fourier library $\sin(z)$ can be Taylor approximated via z and z^3 (which exists in the polynomial library). In general, random variables z , z^2 , and z^3 are not independent. This directly challenges the mutual incoherence condition (c.f. [278] [7.43b]) which requires features to be close orthogonal. Furthermore, considering the stochasticity in the measurements, the observation noise will be transformed into a non-trivial and heteroscedastic noise distribution, which makes the estimation of the tail much more complex for uncertainty quantification.

In short, the SINDy problem exists in a setting that lacks an orthogonal design and involves a complex noise distribution. Although SINDy often addresses an over-determined system (characterized by a tall-skinny matrix), the ill-conditioned nature of the design matrix still leads to a complex nullspace, where multiple approximately correct solutions are possible. Therefore, simply assuming SINDy to be a straightforward classical regression problem oversimplifies its complexity, yet this misconception is common.

Application of SINDy in scientific machine learning

Typically, SINDy [32] applies STLS estimation to perform sparse inference, which is a proxy for ℓ_0 optimization [321, 320] with convergence guarantees [314]. Hirsh et al. [126] perform Bayesian sparse inference in model discovery via Spike-and-slab and regularized horseshoe priors. Fasel et al. [90] combine bootstrap resampling to perform uncertainty estimation with observed acceleration in variable selection for system identification. SINDy has been applied to diverse physical systems including fluid dynamics [180, 181, 179, 113, 69, 38], nonlinear optics [254], turbulence closures [18, 19, 240], ocean closures [312], chemical reactions [131], plasma dynamics [67, 4, 144], structural modeling [162], granular materials [318], and for model predictive control [140]. Extensions of SINDy include the identification of partial differential equations [234, 238], parametrically dependent dynamical models [233], multiscale physics [47], time-dependent PDEs [52], rational function nonlinearities [189, 139], switching dynamical systems [188], control inputs [140], constraints on symmetries [180], control for stability [143], control for robustness [4, 239, 223, 224, 194, 194], stochastic dynamical systems [28, 37], multidimensional approximation on tensors [106], and video data in pixel space [46, 9, 99].

4.2.2 Variable selection, bootstrap resampling, and uncertainty quantification

From the perspective of uncertainty quantification, sparse models have different approaches for uncertainty estimation. The bootstrap method is one way to obtain an uncertainty estimate of sparse regression. Chatterjee and Lahiri [50] first showed residual-bootstrap of LASSO is inconsistent. Chatterjee and Lahiri [51] then showed that the residual-bootstrap of adaptive LASSO estimator is asymptotically consistent. Extending from the bootstrap approximation, Bayesian methods are more favorable for obtaining uncertainty estimates of sparse models with sparsifying priors. Mitchell and Beauchamp [196] first established Spike-and-slab models. These were further developed by George and McCulloch [107]. Ishwaran and Rao proposed the general theoretical framework of Spike-and-slab prior in [137]. Carvalho, Polson, and Scott utilized the horseshoe prior [44, 45] to perform Bayesian sparse inference. Computations of the

above-mentioned Bayesian methods rely heavily on hierarchical MCMC sampling. Ročková and George [228] proposed Spike-and-slab LASSO that focuses on mode searching instead of traversing the entire distribution. Nie and Ročková [207] applied Bayesian bootstrap to Spike-and-slab LASSO to obtain uncertainty estimation with a first-order correctness guarantee. The Spike-and-slab LASSO method is a thresholding procedure(c.f. [207] Eqn. (4)). Giurcanu established the analysis to show the consistency of the residual bootstrap for the thresholding least squares estimator (TLSE) [109].

Bootstrap resampling and aggregation can frequently improve the performance of unstable estimators. Bühlmann and Yu [33] examined this idea through case studies in bagging with indicators and subsampling aggregating decision trees. Bühlmann and Yu [34] also introduced sparse boosting to obtain good performance in ℓ_2 . Later in Bolasso [7], Bach proposed a LASSO method that is model consistent via bootstrapping with improved performances in variable selection. Meinshausen and Bühlmann established stability-based selection which provides a consistent variable selection process [193]. They showed the error control bound of false discovery variables and theoretical guarantee of randomized LASSO algorithm under sparse eigenvalue condition. Recently, Werner [290] proposed loss-guided stability selection; Soloff, Barber and Willett showed bagging can provide algorithm stability that is free of assumption [251].

4.3 Sequential thresholding least-square in sparse regression

In this section, we establish the theoretical foundations of the STLS estimator in a sparse linear model. We constrain our analysis to the primary effect of STLS by only performing the thresholding procedure twice from the ordinary least-squares estimate, which is more accessible for theoretical analysis and occurs often in practice [314].

Consider a linear model that

$$\mathbf{y} = \mathbf{X}\beta + \epsilon, \quad (4.3)$$

where $\mathbf{y} \in \mathbb{R}^n$ is the response; $\mathbf{X} \in \mathbb{R}^{n \times p}$ is the covariate which contains n samples and p

features; $\beta \in \mathbb{R}^p$ is an unknown parameter corresponding to the p features; and $\epsilon \stackrel{\text{i.i.d.}}{\sim} P$ is the noise following a distribution P on $\mathbb{R}$ satisfying $\mathbb{E}[\epsilon] = 0$, and $\text{var}(\epsilon) = \sigma^2 < \infty$. Denote $\mathbf{X}_j$ to be the j -th column (feature) of the data. To facilitate the analysis, we recenter the response and standardize the covariates so that the sample means of $\mathbf{Y}$ and each $\mathbf{X}_j$ to be 0, and the sample variance of each $\mathbf{X}_j$ to be 1. In the context of SINDy (Eqn. (4.2)), the function library $\Theta(\mathbf{X})$ provides the covariate matrix $\mathbf{X}$; $\mathbf{X}$ represents the response vector $\mathbf{y}$; and Ξ is equivalent to the unknown parameter β .

For system identification, we wish to obtain a sparse model that is the most explainable. We set q to be the number of non-zero components of β so that $q < p$. In our setting, we consider the model size p to be of the same (asymptotic) scale to that of the sample n under sparsity constraints on β . We focus on the case when $n > p$ with independent experimental trials.

4.3.1 Thresholding least-squares estimate

Let $\bar{\beta}$ be the ordinary least-squares (OLS) solution of β such that

$$\bar{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}. \quad (4.4)$$

We define ρ_1, ρ_2 to be the minimum and the maximum eigenvalues of $\frac{1}{n} \mathbf{X}^\top \mathbf{X}$. One can achieve asymptotic normality via Lindeberg-Feller central limit theorem (CLT) with the following conditions.

Lemma 6. *Given the following conditions:*

$$(3.1.a) \mathbb{E}[\epsilon] = 0, \text{var}(\epsilon) = \sigma^2 < \infty, \text{ and } \mathbb{E}[\epsilon^4] < \infty \text{ where } \epsilon \stackrel{\text{i.i.d.}}{\sim} P.$$

$$(3.1.b) \rho_1 > 0, \max_i(\mathbf{X}_i^\top \mathbf{X}_i) = O(p), \text{ and } p/(n\rho_1) = o(1). \text{ Then } \forall \mathbf{a} \in \mathbb{R}^p, \text{ and } \|\mathbf{a}\|_2^2 = 1,$$

$$\mathbf{a}^\top (\bar{\beta} - \beta) \xrightarrow{d} \mathcal{N}(0, \sigma^2 \mathbf{a}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{a}), \quad (4.5)$$

and for $\delta \in (0, 1)$ we have

$$\mathbb{P}\left(\mathbf{a}^\top(\bar{\beta} - \beta) \geq \sqrt{2\sigma^2 \log(1/\delta) \mathbf{a}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{a}}\right) \leq \delta. \quad (4.6)$$

Condition (3.1.a) imposes constraints only on the noise, whereas condition (3.1.b) requires the design matrix to satisfy certain regularity properties. Together, these assumptions are tailored to capture scenarios involving ill-conditioned design matrices. We show a proof of Lemma 6 in the Appendix. From Lemma 6, we can have a tail bound of the OLS with probability at least $1 - \delta$,

$$\mathbf{a}^\top(\bar{\beta} - \beta) \leq \sigma \sqrt{2 \log(1/\delta) \mathbf{a}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{a}}. \quad (4.7)$$

However, without further assumption on the noise, the tail behavior of the OLS is only valid in the asymptotic sense. Therefore, in the following Lemma 7, we make a stronger assumption with Gaussian noise to capture the rate of convergence in the non-asymptotic sense. We first introduce the following assumption that

$$\epsilon \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2 \mathbf{I}_n),$$

where $\mathbf{I}_n$ represents an identity matrix with n rows and n columns.

Let I to be the set of all indices where $I = \{1, \dots, p\}$. Here, we let K_β be the non-active indices that $K_\beta = \{j \in I : \beta_j = 0\}$, and we define J_β to be the active indices that $J_\beta = \{j \in I : \beta_j \neq 0\}$. Let $\hat{K}$ be the thresholding estimator of the set of non-active indices of β . To distinguish between different estimators, we use a subscript notation (e.g. $\hat{K}_{STLS}$) to specify the particular estimator. Motivated by Lemma 6 and the Gaussian tail bound, we set the STLS estimate of $\hat{K}$ to be

$$\hat{K}_{STLS} = \{j \in I : |\bar{\beta}_j| \leq \sqrt{2\sigma^2 \log(1/\delta) (\mathbf{X}^\top \mathbf{X})_{jj}^{-1}}\}. \quad (4.8)$$

The sequential thresholding least-squares estimator in our paper ($\hat{\beta}$) is defined as the following:

$$\hat{\beta} = \arg \min_{b \in \mathbb{R}^p} \{(\mathbf{y} - \mathbf{X}b)^\top (\mathbf{y} - \mathbf{X}b) : b_{\hat{K}} = 0\}, \quad (4.9)$$

where $b_{\hat{K}} = 0$ represents that $\forall j \in \hat{K}, b_j = 0$.

Equivalently, we can write $\hat{\beta}$ as

$$\begin{cases} \hat{\beta}_{\hat{K}_{STLS}} = 0, \\ \hat{\beta}_{\hat{J}_{STLS}} = \left(\mathbf{X}_{\hat{J}_{STLS}}^\top \mathbf{X}_{\hat{J}_{STLS}} \right)^{-1} \mathbf{X}_{\hat{J}_{STLS}}^\top \mathbf{y}, \end{cases} \quad (4.10)$$

where $\hat{J}_{STLS} = I \setminus \hat{K}_{STLS}$, and $\mathbf{X}_{\hat{J}}$ denotes the matrix containing all columns of $\mathbf{X}$ indexed by $\hat{J}$.

We define the *False Discovery Probability* (FDP) as follows. Given the non-active set estimation as $\hat{K}$, the FDP is defined as

$$\text{FDP} = 1 - \mathbb{P} \left(K_\beta \subset \hat{K} \right). \quad (4.11)$$

The FDP characterizes the probability of whether the current estimate $\hat{J} = I \setminus \hat{K}$ contains false discovery. This metric is stronger compared to the *False discovery rate* (FDR) due to the fact that the FDR must be 0 when $K_\beta \subset \hat{K}$ is true.

Similarly, we define the *True Discovery Probability* (TDP) such that

$$\text{TDP} = \mathbb{P} \left(\hat{K} \subset K_\beta \right). \quad (4.12)$$

The TDP characterizes the probability of whether the current estimate $\hat{J}$ contains all true active coefficients. This metric is also stronger compared to the *true positive proportion* (TPP) due to the fact that the TPP must be 1 when $\hat{K} \subset K_\beta$ is true.

From [109], we present the following Lemma under a Gaussian tail assumption in the following.

Lemma 7. *Suppose that the following conditions hold:*

$$(3.2.a) \ \epsilon_i \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2 \mathbf{I}_n).$$

$$(3.2.b) \frac{q}{n} = o(1).$$

$$(3.2.c) \rho_1 > 0 \text{ and } \frac{\rho_1 \min_{i \in J} |\beta_i|}{\sigma^2} = \omega\left(\frac{\log(n)}{n}\right).$$

Then $\hat{\beta}$ has the oracle property $\mathbb{P}(\hat{K} = K_\beta) = 1$ since the following holds:

(i) The False Discovery Probability (FDP) satisfies

$$1 - \mathbb{P}\left(K_\beta \subset \hat{K}_{STLS}\right) \leq \frac{q}{n}. \quad (4.13)$$

(ii) The True Discovery Probability (TDP) satisfies

$$\begin{aligned} & \mathbb{P}\left(\hat{K}_{STLS} \subset K_\beta\right) \geq \\ & 1 - 2 \left(-\frac{1}{2} \left(\left(\frac{n^{1/2} \rho_1^{1/2} \min_{i \in J_\beta} |\beta_i|}{\sigma} - \sqrt{2 \log(n)} \right)^2 - 2 \log(q) \right) \right). \end{aligned} \quad (4.14)$$

4.4 Bagging sequential thresholding least-square estimator

Bootstrap aggregating (bagging) is a computational technique to perform uncertainty quantification, variance reduction, and bias estimation. The benefit of the bagging [30] often comes from that fact that it could stabilize unstable statistics [33] (cf. Definition 1.2). In the case of the STLS estimator, the inclusion of an indicator function makes the STLS unstable. As stated in Sec. 2 in [33], we expect to see a boosted effect from bagging due to the smoothing instead of the hard thresholding operation. We present the **Bagging Inclusion Probability-based** (BIP-based) STLS estimate along with the estimation via inclusion probability which can also be understood as a selection probability [193]. We analyze and claim that the inclusion probability-based variable selection can achieve the same oracle property with less restrictive assumptions on sample size.

4.4.1 Bagging inclusion probability STLS

Algorithm 3 Bagging inclusion probability thresholding least-square estimation

Input: Covariate matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, target vector $\mathbf{y} \in \mathbb{R}^n$, threshold γ , subsample proportion c , inclusion probability threshold p_c , noise level σ .

Output: Estimated support set $\hat{\mathbf{J}}$

```

1: function BAGGINGINCLUSIONPROBABILITY( $\mathbf{X}, \mathbf{y}, \gamma, c, p_c, \sigma$ )
2:   IncludeList = List();
3:   for  $i$  in  $1, 2, \dots, n$  do
4:      $\mathbf{X}^b, \mathbf{y}^b = \text{Subsample}(\mathbf{X}, \mathbf{y}, c)$ ;
5:      $\boldsymbol{\beta}^b = \text{LeastSquares}(\mathbf{X}^b, \mathbf{y}^b)$ ;
6:     thresholds $_{1:p} = \sigma \sqrt{\frac{\gamma}{(\mathbf{X}^b)^\top \mathbf{X}^b}}$ ;
7:     include $_{1:p} = \text{BooleanVector}(|\boldsymbol{\beta}^b| > \text{thresholds})$ ;
8:     IncludeList.append(include);
9:   end for
10:  InclusionProb $_{1:p} = \frac{1}{n} \sum_{i=1}^n \text{IncludeList}[i]$ ;
11:   $\hat{\mathbf{J}}_{1:p} = \text{BooleanVector}(\text{InclusionProb} > p_c)$ ;
12:  return  $\hat{\mathbf{J}}$ 
13: end function

```

We use B bootstrap replicates and create B estimates denoted by $\{\hat{K}_{STLS}^b\}_{b=1}^B$. We apply the standard bootstrap procedure to facilitate later analysis of distributional convergence in Sec. 4.5. In the current subsection, we consider the bagging estimate of the inclusion probability of $j \in I$ with a constant thresholding probability p_c where

$$\hat{K}_{BIP} = \left\{ j \in I : \left(\frac{1}{n} \sum_{b=1}^n \mathbb{1} \left\{ |\bar{\beta}_j^b| > \sigma \sqrt{\frac{2 \log(1/\delta)}{(\mathbf{X}^b)^\top \mathbf{X}^b}_{jj}}} \right\} \right) \leq p_c \right\},$$

where $\mathbb{1}\{\cdot\}$ is an indicator function that takes value 1 if the event in brackets is true and 0 otherwise.

The idea of bagging inclusion probability estimation $\hat{K}_{BIP}$ (demonstrated in Algorithm 3) is very straightforward. We count the number of index j that appear in all bootstrap replicates $\hat{K}_{STLS}^b$ and divide the count by n . If the inclusion probability for an index j is low (e.g. $p_j < 0.05$), we say j is not active since most bootstrap replicates do not contain j . On the other side, we say the index j is active when the inclusion probability is high. We note here that Algorithm 3 simplifies to STLS algorithm when the for loop is omitted and all the data is used

for least-squares (by setting $c = 0$). In other words, without the bootstrap resampling steps, Algorithm 3 reduces to the standard STLS estimation. We note that, to facilitate theoretical analysis, the bagging inclusion probability STLS algorithm simplifies the original E-SINDy algorithm by using (i) only two rounds of the thresholding procedure and (ii) the bootstrap mean instead of the bootstrap median.

4.4.2 The analysis on bagging inclusion probability estimation

In the following, we show that the bagging inclusion probability estimate $\hat{K}_{BIP}$ has ℓ_0 consistency that, with a sufficient number of samples, $\hat{K}_{BIP}$ will converge to the correct K_β .

We use the following theorem to show the goodness of the bootstrap inclusion probability estimator $\hat{K}_{BIP}$.

Theorem 5. *Suppose that the following conditions hold and let $\Phi(\cdot)$ be the CDF of the standard normal distribution:*

$$(4.1.a) \ \epsilon_i \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2 \mathbf{I}_n).$$

$$(4.1.b) \ \frac{q}{p} = O(1), \ \frac{p}{\exp(n)} = o(1).$$

$$(4.1.c) \ \text{For any bootstrap replicate } \mathbf{X}^b, \text{ the lowest eigenvalue of } \frac{1}{n} \mathbf{X}^{b\top} \mathbf{X}^b \text{ is lower bounded by } r_0 \rho_1$$

where $r_0 \in \mathbb{R}$ is a constant.

$$(4.1.d) \ \rho_1 > 0, \text{ and } \frac{\rho_1 \min_{i \in J} |\beta_i|}{\sigma^2} = \omega \left(\frac{\log(n)}{n} \right).$$

Then, we have the following:

(i) *The False Discovery Probability (FDP) satisfies*

$$1 - \mathbb{P} \left(K_\beta \subset \hat{K}_{BIP} \right) \leq (p - q) \exp \left(\frac{1}{3} - \frac{np_c}{3} \right). \quad (4.15)$$

(ii) *Define $p_{j_{\min}} = \min_{j \in J_\beta} \Phi \left(\frac{\sqrt{nr_0 \rho_1} |\beta_j|}{\sigma} - \sqrt{2 \log(n)} \right)$. The True Discovery Probability (TDP) satisfies*

$$\mathbb{P} \left(\hat{K}_{BIP} \subset K_\beta \right) \geq 1 - q \exp \left(-\frac{p_c^2}{3p_{j_{\min}}} + \frac{2p_c}{3} - \frac{np_{j_{\min}}}{3} \right). \quad (4.16)$$

(iii) Under (4.1.d), we have the Oracle Property that

$$\mathbb{P} \left(\hat{K}_{BIP} = K_\beta \right) \rightarrow 1. \quad (4.17)$$

We defer the proof to the Appendix. This proof achieves an explicit rate of convergence from the Gaussian tail assumption in (4.1.a) with the associated concentration bound. We require (4.1.b) to show the asymptotic convergence and (4.1.c) to control the quality of the data after bootstrap replication. Generally, one could expect the least-square estimate of sparse coefficients will shrink towards 0. The inclusion probability estimated by bootstrapping will be more accurate as the sample size increases, so the sparse subset estimation will become more robust given a reasonable threshold probability.

It is important to understand the role of (4.1.d), where the lowest eigenvalue can go to zero with increasing n , but needs to be asymptotically larger than $O\left(\frac{\log(n)}{n}\right)$. This shrinking lowest eigenvalue assumption can violate the mutual incoherence condition for LASSO recovery (c.f. [278] [7.43b]). In Sec. 6.2, we will use synthetic simulation to demonstrate how LASSO loses its advantage in this regime.

From Thm. 5, we know $\hat{K}_{BIP}$ is a very good estimator of K_β with an exponential convergence rate for both FDP and TDP. STLS could only reach exponential behavior on one side (either FDP or TDP) but not both. The exponential rate shows rapid decay in FDP and rapid increase in TPD in Sec. 6. Utilizing this good estimation on K_β , one expects to achieve accurate inference on the unknown parameter β with many estimators.

Determine a threshold probability. The setting of hyperparameters is not negligible for sparse regression methods. For example, LASSO depends heavily on the ℓ_1 constraint and STLS requires a reasonable least-square threshold. Bagging inclusion probability-based variable selection provides an alternative to determine the threshold probability via data analysis. The inclusion probability usually shows a clear gap between active and non-active indices. We define this gap as Δ and formalize this idea by extending Thm. 5 with the following Lemma.

Lemma 8. *Suppose that the following conditions hold and let $\Phi(\cdot)$ be the CDF of standard normal*

distribution:

$$(4.2.a) \epsilon_i \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2 \mathbf{I}_n).$$

(4.2.b) For any bootstrap replicate $\mathbf{X}^b$, the lowest eigenvalue of $\frac{1}{n} \mathbf{X}^{b\top} \mathbf{X}^b$ is lower bounded by $r_0 \rho_1$

where $r_0 \in \mathbb{R}$ is a constant.

$$(4.2.c) \rho_1 > 0, \epsilon \text{ satisfies } \epsilon + \frac{1}{n} < p_{j_{\min}} \text{ with the definition that } p_{j_{\min}} = \min_{j \in J_\beta} \Phi \left(\frac{\sqrt{n r_0 \rho_1} |\beta_j|}{\sigma} - \sqrt{2 \log(n)} \right).$$

Then, the inclusion probability gap Δ satisfies $\Delta > \epsilon$ with probability at least $1 - p e^{-n(\frac{1}{2} p_{j_{\min}} - \frac{\epsilon}{2} - \frac{1}{2n})^2 / 2}$

where ϵ is a constant that $\epsilon \in (0, p_{j_{\min}})$.

The previous Lemma 8 is derived via a similar process to the proof of Thm. 5. Consider the case when $p = 30, n = 100$, according to the bound, we have with probability at least 97% that we can observe the inclusion probability gap between the minimum of active indices and the maximum of non-active indices greater than 0.2. By running the algorithm once, one could clearly determine the active set of indices, and utilize this information to select the threshold probability. We further wish to note that in practice, the gap should be even more obvious compared to the above Lemma - as we simplified the discussion of $\mathcal{I}_1$ in the proof. In numerical experiments, the gap could be very significant where we observe the inclusion probabilities of active indices are empirically lower bounded by 0.99, and non-active indices empirically are upper bounded by 0.01. Even under cases with exceptionally high observation noise, it is accessible to perform cross-validation and use the threshold that has the lowest validation error.

4.5 Uncertainty quantification with bootstrap

In practice, it is essential to understand the uncertainty of parameter inference. Bootstrapping is one of the most classical computational methods to quantify uncertainty. Bootstrap uncertainty estimates are frequently preferred for "non-asymptotic" sample size [279]. In the context of sparse identification of dynamical systems, bootstrapping is an important technique for uncertainty estimation where the noise often follows a complex distribution.

We first recall the residual-bootstrap inference process from [94, 80]. Define sample residuals $E_{1:n} = \{\epsilon_1, \dots, \epsilon_n\}$ that $\epsilon_i = \mathbf{y}_i - \mathbf{X}_i^\top \hat{\beta}$. Let $\mathbb{P} = n^{-1} \sum_{i=1}^n \delta_{\epsilon_i}$ to be the empirical

distribution where δ_{ϵ_i} is a point mass. By sampling with replacement, we obtain a bootstrap sample $E_{1:m}^* = \{\epsilon_1^*, \dots, \epsilon_m^*\}$ which consists of m conditional i.i.d. samples from $\mathbb{P}$. Then we apply the resampled residuals to the target via $\mathbf{y}_i^* = \mathbf{X}_i^\top \hat{\beta} + \epsilon_i^*$.

A very natural way is to construct bootstrap replicates using $\arg \min_{b \in \mathbb{R}^p} \{(\mathbf{y}^* - \mathbf{X}b)^\top (\mathbf{y}^* - \mathbf{X}b) : b_{\hat{K}^*} = 0\}$ where $\hat{K}^*$ is estimated from different bootstrap replicates. However, in this case, each bootstrap replicate will potentially have different sparse subset estimation $\hat{K}^*$, which could be troublesome to analyze in theory. Therefore, we utilize $\hat{K}_{BIP}$ defined in the previous section which could similarly make use of all the bootstrap replicates and, at the same time, avoid the randomized behavior of bootstrapped subset estimation in theory. In other words, we consider the estimator $\hat{\beta}^*$ is computed via that

$$\hat{\beta}^* = \arg \min_{b \in \mathbb{R}^p} \{(\mathbf{y}^* - \mathbf{X}b)^\top (\mathbf{y}^* - \mathbf{X}b) : b_{\hat{K}_{BIP}} = 0\}. \quad (4.18)$$

From each bootstrap replicate $\hat{\beta}^*$, the starred residuals are defined as $\hat{\epsilon}_{1:m}^* = \mathbf{y}^* - \mathbf{X}\hat{\beta}^*$. Notice it is different from the setting in prior work [109] which only utilizes $\hat{\beta}$. The bootstrap version of standard error estimates are

$$\mu_m^* = \frac{1}{m} \sum_{i=1}^m \hat{\epsilon}_i^* \quad \text{and} \quad \hat{\sigma}_m^{*2} = \frac{1}{m} \sum_{i=1}^m \hat{\epsilon}_i^{*2} - \mu_m^{*2}.$$

From the bootstrap estimate of the standard error, we define $\hat{s}_m$ as

$$\hat{s}_m = n^{-1} \hat{\sigma}_m^{*2} \mathbf{a}_j^\top (\mathbf{X}_j^\top \mathbf{X}_j)^{-1} \mathbf{a}_j. \quad (4.19)$$

4.5.1 Consistency of bootstrap inference

To understand the consistency of bootstrap inference in an asymptotic sense. It facilitates an asymptotically correct distributional estimation of the pivot quantity $\sqrt{n}(\hat{\beta} - \beta)$ from Monte Carlo samples of $\sqrt{n}(\hat{\beta}^* - \hat{\beta})$. Based on the selection probability, one could propose bootstrap replicates using $\hat{\beta}^*$ based on $\hat{K}_{BIP}$. The bagging inclusion probability-based bootstrapping uncertainty quantification does not require any extra distributional assumptions, which is an important advantage for practical applications. Under the following conditions, we can see

the bootstrap distribution $\mathcal{L}(\hat{s}_m^{*-1/2} a^\top (\hat{\beta}_m^* - \hat{\beta}) \mid \mathbf{y})$ is consistent.

Theorem 6. *Suppose that the following conditions hold:*

$$(5.1.a) \mathbb{E}[\epsilon] = 0, \text{var}(\epsilon) = \sigma^2 < \infty, \epsilon \stackrel{i.i.d.}{\sim} P.$$

$$(5.1.b) \rho_1 > 0, \max_i (\mathbf{X}_i^\top \mathbf{X}_i) = O(p), \text{ and } p/(n\rho_1) = o(1).$$

(5.1.c) *The min and max eigenvalues ρ_1, ρ_2 satisfy $\frac{\rho_1 \rho_2^{-1} \gamma^2 - q}{q^{1/2}} \rightarrow \infty$ where $\gamma \in \mathbb{R}$ is a thresholding parameter.*

$$(5.1.d) \frac{\rho_1 \rho_2^{-1} \left(n^{1/2} \sigma^{-1} \rho_1^{1/2} \min_{j \in J_\beta} |\beta_j| - \gamma \right)^2 - q}{q^{1/2}} \rightarrow \infty.$$

Then, for $a \in \mathbb{R}^p$ and $\|a\|_2^2 = 1$, we have

$$\mathcal{L}(\hat{s}_m^{*-1/2} a^\top (\hat{\beta}_m^* - \hat{\beta}) \mid \mathbf{y}) \xrightarrow{Pr} \mathcal{N}(0, 1), \quad (4.20)$$

for $\beta \in \Theta$.

Here, $\mathcal{L}(\cdot \mid \mathbf{y})$ denotes the conditional law (distribution) of the random variable given $\mathbf{y}$. We demonstrate the proof of Thm. 6 in the Appendix motivated from [109] by utilizing the Lindeberg-Feller Central Limit Theorem. This theorem demonstrates that the bagging inclusion probability-based bootstrapping uncertainty quantification is consistent in the asymptotic sense under the finite variance assumption in (5.1.a). We note here that the finite variance assumption is much weaker compared to the assumption in prior work. We require (5.1.b) to keep the Gram matrix well-behaved as the library size grows. The lowest eigenvalue ρ_1 may shrink with n , but the least-squares estimates remain stable. The conditions (5.1.c) and (5.1.d) are used to control the thresholding parameter γ and the minimum of active coefficients $\min_{j \in J_\beta} |\beta_j|$. We ensure the physics does not decay with increasing n and the thresholding parameter is within a reasonable range. The asymptotic consistency provides theoretical evidence of our algorithm for uncertainty quantification.

4.5.2 Connection to Bayesian uncertainty quantification

Bayesian MCMC with sparsifying priors gives a full Bayesian approach for sparse model inference [137, 126]. Exemplified with a Spike-and-slab prior, we use a hierarchical Bayesian

model for UQ-SINDy [126] such that

$$\begin{aligned}\beta_j \mid \lambda_j &\sim \lambda_j \mathcal{N}(0, c^2) + (1 - \lambda_j) \mathcal{N}(0, \epsilon^2), \\ \lambda_j &\sim \text{Ber}(\pi),\end{aligned}\tag{4.21}$$

with normal likelihood model (c.f. [126] Eqn. (3.3)), and $\text{Ber}(\cdot)$ denotes the Bernoulli distribution. The likelihood model in UQ-SINDy employs an integration over the entire trajectory, which is equivalent to having an estimating equation with more constraints. In theory, we consider a simplified version via Bayesian sparse linear regression in the following. We note here that the UQ-SINDy is likely to have a much lower sample complexity compared to this simplification, though demonstrating this rigorously is challenging. A classical result shows that the unrescaled spike and slab model admits the form of generalized ridge regression, and its posterior mean asymptotically converges in probability to the OLS. This implies that the spike and slab model estimation also converges in distribution to the OLS estimate.

Define the Wasserstein-2 (W2) distance [275] of two probability distributions (measures) μ and ν to be

$$\mathcal{W}_2(\mu, \nu) = \inf_{\pi \in \Pi(\mu, \nu)} \left(\mathbb{E}_{(x, y) \sim \pi} \left[\|x - y\|_2^2 \right] \right)^{1/2}, \tag{4.22}$$

where $\Pi(\mu, \nu)$ is the set of all couplings (possible joint distributions) between μ and ν . We have the following Theorem.

Theorem 7. *Following the same assumptions in Thm. 6, we have the asymptotic distribution of bagging inclusion probability estimation $\hat{\beta}^*$ and spike and slab model estimate $\tilde{\beta}_{SSL}$ satisfies*

$$\mathcal{W}_2^2(\mathcal{L}(\hat{\beta}^*), \mathcal{L}(\tilde{\beta}_{SSL})) \rightarrow 0. \tag{4.23}$$

Here, $\mathcal{W}_2^2(\cdot, \cdot)$ denotes the square of the Wasserstein-2 distance. The convergence in Wasserstein distance demonstrates that the Bayesian spike and slab distributional estimate is asymptotically equivalent to the bagging inclusion probability distributional estimate.

Another viewpoint is from the combination of Bayesian bootstrap Spike-and-Slab

Table 4.1: The computational difference in CPU time (seconds) for Bayesian UQ-SINDy [126] and Ensemble SINDy [90] in a Lotka–Volterra system.

	$n = 50$	$n = 100$	$n = 150$	$n = 250$
Bayesian	2.9×10^4	4.1×10^4	5.8×10^4	3.4×10^4
Ensemble	12.7	17.0	17.3	18.0

LASSO [228] using Expectation-Maximization Variable Selection [227, 282]. The Spike-and-Slab LASSO is a thresholding procedure. The threshold of index j is estimated by $\Delta_j = \inf_{t>0} \left(\|X_j\|^2 t / 2 - \sigma^2 \left(-\lambda_1 |t| + \log \left(\frac{p_\theta^*(0)}{p_\theta^*(t)} \right) \right) \right)$, where $p_\theta^*(t) = \frac{\theta \psi_1(t)}{\theta \psi_1(t) + (1-\theta) \psi_0(t)}$, $\psi_0(\cdot)$ denotes a spike distribution, and $\psi_1(\cdot)$ denotes a slab distribution. The TLS estimate approximates Δ_j by $\gamma \bar{\sigma}_{jj}$ which can be understood as a carefully constructed $\psi_0(\cdot), \psi_1(\cdot)$. The bagging-based is an approximation to the Bayesian bootstrap [231]. Therefore, the bootstrapping STLS is an approximation of a Bayesian bootstrap Spike-and-Slab LASSO process (BB-SSL). The BB-SSL can perform uncertainty quantification with a minimax-optimal concentration rate [207], which can also serve as a connection to the distributional estimate of the Bayesian.

A major drawback of the Bayesian Spike-and-slab prior comes from its computational requirement. We formulate the following Remark to state the computational difference.

Bayesian MCMC with Spike-and-slab model has computational complexity for each iteration that scales as $\Omega(n^2 p)$ [25]. Thus bagging inclusion probability has computational complexity for each bootstrap replicate of $\mathcal{O}(np^2)$.

Computational cost

For dynamical system identification problems under the $n > p$ regime, the computational complexity of bootstrap is lower compared to the Bayesian MCMC method. To obtain n MCMC samples from the Spike-and-slab model, the lower bound of computational complexity scales with $\Omega(n^3)$. Additionally, in theory, the mixing time of MCMC is generally dependent on the condition number R with $\tilde{\mathcal{O}}(n^2 R^2)$. Some recent efforts on Condition-number-independent rate will result in a complexity that scales with the sample size from $\tilde{\mathcal{O}}(n^2)$ to $\tilde{\mathcal{O}}(n^3)$ [154]. Alternatively, the bootstrap approximation only requires a computational upper bound $\mathcal{O}(n^2)$. In real experiments for physical model discovery, as shown in Table, 4.1, the expected running time of the spike-and-slab model can be much higher than n^3 due to the inflated mixing

time from the ill-conditioned design matrix. Typically, performing Bayesian UQ-SINDy (spike-and-slab) with 150 samples would require around 16 hours, which is over 3,000 times more expensive compared to the execution time of Ensemble SINDy (around 17 seconds). We note that in high-dimensional regimes with $p \gg n$, however, Bayesian spike-and-slab methods could be more favorable.

4.6 Experiments

In the following subsections, we apply bagging inclusion probability STLS to various cases, including synthetic linear regression and physical model discovery. We first investigate sparse linear regression under different data generation models. Compared to baseline methods, the bagging inclusion probability STLS outperforms other methods in terms of variable selection. Additionally, we study the robustness of hyperparameters in changing noise and sparsity levels. Bagging inclusion probability STLS has very stable performances with respect to the changing experimental settings while other methods fail to achieve stable performance. For governing equation discovery, we observe E-SINDy with bagging inclusion probability STLS can perform correct variable selection and uncertainty quantification. The estimation bias quickly reduces with more observation points and the uncertainty estimation is valid under noisy settings.

4.6.1 Implementation

There exists several variants of bagging-based STLS in practice. One strategy is to construct common statistics from the bootstrap samples, like the mean, median, and maximum. For example, we could consider the mean and variance of bootstrap samples to perform hypothesis testing. Also, we may utilize the maximum of the absolute value of all bootstrap replicates $\hat{\beta}^*$ with a threshold and select variables based on the inclusion probability. Constructing different statistics may have different advantages and drawbacks from case to case.

In the numerical simulation, we specifically utilize a re-weighted bagging inclusion probability selection using the out-of-bag (OOB) error which empirically achieves the best

results. Instead of naively putting each bootstrap sample with equal weights, re-weighting each sample based on its testing error helps to improve the performance of variable selection. There have been similar ideas in loss-guided stability selection [290] and cross-validation variable selection [244]. Intuitively, the mean squared error of a bootstrap replicate estimate should be low when the subset selection is perfect but can be very high when the subset is incorrect. Therefore, by adding the information from the testing loss, the bootstrap replicates can be utilized in a better way. This variant has the same theoretical guarantee as Thm. 5 in asymptotics. From our experimental study, re-weighting with OOB error can help to improve the performance experimentally. We demonstrate the implementation in Algorithm 2 which can be found in the Appendix.

4.6.2 Simulation of linear models: consistent variable selection

Simulation model setup

To compare fairly to the previous models, we first examine our model based on standard model generation as in [109, 135]. Then, we incorporate more challenging settings with very noisy environments. We also report a False discovery rate study following the setup from [258]. The implementation is based on Python with NumPy for STLS, bootstrap STLS, and sklearn for LASSO.

We consider the following models from $\mathbf{X} \in \mathbb{R}^{n \times p}$ with

$$\mathbf{y} = \mathbf{X}\beta + \epsilon, \quad (4.24)$$

where $\epsilon \sim \mathcal{N}(0, \sigma)$.

Model 1. [Linear regression, independent correlation structure] We set the data to $p = 30$, $\sigma = 1.0$, $X_i \sim \text{i.i.d. } \mathcal{N}(0, I_{30})$. We set the coefficient β to $\beta_{1:15} = 0.0$, $\beta_{16:30} = 1.0$.

Model 2. [Linear regression, mild toeplitz correlation structure] We set the data to $p = 30$, $\sigma = 0.6$, $X_i \sim \text{i.i.d. } \mathcal{N}(0, \Sigma)$ where $\Sigma_{ij} = r^{|i-j|}$, $r = 0.3$. We set the coefficient β to $\beta_{1:15} = 0.0$, $\beta_{16:20} = 0.5$, $\beta_{21:25} = 1.5$, $\beta_{25:30} = 2.5$.

Model 3. [Sparse regression for model discovery] We generate the design matrix X using

the following process. First generate $z = [z_1, z_2]$ where $z \sim \mathcal{N}(0, I_{2 \times 2})$. From z , construct $X = \Theta(z)$ such that

$$\Theta(z) = [1, z_1, z_2, z_1^2, z_1 z_2, z_2^2, z_1^3, z_1 z_2^2, z_1^2 z_2, z_2^3, \dots, z_2^4]^\top.$$

Finally, we use $\beta \in \mathbb{R}^{15 \times 2}$ to generate $y \in \mathbb{R}^{n \times 2}$ following (4.24). We study a synthetic Lotka-Volterra setting with $\beta[1, 0] = 1.0$, $\beta[5, 0] = -0.68$, $\beta[1, 1] = -1.5$, $\beta[5, 1] = 0.82$ and all other terms set to zero. Note that for this setting, we assume that the target (temporal derivatives) are known and we directly generate the temporal derivatives from the Lotka-Volterra equation.

Evaluation metrics We study the three simulation models from a basic linear regression, mild correlation structure [135], and sparse symbolic regression [32, 66, 241]. We access the performance of variable selection by the defined metric that follows.

We consider a classical metric in variable selection using the empirical relative frequency of correct identification. For index j , the relative frequency of correct identification is defined as

$$\hat{p}_j = \begin{cases} \frac{1}{S} \sum_{s=1}^S \mathbb{1} \{ \hat{\beta}_j^s = 0 \}, & j \in K_\beta \\ \frac{1}{S} \sum_{s=1}^S \mathbb{1} \{ \hat{\beta}_j^s \neq 0 \}, & j \notin K_\beta \end{cases}, \quad (4.25)$$

where S is the total number of simulations, $\hat{\beta}^s = (\hat{\beta}_1^s, \dots, \hat{\beta}_p^s)^\top$ is the estimation of s -th simulation.

Simulation outcomes

In Fig. 4.2, we show the simulation outcome of the three simulation models. All models are fine-tuned in the following process. We first select the optimal hyper-parameter of LASSO and STLS. Then, we perform the bootstrapping inclusion probability selection process based on the fine-tuned LASSO and STLS. Finally, we decide the inclusion probability threshold separately for bagging inclusion probability-based LASSO and bagging inclusion probability-based STLS. We replicate the experiments 200 times to compute the relative frequency.

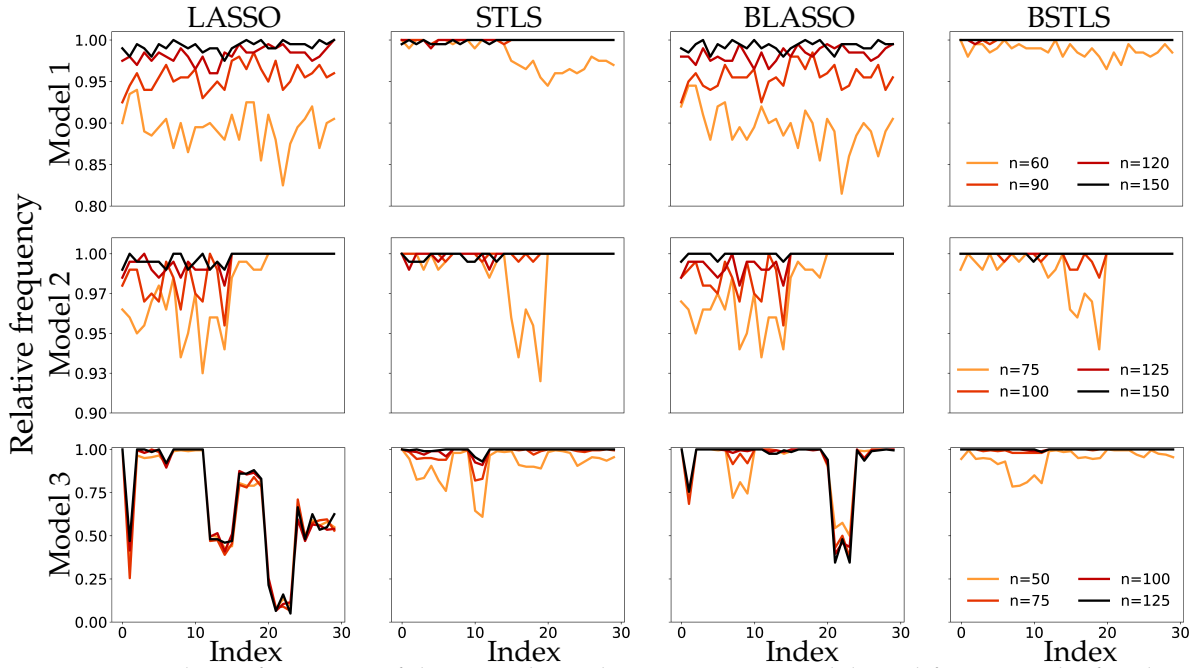


Figure 4.2: Relative frequency of three synthetic data generation models. Subfigures in the first line show the simulation outcome of Model 1; subfigures in the second line show the simulation outcome of Model 2; subfigures in the third line show the simulation outcome of Model 3. The BLASSO and BSTLS stand for bagging LASSO and bagging STLS.

Model 1 generates independent experimental trails with 1.0 signal-to-noise ratio. We set $\gamma = 0.15p \log(p)$ for STLS, $\lambda = 0.4$ as ℓ_1 constraint in LASSO. For both bagging inclusion probability-based LASSO and bagging inclusion probability-based STLS, we subsample 80% of data and set the inclusion probability threshold to be 0.45. Comparing LASSO and STLS, STLS has better performance with smaller sample size, but is relatively worse when the sample size is large. For Bagging inclusion probability selection, both bagging inclusion probability-based LASSO and bagging inclusion probability-based STLS improve the performance of LASSO and STLS respectively. bagging inclusion probability-based STLS outperforms all other methods for all sample sizes.

Model 2 generates design with mild correlation structure. We set $\gamma = 0.1p \log(p)$ for STLS, $\lambda = 0.2$ as ℓ_1 constraint in LASSO. For both bagging inclusion probability-based LASSO and bagging inclusion probability-based STLS, we subsample 80% of data and set the inclusion probability threshold to be 0.7. STLS has better performances with increasing number of samples compared to LASSO. When $n = 60$, notice the minimum of empirical relative frequency of STLS is 98% while LASSO could only reach around 85%. The sample complexity

of LASSO in this case is clearly higher compared to STLS since LASSO still fails to robustly perform correct identification when $n = 150$. The poor performance of LASSO affects the bagging inclusion probability-based LASSO estimator, which even underperforms STLS. For bagging inclusion probability-based STLS, it has similar performance compared to STLS when the sample size is small, but performs much better with larger number of samples. When $n = 150$, bagging inclusion probability-based STLS performs perfect variable selection.

Model 3 generates a design matrix from the sparse physics model discovery setting. We set $\gamma = 50p \log(p)$ as a constant threshold for STLS, $\lambda = 0.5$ for ℓ_1 constraint in LASSO. For both bagging inclusion probability-based LASSO and bagging inclusion probability-based STLS, we subsample 50% of data and set the inclusion probability threshold to be 0.8. In this setting, the terms in X are dependent on the latent variables. This dependence structure violates many statistical conditions which are challenging for LASSO based estimators. In Fig. 4.2 (i), (k), the LASSO based estimate of $\beta_{0,0}$ is consistently incorrect with ℓ_1 constraints ranging from $[0.01, 100.0]$. For this setup, LASSO fails to identify any constant term from symbolic regression. Even if the bagging inclusion probability-based method helps LASSO to improve the relative frequency for the other terms, but the identification of $\beta_{0,0}$ is still incorrect. By contrast, STLS has good performances with increasing number of samples, and bagging inclusion probability-based STLS significantly improves from STLS. Bagging inclusion probability-based STLS behaves the best among these methods.

Robustness of hyper-parameters in varying sparse linear regression settings

In practice, one may not have any prior knowledge about the sparsity level of a specific system. We exemplify the idea of hyperparameter robustness in this experiment. For example in Model 1, we only know $p = 30$, while q can be an arbitrary value in the range $[0, 30]$. However, in non-asymptotic regimes, the LASSO type of variable selection methods are very sensitive to the choices of hyper-parameters. When q is small (e.g. $q = 1$), a large ℓ_1 constraint is preferred; while, in contrast when $q \approx p$, a small ℓ_1 constraint is preferred. It is important to understand if the hyperparameter setting can be generalized to practical experiments.

In the following, we perform the experiments similar to Model 1 with: (a) varying

noise levels that $q = 15, \sigma = [0.25, 0.5, 0.75, 1.0]$ and (b) varying number of active terms $\sigma = 0.5, q = [1, 5, 10, 15]$. Different from many previous studies, we fix the hyperparameter setting, and understand how robust the hyperparameter is given the changes. We consistently fine tune threshold for all methods to obtain good empirical True discovery probability (ETDP), and compare their performances for empirical false discovery probability (EFDP).

As shown in Fig. 4.3 and Fig. 4.4, we notice bagging inclusion probability-based STLS is very robust to experimental settings. For both changing number of non-zero components and noise level, bagging inclusion probability-based STLS performs similar in the control of false discovery probability. Even if bagging inclusion probability-based STLS behaves slightly different in the empirical true discovery probability, the difference quickly vanishes after having relatively larger sample sizes (around $n = 100$). STLS is also robust to changing experimental setting with relative worse performance comparing to bagging inclusion probability-based methods. For example, in Fig. 4.3 and Fig. 4.4, the empirical FDP of bagging inclusion probability-based methods perform similarly for both small noise (non-active terms) and large noise (non-active terms).

LASSO-based methods all suffer from these changes that the false discovery controls perform very differently among these settings. Even if bagging inclusion probability-based LASSO could help to control false discoveries comparing to LASSO, we could observe the false discovery curves behave very differently among these settings. For example, in Fig. 4.3, bagging inclusion probability-based LASSO performs well when q is large, but is relatively worse with higher sparsity rate. Similarly, in Fig. 4.4, bagging inclusion probability-based LASSO performs well when $\sigma = 0.25$, and behaves progressively worse with increasing noise levels. Furthermore, unlike STLS and bagging inclusion probability-based STLS, the difference does not vanish even with relatively large sample sizes ($n = 250$).

4.6.3 Lotka–Volterra model discovery via Ensemble SINDy

The Lotka-Volterra model is frequently used to simulate the interaction between two competing groups u and v like predator-prey, chemical reactions, and economics. The model can be

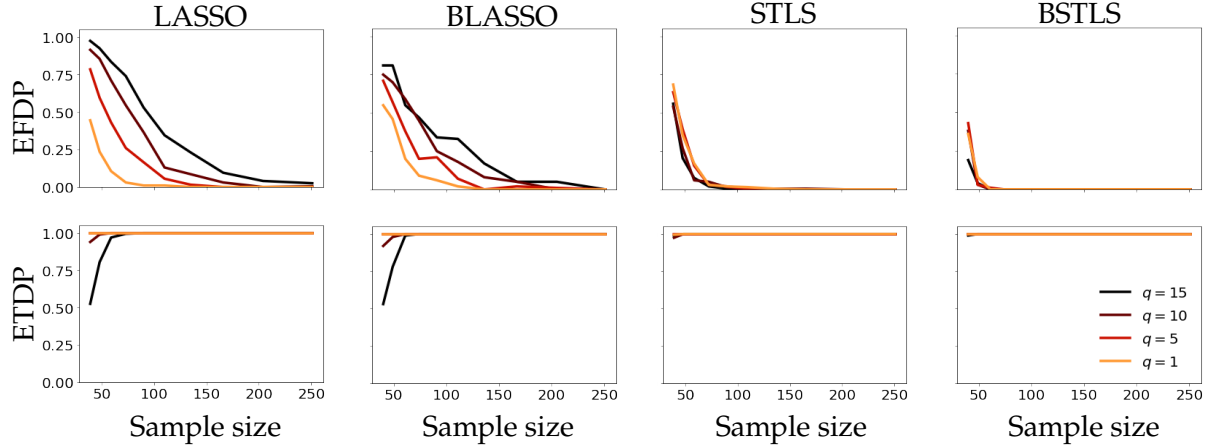


Figure 4.3: Empirical false discovery probability and success true discovery probability with increasing number of samples under **different sparsity levels**. For better comparison, the hyperparameters of all methods are fine tuned for STDP that all non-zero coefficients are identified. We compare each method's performance via false discovery control. The BLASSO and BSTLS stand for bagging LASSO and bagging STLS. On the y-axis, EFDP represents the empirical FDP (Eqn. (4.11)) and ETDP represents the empirical TDP (Eqn. (4.12)).

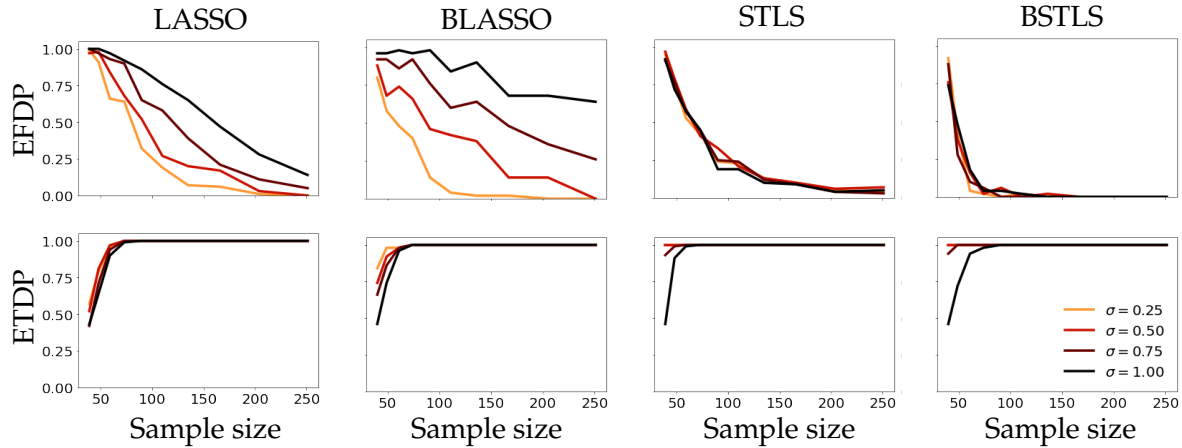


Figure 4.4: Empirical false discovery probability and success true discovery probability with increasing number of samples under **various levels of noises**. For better comparison, the hyperparameters of all methods are tuned to control the worse case of ETDP is around 50%. We compare each method's performance via false discovery control. The BLASSO and BSTLS stand for bagging LASSO and bagging STLS. On the y-axis, EFDP represents the empirical FDP (Eqn. 4.11) and ETDP represents the empirical TDP (Eqn. 4.12).

defined using the following set of differential equations:

$$\dot{u} = \alpha u + \beta uv, \quad (4.26)$$

$$\dot{v} = \gamma v + \delta uv, \quad (4.27)$$

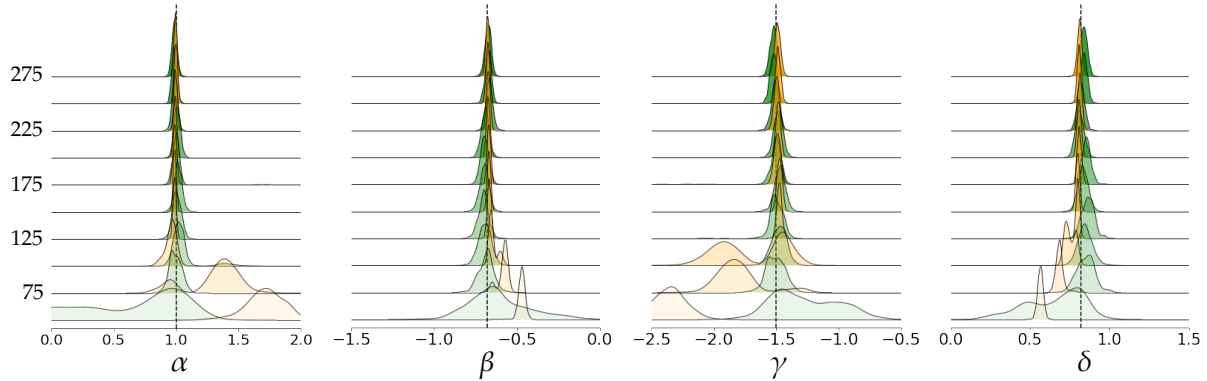


Figure 4.5: Distributional approximation with uncertainty quantification of Bayesian SINDy (green-colored) and E-SINDy (orange-colored) to the ground-truth value (dotted line) for all active terms.

where $\alpha = 1.0, \beta = -0.1, \gamma = -1.5, \delta = 0.075$. We set the initial conditions to $[u_0, v_0] = [10, 5]$ with $\text{Lognormal}(0, 0.1)$ noise. The number of measurement data are uniformly sampled from $t \in [0, 24]$ with $n = [100, 200, 300, 400, 500]$. We perform data normalization the measurement data via $u/\hat{\sigma}(u)$ and $v/\hat{\sigma}(v)$ where $\hat{\sigma}(\cdot)$ is the empirical standard deviation estimate of the trajectories respectively. After normalization, the governing differential equation will have a different set of parameters $\tilde{\alpha} = 1.0, \tilde{\beta} = -0.68, \tilde{\gamma} = -1.5, \tilde{\delta} = 0.82$. Compared to the synthetic simulation in 4.6.2, the study in this subsection generates a time-series data from the initial condition, and the target $\dot{\mathbf{X}}$ is computed from the time-series data via a smoothed finite difference method. As a result, these data samples are no longer independent and the true targets are no longer directly accessible. This setting closely mimics the data collected by real-world sensors, and the temporal dependency structure presents a particularly challenging aspect.

We visualize the distributional evolution for the four active coefficient estimates using E-SINDy and Bayesian SINDy [126] in Fig. 4.5. Appendix (Sec. Additional plottings) includes the distributional evolution for all non-active coefficients. For all four active indices $(\tilde{\alpha}u, \tilde{\beta}uv, \tilde{\gamma}v, \tilde{\delta}uv)$, we observe the distributional approximation of E-SINDy would first transit from a wrong distribution (with biased mean, high variance) to a delta function concentrating around the true value. This observed behavior of E-SINDy makes it a suitable method for uncertainty quantification. While Bayesian SINDy can be computationally expensive, as shown in Fig. 4.5, it produces nearly unbiased distributional estimates even with

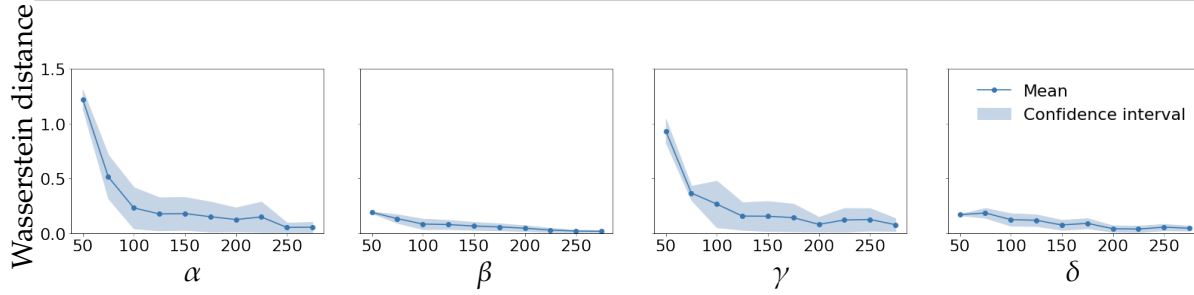


Figure 4.6: Convergence in Wasserstein-2 distance between E-SINDy and the Bayesian SINDy estimates for all four active indices. Shaded regions denote 95% confidence intervals.

very little data, which is remarkable for UQ purposes. We further examine the convergence in Wasserstein between Bayesian SINDy and E-SINDy for all four active indices in Fig. 4.6. For all indices, the Wasserstein distance converges quickly towards zero with larger number of samples.

4.7 Conclusion and discussion

In this paper, we introduce the bagging inclusion probability-based sequentially thresholding least-squares estimation and establish the analysis of its statistical properties on variable selection as well as uncertainty quantification. In particular, we achieve our proof within the context of dynamical system discovery, which involves dependent rows and features. From one perspective, for variable selection, bagging inclusion probability-based STLS has oracle property as stated in Thm. 5. The bagging-based STLS requires weaker condition compared to STLS in theory to show the convergence of FDP and TDP. We study and compare these two algorithms in Sec. 4.6, which shows accelerated variable selection in various case studies. We note here that the error bound shown in this manuscript is not tight. This is because we pay linearly with increasing model sizes p in the empirical process with maximal of n Binomial trials using union bounds. Experimentally, we observe that the errors grow sublinearly with p .

From another perspective, bagging inclusion probability-based STLS also has a theoretical guarantee on variance estimation. We achieve asymptotic normality only requiring zero mean and finite variance assumptions. As shown in Thm. 6, we can use bootstrap replicates to provide approximate distributional estimation. In the experimental study with governing

equation discovery (Sec. 4.6.3), we visualize the distributional estimation in Fig. 4.5 to understand the behavior of uncertainty quantification of the bagging inclusion probability-based STLS estimator. The bagging-based STLS also has a close connection to Bayesian sparse regression methods via MCMC. Based on the rich history of understanding bootstrapping procedure as an approximate Bayesian inference [204, 208, 231], it is possible to understand bagging inclusion probability-based STLS as an approximate Bayesian sparse method. We show their equivalence by showing these two methods are asymptotically equivalent in Thm. 7, yet Bayesian sparse inference via spike-and-slab prior has a very high computational requirement.

There are many interesting future extensions of our work. Given the computational efficiency and accuracy of ensemble-based sparse regression methods, an avenue of ongoing research is to apply the technique to the broader class of sparse optimization problems like matrix completion and compressive sensing [320, 39, 40, 41]. These thresholding-based algorithms could also be applied in more modern machine learning problems with dependent-design including transfer learning, out-of-distribution generalization, and sparse neural network training. Second, it will be interesting to study the behavior of bootstrap sparse regression in high-dimensional settings [147, 61]. Fasel et al. [90] empirically show that the E-SINDy algorithm with bagging inclusion probability is robust in high-dimensional settings for the discovery of PDE. This observation suggests the bootstrapping-based sparse regression method could mitigate some challenges reported in high dimensional settings, where it is generally difficult to have robust bootstrapping techniques [193, 7, 51]. It is also important to include better subset selection [2] and post-selection inference methods to debias the uncertainty estimates [166]. Finally, it will be exciting to extend the established theory to the different variants of ensembling techniques in E-SINDy [145]. It is interesting to study the explicit noise distribution amplified by SINDy library terms from the observation, and understand how empirical bootstrap could quantify uncertainty for this type of noise. This will be useful for practitioners to gain a better understanding of the different ensembling techniques in E-SINDy and their performance and applicability to different sparse model discovery settings.

Proof of Lemma 6

Recall that ρ_1 is the minimum eigenvalue of $\frac{1}{n}\mathbf{X}^\top\mathbf{X}$. We introduce the following regularity conditions.

$$(3.1.a) \mathbb{E}[\epsilon] = 0, \text{var}(\epsilon) = \sigma^2 < \infty, \text{ and } \mathbb{E}[\epsilon^4] < \infty \text{ where } \epsilon \stackrel{\text{i.i.d.}}{\sim} P.$$

$$(3.1.b) \rho_1 > 0, \max_i(\mathbf{X}_i^\top\mathbf{X}_i) = O(p), \text{ and } p/(n\rho_1) = o(1).$$

Proof. We first show Eqn. (4.5). We could see that $\bar{\beta}$ is an unbiased estimate that

$$\mathbb{E}_\epsilon [\mathbf{a}^\top(\bar{\beta} - \beta)] = \mathbb{E}_\epsilon [\mathbf{a}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\epsilon] = 0. \quad (4.28)$$

We write

$$\mathbf{a}^\top(\bar{\beta} - \beta) = \mathbf{a}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\epsilon = \sum_{i=1}^n \alpha_i \epsilon_i, \quad (4.29)$$

where $\alpha_i = \mathbf{a}^\top(\mathbf{X}^\top\mathbf{X})^{-1}(\mathbf{X}^\top)_i$.

Showing Eqn. (4.5) is equivalent to show $\sum_{i=1}^n \alpha_i \epsilon_i \rightarrow \mathcal{N}(0, \sigma^2 \mathbf{a}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{a})$.

Due to Eqn. (4.28), we see $\mathbb{E}[\sum_{i=1}^n \alpha_i \epsilon_i] = 0$.

Consider the variance of $\alpha_i \epsilon_i$,

$$\text{var}(\alpha_i \epsilon_i) = \sigma^2 \mathbf{a}^\top(\mathbf{X}^\top\mathbf{X})^{-1}(\mathbf{X}^\top)_i(\mathbf{X}^\top)_i^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{a} \quad (4.30)$$

We have

$$\sum_{i=1}^n \text{var}(\alpha_i \epsilon_i) = \sigma^2 \mathbf{a}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{a} \quad (4.31)$$

By Lindeberg-Feller Central Limit Theorem (Lemma 9), we can show the weak convergence to normal distribution. To apply Lindeberg-Feller CLT, we need to verify the Lindeberg

condition that

$$\sum_{i=1}^n \mathbb{E} \left[|\alpha_i \epsilon_i|^2 \mathbb{1}_{\{|\alpha_i \epsilon_i| \geq \delta\}} \right] \rightarrow 0, \quad \forall \delta > 0. \quad (4.32)$$

We know

$$\sum_{i=1}^n \mathbb{E} \left[|\alpha_i \epsilon_i|^2 \mathbb{1}_{\{|\alpha_i \epsilon_i| \geq \delta\}} \right] \leq \sum_{i=1}^n \mathbb{E} \left[\alpha_i^2 \epsilon_i^2 \right] \quad (4.33)$$

$$= \sigma^2 \mathbf{a}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{a} \quad (4.34)$$

$$\leq \max_{\|\mathbf{a}\|=1} \sigma^2 \mathbf{a}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{a} \quad (4.35)$$

$$= \frac{\sigma^2}{n \rho_1}, \quad (4.36)$$

which is $o(1)$. We could see the connection between $\max_{\|\mathbf{a}\|_2=1} \sigma^2 \mathbf{a}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{a}$ by doing matrix diagonalization of $(\mathbf{X}^\top \mathbf{X})^{-1}$.

Since Lindeberg condition is satisfied, using Lindeberg-Feller CLT and Slutsky's theorem, we can conclude Eqn. (4.5) that

$$\mathbf{a}^\top (\bar{\beta} - \beta) \xrightarrow{d} \mathcal{N}(0, \sigma^2 \mathbf{a}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{a}). \quad (4.37)$$

Define the quantity $Z = \mathbf{a}^\top (\hat{\beta} - \beta)$, and notice here Z is only an one-dimensional random variable. Since Z is asymptotic normal, therefore Z is also $\text{SubG}(0, \sigma^2 \mathbf{a}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{a})$ in asymptotic. Therefore, by Lemma 11 (Eqn. (2.9) in [278]), we see

$$\mathbb{P}(Z \geq t) \leq \exp\left(-\frac{t^2}{2\sigma^2 \mathbf{a}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{a}}\right) \quad (4.38)$$

By setting $\delta = \exp\left(-\frac{t^2}{2\sigma^2 \mathbf{a}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{a}}\right)$, we see $t = \sqrt{2\sigma^2 \mathbf{a}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{a} \log(1/\delta)}$. This is the

high-probability form

$$\mathbb{P}\left(\mathbf{a}^\top(\hat{\beta} - \beta) \geq \sqrt{2\sigma^2 \mathbf{a}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{a} \log(1/\delta)}\right) \leq \delta \quad (4.39)$$

as in Eqn. (4.6). $\square$

Proof of Lemma 7

Say q is the number of non-active components in β (equivalently $|K_\beta|$), n is the number of sample, and p is the model size. We require the following regularity conditions to show this Lemma.

$$(3.2.a) \ \epsilon \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2 \mathbf{I}_n).$$

$$(3.2.b) \ \frac{q}{n} = o(1).$$

$$(3.2.c) \ \frac{\rho_1 \min_{i \in J} |\beta_i|}{\sigma^2} = \omega\left(\frac{\log(n)}{n}\right).$$

Condition (3.2.b) requires the sample size goes to infinity. Condition (3.2.c) allows the lowest eigenvalue of $\frac{1}{n} \mathbf{X}^\top \mathbf{X}$ to go to zero slower than $\frac{\log(n)}{n}$, which fits into the context of dynamical system.

Proof. Recall that we let K_β be the non-active indices, and we define J_β to be the active indices. Let $\hat{K}$ be the thresholding estimator of the set of non-active indices K_β , and let $\hat{K}_{STLS}$ to specify estimate of K_β using the STLS.

To show $\mathbb{P}(\hat{K} = K_\beta) = 1$, we first consider the event $\mathbb{P}(K_\beta \subseteq \hat{K})$, which corresponds to the probability that no inactive variables are falsely selected. By setting $\delta = \frac{1}{n}$, we can bound the probability of false discovery as follows:

$$\mathbb{P}\left(\max_{j \in K_\beta} \left(|\tilde{\beta}_j| - \sqrt{2\sigma^2 \log(1/\delta) \cdot (\mathbf{X}^\top \mathbf{X})_{jj}^{-1}}\right) \geq 0\right) \quad (4.40)$$

$$\leq \sum_{j \in K_\beta} \mathbb{P}\left(|\tilde{\beta}_j| - \sqrt{2\sigma^2 \log(1/\delta) \cdot (\mathbf{X}^\top \mathbf{X})_{jj}^{-1}} \geq 0\right) \quad (4.41)$$

$$\leq |K_\beta| \cdot \delta \leq \frac{q}{n}. \quad (4.42)$$

Here, $(\mathbf{X}^\top \mathbf{X})_{jj}$ denotes the j -th diagonal entry of the Gram matrix $\mathbf{X}^\top \mathbf{X}$, and $(\mathbf{X}^\top \mathbf{X})_{jj}^{-1}$ is the corresponding diagonal entry of the inverse covariance matrix for the OLS estimate $\bar{\beta}_j$. We apply a Gaussian tail bound to each $\bar{\beta}_j$, assuming independence and sub-Gaussianity. For simplicity, we conservatively upper bound $|K_\beta| \leq q$ in sparse regimes.

The above inequality helps to show Lemma 7(i), as it bounds the complement of the event

$$\mathbb{P}\left(K_\beta \subset \hat{K}_{\text{STLS}}\right) = 1 - \mathbb{P}\left(\max_{j \in K_\beta} \left(|\bar{\beta}_j| - \sqrt{2\sigma^2 \log(1/\delta) \cdot (\mathbf{X}^\top \mathbf{X})_{jj}^{-1}}\right) \geq 0\right). \quad (4.43)$$

This probability approaches one as $q/n \rightarrow 0$ (3.2.b), justifying the desired property in the sparse regime.

From (3.2.b) we know $\frac{q}{n} = o(1)$. Therefore,

$$\mathbb{P}\left(|\bar{\beta}_j| \leq \sqrt{2\sigma^2 \log(1/\delta) (\mathbf{X}^\top \mathbf{X})_{jj}^{-1}} \text{ for all } j \in K_\beta\right) \rightarrow 1. \quad (4.44)$$

$$\mathbb{P}\left(K_\beta \subseteq \hat{K}\right) \rightarrow 1. \quad (4.45)$$

From the other side, we could upper bound the probability that the minimum least-squares estimate of an active index will below the threshold that

$$\mathbb{P}\left(\min_{j \in J_\beta} \left(|\bar{\beta}_j| - \sqrt{2\sigma^2 \log(1/\delta) (\mathbf{X}^\top \mathbf{X})_{jj}^{-1}}\right) \leq 0\right) \quad (4.46)$$

$$\leq \mathbb{P}\left(\min_{j \in J_\beta} \left(|\beta_j| - |\bar{\beta}_j - \beta_j| - \sqrt{2\sigma^2 \log(1/\delta) (\mathbf{X}^\top \mathbf{X})_{jj}^{-1}}\right) \leq 0\right) \quad (4.47)$$

$$\leq \mathbb{P}\left(\max_{j \in J_\beta} \left(|\bar{\beta}_j - \beta_j| + \sqrt{2\sigma^2 \log(1/\delta) (\mathbf{X}^\top \mathbf{X})_{jj}^{-1}}\right) \geq \min_{i \in J_\beta} |\beta_i|\right) \quad (4.48)$$

$$\leq 2\mathbb{P}\left(\max_{j \in J_\beta} \left(|\bar{\beta}_j - \beta_j| + \sqrt{2\sigma^2 \log(1/\delta) (\mathbf{X}^\top \mathbf{X})_{jj}^{-1}}\right) \geq \min_{i \in J_\beta} |\beta_i|\right). \quad (4.49)$$

Define $G_j = \frac{\bar{\beta}_j - \beta_j}{\sigma \sqrt{(\mathbf{X}^\top \mathbf{X})_{jj}^{-1}}}$, we know $G_j \sim \mathcal{N}(0, 1)$. Therefore,

$$\mathbb{P} \left(\max_{j \in J_\beta} \left((\bar{\beta}_j - \beta_j) + \sqrt{2\sigma^2 \log(1/\delta)(\mathbf{X}^\top \mathbf{X})_{jj}^{-1}} \right) \geq \min_{i \in J_\beta} |\beta_i| \right) \quad (4.50)$$

$$= \mathbb{P} \left(\max_{j \in J_\beta} G_j \sigma \sqrt{(\mathbf{X}^\top \mathbf{X})_{jj}^{-1}} \geq \min_{i \in J_\beta} |\beta_i| - \sqrt{2\sigma^2 \log(1/\delta)(\mathbf{X}^\top \mathbf{X})_{jj}^{-1}} \right) \quad (4.51)$$

$$\leq \mathbb{P} \left(\max_{j \in J_\beta} G_i \geq \frac{n^{1/2} \rho_1^{1/2} \min_{i \in J_\beta} |\beta_i|}{\sigma} - \sqrt{2 \log(1/\delta)} \right). \quad (4.52)$$

Using the maximal inequality of sub-Gaussian random variables, we see

$$\mathbb{P} \left(\max_{j \in J_\beta} G_i \geq \frac{n^{1/2} \rho_1^{1/2} \min_{i \in J_\beta} |\beta_i|}{\sigma} - \sqrt{2 \log(1/\delta)} \right) \quad (4.53)$$

$$\leq q \exp \left(-\frac{1}{2} \left(\frac{n^{1/2} \rho_1^{1/2} \min_{i \in J_\beta} |\beta_i|}{\sigma} - \sqrt{2 \log(1/\delta)} \right)^2 \right). \quad (4.54)$$

Summarizing all the terms and letting $\delta = \frac{1}{n}$, we know

$$\mathbb{P} \left(\min_{j \in J_\beta} \left(|\bar{\beta}_j| - \sqrt{2\sigma^2 \log(1/\delta)(\mathbf{X}^\top \mathbf{X})_{jj}^{-1}} \right) \leq 0 \right) \quad (4.55)$$

$$\leq 2 \exp \left(-\frac{1}{2} \left(\left(\frac{n^{1/2} \rho_1^{1/2} \min_{i \in J_\beta} |\beta_i|}{\sigma} - \sqrt{2 \log(n)} \right)^2 - 2 \log(q) \right) \right). \quad (4.56)$$

Therefore, we obtain the following probability bound:

$$\mathbb{P} \left(\hat{K}_{\text{STLS}} \subset K_\beta \right) \geq 1 - 2 \exp \left(-\frac{1}{2} \left(\left(\frac{n^{1/2} \rho_1^{1/2} \min_{i \in J_\beta} |\beta_i|}{\sigma} - \sqrt{2 \log n} \right)^2 - 2 \log(q) \right) \right), \quad (4.57)$$

which supports the high-probability conclusion in Lemma 7(ii).

Under assumptions (3.2.b–c), we have $n \gg q$ and $\rho_1 \gtrsim \frac{\log n}{n}$. Therefore, the quantity

$$\left(\frac{n^{1/2} \rho_1^{1/2} \min_{i \in J_\beta} |\beta_i|}{\sigma} - \sqrt{2 \log n} \right)^2 - 2 \log q \rightarrow \infty, \quad (4.58)$$

which implies that the probability

$$\mathbb{P} \left(\min_{j \in J_\beta} |\bar{\beta}_j| \leq \sqrt{2\sigma^2 \log n \cdot (\mathbf{X}^\top \mathbf{X})_{jj}^{-1}} \right) \rightarrow 0. \quad (4.59)$$

From the variable selection consistency and the nature of the algorithm (least-squares after thresholding), we also have the limiting distribution, the same as if the sparsity pattern is known before inference. Combining all these steps, we have shown that this estimator has the oracle property.

□

Proof of Theorem 5

We require the following conditions.

$$(4.1.a) \ \epsilon \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2 \mathbf{I}_n).$$

$$(4.1.b) \ \frac{q}{p} = O(1), \ \frac{p}{\exp(n)} = o(1).$$

(4.1.c) For any bootstrap replicate $\mathbf{X}^b$, the lowest eigenvalue of $\frac{1}{n} \mathbf{X}^{b\top} \mathbf{X}^b$ is lower bounded by $r_0 \rho_1$ where $r_0 \in \mathbb{R}$ is a constant.

$$(4.1.d) \ \rho_1 > 0, \text{ and } \frac{\rho_1 \min_{i \in J} |\beta_i|}{\sigma^2} = \omega \left(\frac{\log(n)}{n} \right).$$

Proof. **Part (i)**

First, we know $\bar{\beta}_j^b$ is just the Bootstrap replicate of OLS estimator. From the assumption of the tail behavior, we see that $\bar{\beta}$ follows

$$\bar{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top y \quad (4.60)$$

$$= (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top (\beta \mathbf{X} + \epsilon) \quad (4.61)$$

$$= \beta + (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \epsilon \quad (4.62)$$

$$\sim \mathcal{N}(\beta, \sigma^2 (\mathbf{X}^\top \mathbf{X})^{-1}). \quad (4.63)$$

Again, recall that we let K_β be the non-active indices, and we define J_β to be the active indices.

Therefore, from the sub-Gaussian tail behavior, by taking a as the standard basis with index j , there exists some $\delta > 0$ that

$$\mathbb{P} \left(|\bar{\beta}_j - \beta_j| > \sigma \sqrt{2 \log(1/\delta) (\mathbf{X}^\top \mathbf{X})_{jj}^{-1}} \right) \leq \delta. \quad (4.64)$$

Similarly from the sub-Gaussian tail, we could see that there exists some other $\delta > 0$ that

$$\mathbb{P} \left(|\bar{\beta}_j^b - \beta_j| > \sigma \sqrt{2 \log(1/\delta) (\mathbf{X}^{b\top} \mathbf{X}^b)_{jj}^{-1}} \right) \leq \delta. \quad (4.65)$$

The quantity $\mathbb{1} \left\{ |\bar{\beta}_j^b| > \sigma \sqrt{2 \log(1/\delta) (\mathbf{X}^{b\top} \mathbf{X}^b)_{jj}^{-1}} \right\}$ only takes $\{0, 1\}$ which is a Bernoulli random variable. Therefore, we can consider a Bernoulli process $B_j^b \sim \text{Ber}(\delta)$. We know that $\mathbf{B}_j = \sum_{b=1}^n B_j^b \sim \text{Binomial}(n, \delta)$. We can compute

$$\mathbb{P} \left(\max_{j \in K_\beta} \frac{1}{n} \sum_{b=1}^n \mathbb{1} \left\{ |\bar{\beta}_j^b| > \sigma \sqrt{2 \log(1/\delta) (\mathbf{X}^{b\top} \mathbf{X}^b)_{jj}^{-1}} \right\} > p_c \right) \quad (4.66)$$

$$\leq \mathbb{P} \left(\max_{j \in K_\beta} \sum_{b=1}^n B_j^b > np_c \right) \quad (4.67)$$

$$= \mathbb{P} \left(\max_{j \in K_\beta} \mathbf{B}_j > np_c \right) \quad (4.68)$$

$$= \mathbb{P} \left(\bigcup_{j \in K_\beta} (\mathbf{B}_j > np_c) \right) \quad (4.69)$$

$$\leq (p - q) \mathbb{P} (\mathbf{B}_1 > np_c). \quad (4.70)$$

Here we use $\mathbf{B}_1$ because for any $j \in K_\beta$, $\mathbf{B}_j$ is the same after applying the Union bound.

From Eqn. (4.66), we cast the original problem into an empirical process with Binomial trails. We apply the **Chernoff bound** in the following.

Since $B_j^b \in [0, 1]$, from Chernoff Upper Tail, and suppose $p_c \geq \delta$, we know

$$\mathbb{P}(\mathbf{B}_j \geq np_c) \leq \exp\left(-\frac{n\delta(p_c/\delta - 1)}{3}\right). \quad (4.71)$$

By setting $\delta = \frac{1}{n}$,

$$\mathbb{P}(\mathbf{B}_j \geq np_c) \leq \exp\left(\frac{1}{3} - \frac{np_c}{3}\right). \quad (4.72)$$

It is also possible to apply **Hoeffding bound** which obtains a similar rate of convergence.

The inequalities above illustrates (i) in Thm. 5 because

$$\mathbb{P}(K_\beta \subset \hat{K}_{BIP}) = 1 - \mathbb{P}\left(\max_{j \in K_\beta} \frac{1}{n} \sum_{b=1}^n \mathbb{1}\left\{|\bar{\beta}_j^b| > \sigma \sqrt{2 \log(1/\delta)(\mathbf{X}^{b\top} \mathbf{X}^b)^{-1}_{jj}}\right\} > p_c\right) \quad (4.73)$$

Part (ii)

Without loss of generality, we set $|\beta_1| = \min_{j \in J_\beta} |\beta_j|$ is the smallest.

$$\mathbb{P}\left(\min_{j \in J_\beta} \frac{1}{n} \sum_{b=1}^n \mathbb{1}\left\{|\bar{\beta}_j^b| > \sigma \sqrt{2 \log(1/\delta)(\mathbf{X}^{b\top} \mathbf{X}^b)^{-1}_{jj}}\right\} < p_c\right) \quad (4.74)$$

$$\leq \mathbb{P}\left(\bigcup_{j \in J_\beta} \left(\frac{1}{n} \sum_{b=1}^n \mathbb{1}\left\{|\bar{\beta}_j^b| > \sigma \sqrt{2 \log(1/\delta)(\mathbf{X}^{b\top} \mathbf{X}^b)^{-1}_{jj}}\right\} < p_c\right)\right) \quad (4.75)$$

$$\leq q \cdot \mathbb{P}\left(\frac{1}{n} \sum_{b=1}^n \mathbb{1}\left\{|\bar{\beta}_j^b| > \sigma \sqrt{2 \log(1/\delta)(\mathbf{X}^{b\top} \mathbf{X}^b)^{-1}_{jj}}\right\} < p_c\right) \quad (4.76)$$

$$\leq q \cdot \mathbb{P}\left(\frac{1}{n} \sum_{b=1}^n \mathbb{1}\left\{\frac{|\bar{\beta}_j^b|}{\sigma \sqrt{(\mathbf{X}^{b\top} \mathbf{X}^b)^{-1}_{jj}}} > \sqrt{2 \log(1/\delta)}\right\} < p_c\right). \quad (4.77)$$

Consider the quantity on the left hand side $\frac{|\bar{\beta}_j^b|}{\sigma \sqrt{(\mathbf{X}^{b\top} \mathbf{X}^b)^{-1}_{jj}}}$. We know that the bootstrap replicate $\bar{\beta}_j^b \sim \mathcal{N}\left(\beta_j, \sigma \sqrt{(\mathbf{X}^{b\top} \mathbf{X}^b)^{-1}_{jj}}\right)$. Without loss of generality, we assume this $\bar{\beta}_j^b > 0$.

Denote the CDF of standard normal distribution to be $\Phi(\cdot)$. For each bootstrap replicate,

$$\mathbb{P}\left(|\tilde{\beta}_j^b| > \sqrt{2\log(1/\delta)}\right) = \mathbb{P}\left(\tilde{\beta}_j^b > \sqrt{2\log(1/\delta)}\right) + \mathbb{P}\left(-\tilde{\beta}_j^b > \sqrt{2\log(1/\delta)}\right) \quad (4.78)$$

$$\geq \mathbb{P}\left(\tilde{\beta}_j^b > \sqrt{2\log(1/\delta)}\right) \quad [\text{ignoring the lower tail from the W.L.O.G.}] \quad (4.79)$$

$$= \Phi\left(\frac{|\beta_j|}{\sigma\sqrt{(\mathbf{X}^{b\top}\mathbf{X}^b)^{-1}_{jj}}} - \sqrt{2\log(1/\delta)}\right) \quad (4.80)$$

$$\text{from (4.1.c)} \geq \Phi\left(\frac{\sqrt{n}|\beta_j|}{\sigma\sqrt{(r_0\rho_1)^{-1}}} - \sqrt{2\log(1/\delta)}\right) \quad (4.81)$$

$$\geq \Phi\left(\frac{\sqrt{nr_0\rho_1}\min_{i\in J_\beta}|\beta_i|}{\sigma} - \sqrt{2\log(1/\delta)}\right) := p_{j_{\min}} \quad (4.82)$$

Therefore, from (4.1.d), we know $\rho_1 > 0$. Therefore, we can always choose a constant threshold $p_c < \min_{j\in J_\beta} \Phi\left(\frac{\sqrt{nr_0\rho_1}|\beta_i|}{\sigma} - \sqrt{2\log(1/\delta)}\right)$.

Then, we could set $p_{j_{\min}} = \min_{j\in J_\beta} \Phi\left(\frac{\sqrt{nr_0\rho_1}|\beta_i|}{\sigma} - \sqrt{2\log(1/\delta)}\right)$, and treat $B_j^b \sim \text{Bernoulli}(p_{j_{\min}})$. We know that $\mathbf{B}_j = \sum_{b=1}^n B_j^b \sim \text{Binomial}(n, p_{j_{\min}})$. We can compute using Chernoff Lower Tail

$$\mathbb{P}(\mathbf{B}_j \leq np_c) \leq \exp\left(-\frac{np_{j_{\min}}(1 - p_c/p_{j_{\min}})^2}{3}\right) \quad (4.83)$$

$$= \exp\left(-\frac{np_c^2}{3p_{j_{\min}}} + \frac{2np_c}{3} - \frac{np_{j_{\min}}}{3}\right) \quad (4.84)$$

$$= \exp\left(\left(-\frac{p_c^2}{3p_{j_{\min}}} + \frac{2p_c}{3} - \frac{np_{j_{\min}}}{3}\right)\right) \quad (4.85)$$

Therefore, we know

$$\mathbb{P}\left(\hat{K}_{BIP} \subset K_\beta\right) = 1 - \mathbb{P}\left(\min_{j\in J_\beta} \frac{1}{n} \sum_{b=1}^n \mathbb{1}\left\{|\tilde{\beta}_j^b| > \sigma\sqrt{2\log(1/\delta)(\mathbf{X}^{b\top}\mathbf{X}^b)^{-1}_{jj}}\right\} < p_c\right) \quad (4.86)$$

$$\geq 1 - q \exp\left(\left(-\frac{p_c^2}{3p_{j_{\min}}} + \frac{2p_c}{3} - \frac{np_{j_{\min}}}{3}\right)\right), \quad (4.87)$$

as stated in Thm. 5 (ii).

Part (iii)

To show $\mathbb{P}(\hat{K}_{BIP} = K_\beta) \rightarrow 1$, we need to jointly consider both sides that $\mathbb{P}(K_\beta \subset \hat{K}_{BIP}) \rightarrow 1$ and $\mathbb{P}(\hat{K}_{BIP} \subset K_\beta) \rightarrow 1$.

From condition (4.1.b), we see

$$\mathbb{P}(K_\beta \subset \hat{K}_{BIP}) = 1 - \mathbb{P}\left(\max_{j \in K_\beta} \frac{1}{n} \sum_{b=1}^n \mathbb{1}\left\{|\bar{\beta}_j^b| > \sigma \sqrt{2 \log(1/\delta) (\mathbf{X}^{b\top} \mathbf{X}^b)^{-1}_{jj}}\right\} > p_c\right) \quad (4.88)$$

$$\geq 1 - (p - q) \exp\left(\frac{1}{3} - \frac{np_c}{3}\right) \rightarrow 1. \quad (4.89)$$

From condition (4.1.d), we have that $\Phi\left(\frac{\sqrt{nr_0\rho_1} \min_{i \in J_\beta} |\beta_i|}{\sigma} - \sqrt{2 \log(1/\delta)}\right)$ does not shrink towards 0 as $n \rightarrow \infty$. Therefore, with (4.1.b), we have

$$\mathbb{P}(\hat{K}_{BIP} \subset K_\beta) = 1 - \mathbb{P}\left(\max_{j \in K_\beta} \frac{1}{n} \sum_{b=1}^n \mathbb{1}\left\{|\bar{\beta}_j^b| > \sigma \sqrt{2 \log(1/\delta) (\mathbf{X}^{b\top} \mathbf{X}^b)^{-1}_{jj}}\right\} > p_c\right) \quad (4.90)$$

$$\geq 1 - \exp\left(\left(-\frac{p_c^2}{3p_{j_{\min}}} + \frac{2p_c}{3} - \frac{np_{j_{\min}}}{3}\right)\right) \rightarrow 1. \quad (4.91)$$

Summarizing the results from both sides, we can conclude that

$$\mathbb{P}(\hat{K}_{BIP} = K_\beta) \rightarrow 1. \quad (4.92)$$

Finally, from the nature of the algorithm (least-squares after variable selection), we also have the limiting distribution, the same as if the sparsity pattern is known before inference, which is the oracle property.

□

Proof of Lemma 4.2

We require the following conditions like Thm. 5.

$$(4.2.a) \ \epsilon \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2 \mathbf{I}_n).$$

$$(4.2.b) \text{ For any bootstrap replicate } \mathbf{X}^b, \text{ the lowest eigenvalue of } \frac{1}{n} \mathbf{X}^{b\top} \mathbf{X}^b \text{ is lower bounded}$$

by $r_0\rho_1$ where $r_0 \in \mathbb{R}$ is a constant.

(4.2.c) $\rho_1 > 0$ and $p_{j_{\min}} = \min_{j \in J_\beta} \Phi\left(\frac{\sqrt{nr_0\rho_1}|\beta_i|}{\sigma} - \sqrt{2\log(n)}\right)$, the constant

eta satisfies

$\eta + \frac{1}{n} < p_{j_{\min}}$ Here $\Phi(\cdot)$ is the CDF of standard normal distribution.

Proof. Consider this definition of Δ

$$\Delta = \underbrace{\min_{j \in J_\beta} \frac{1}{n} \sum_{b=1}^n \mathbb{1} \left\{ |\bar{\beta}_j^b| > \sigma \sqrt{2 \log(1/\delta) (\mathbf{X}^{b\top} \mathbf{X}^b)^{-1}_{jj}} \right\}}_{\mathcal{I}_1} - \underbrace{\max_{k \in K_\beta} \frac{1}{n} \sum_{b=1}^n \mathbb{1} \left\{ |\bar{\beta}_k^b| > \sigma \sqrt{2 \log(1/\delta) (\mathbf{X}^{b\top} \mathbf{X}^b)^{-1}_{kk}} \right\}}_{\mathcal{I}_2}. \quad (4.93)$$

Recall that we set $\delta = \frac{1}{n}$ and define $p_{j_{\min}} = \min_{j \in J_\beta} \Phi\left(\frac{\sqrt{nr_0\rho_1}|\beta_i|}{\sigma} - \sqrt{2\log(1/\delta)}\right)$. Given conditions (4.2.a), (4.2.b) and (4.2.c), recall the proof in Thm. 5, we know that the random variables in $\mathcal{I}_1$ and $\mathcal{I}_2$ can be lower/upper bounded by $\text{Binomial}(n, p_{j_{\min}})$ and $\text{Binomial}(n, 1/n)$ respectively.

Therefore, in $\mathcal{I}_1$, using Chernoff bound, we see with probability at least $1 - qe^{-n(\frac{1}{2}p_{j_{\min}} - \frac{\eta}{2} - \frac{1}{2n})^2/2}$,

$$\mathcal{I}_1 \geq \frac{1}{2}p_{j_{\min}} + \frac{\eta}{2} + \frac{1}{2n}. \quad (4.94)$$

Similarly, in $\mathcal{I}_2$, using Chernoff bound, we see with probability at least $1 - (p - q)e^{-n(\frac{1}{2}p_{j_{\min}} - \frac{\eta}{2} - \frac{1}{2n})^2/2}$,

$$\mathcal{I}_2 \leq \frac{1}{2}p_{j_{\min}} - \frac{\eta}{2} + \frac{1}{2n}. \quad (4.95)$$

The construction here involves $\frac{1}{2}p_{j_{\min}}$ and $\frac{1}{2n}$ to matches up the two probabilities with the same exponential factor. From this construction, we could consider (and simplify) the probability of the joint event with an union bound, we know that $\Delta \geq \eta$ holds with probability

at least

$$\left(1 - e^{-n\left(\frac{1}{2}p_{j_{\min}} - \frac{\eta}{2} - \frac{1}{2n}\right)^2/2}\right)^q \left(1 - e^{-n\left(\frac{1}{2}p_{j_{\min}} - \frac{\eta}{2} - \frac{1}{2n}\right)^2/2}\right)^{p-q} \quad (4.96)$$

$$= \left(1 - e^{-n\left(\frac{1}{2}p_{j_{\min}} - \frac{\eta}{2} - \frac{1}{2n}\right)^2/2}\right)^p \quad (4.97)$$

$$\geq 1 - pe^{-n\left(\frac{1}{2}p_{j_{\min}} - \frac{\eta}{2} - \frac{1}{2n}\right)^2/2}. \quad (4.98)$$

□

Proof Theorem 6

In the asymptotic settings, we do not require further distributional assumptions (as shown in C.1). However, since we do not have non-asymptotic Gaussian tail behaviors, the threshold for subset estimation is alternatively determined via the standard deviation estimate with a factor γ as in [109]. In specific,

$$\hat{K}_{BIP} = \left\{ j \in I : \frac{1}{n} \sum_{b=1}^n \mathbb{1} \left\{ |\bar{\beta}_j^b| > \gamma \bar{\sigma}_{jj}^b \right\} \leq p_c \right\}. \quad (4.99)$$

We further require the following conditions:

$$(5.1.a) \mathbb{E}[\epsilon] = 0, \text{var}(\epsilon) = \sigma^2 < \infty, \epsilon \stackrel{\text{i.i.d.}}{\sim} P.$$

$$(5.1.b) \rho_1 > 0, \max_i (\mathbf{e} X_i^\top \mathbf{e} X_i) = O(p), \text{ and } p/(n\rho_1) = o(1).$$

$$(5.1.c) \text{ The min eigenvalue } \rho_1 \text{ and max eigenvalue } \rho_2 \text{ satisfy } \frac{\rho_1 \rho_2^{-1} \gamma^{2-q}}{q^{1/2}} \rightarrow \infty.$$

$$(5.1.d) \frac{\rho_1 \rho_2^{-1} \left(n^{1/2} \sigma^{-1} \rho_1^{1/2} \min_{j \in J_\beta} |\beta_j| - \gamma \right)^2 - q}{q^{1/2}} \rightarrow \infty$$

Proof. First, by the construction of $\hat{\beta}^*$, it performs least-square estimation based on the bagging inclusion probability. The Bootstrap replicates are generated from resampling of the original data, which can be traversed. Therefore, when $m \rightarrow \infty$, the conditional estimation of $\hat{K}_{BIP}^\infty \mid Y$ will be a fixed set of indices.

Then, we show that $\hat{\sigma}_m^{*2} \xrightarrow{P} \sigma^2$. Denote the event $\mathcal{E}$ which includes all cases when $\{\hat{K}_{BIP}^\infty = \hat{K} = K_\beta \mid Y\}$. This event describes when both BIP and TLS are performing a correct

subset estimation. We have

$$\lim_{n,m \rightarrow \infty} \left\{ \mathbb{P}(|\hat{\sigma}_m^{*2} - \sigma^2| > \epsilon) \mid Y \right\} \leq \lim_{n,m \rightarrow \infty} \left\{ \mathbb{P}(|\hat{\sigma}_m^{*2} - \sigma^2| > \epsilon \mid \mathcal{E}) P(\mathcal{E}) + \mathbb{P}(|\hat{\sigma}_m^{*2} - \sigma^2| > \epsilon \mid \mathcal{E}^c) P(\mathcal{E}^c) \right\} \quad (4.100)$$

$$= 0 + o_P(1) = o_P(1). \quad (4.101)$$

We see the probability $\mathbb{P}(|\hat{\sigma}_m^{*2} - \sigma^2| > \epsilon \mid \mathcal{E})$ converges to zero by Thm. 2.2 (b) [94]. For the other term, notice that $P(\mathcal{E})$ in the asymptotic regime ($n \rightarrow \infty$), with all the regularity conditions, we have $P(\hat{K} \neq K_\beta) = o_P(1)$ and $P(\hat{K}_{BLP} = K_\beta) \leq o_P(1)$. By using the union bound, we see $P(\mathcal{E}^c) = o_P(1)$, which gives us $\hat{\sigma}_m^{*2} = \sigma^2 + o_P(1)$. In general, most of the convergence in probability results in the Theorem can be shown in this way due to $P(\mathcal{E}^c) = o_P(1)$. We constraint first within the event $\mathcal{E}$ in the following analysis.

Conditioning on the event $\mathcal{E}$, we have $\hat{K}_{BLP}^\infty = \hat{K} = K_\beta$. By the Slutsky's theorem, it suffices to show that the law of $\hat{s}^{-1/2} a^\top (\hat{\beta}^* - \hat{\beta} \mid Y)$ is consistent.

$$\hat{s}^{-1/2} a^\top (\hat{\beta}^* - \hat{\beta}) = \hat{s}^{-1/2} a_j^\top ((X_j(m)^\top X_j(m))^{-1} X_j^\top(m) (X_j(m) \hat{\beta} + \hat{\epsilon}^*(m)) - \hat{\beta}) \quad (4.102)$$

$$= \hat{s}^{-1/2} a_j^\top ((X_j(m)^\top X_j(m))^{-1} X_j(m)^\top \hat{\epsilon}^*(m)) \quad (4.103)$$

$$= \sum_{i=1}^m \hat{\alpha}_i \hat{\epsilon}_i^*, \quad (4.104)$$

where $\hat{\alpha}_i = \hat{s}^{-1/2} a_j^\top ((X_j(m)^\top X_j(m))^{-1} X_{i,j}(m))$. To show the law convergence in probability to standard normal, we have to show

$$\mathcal{L} \left(\sum_{i=1}^m \hat{\alpha}_i \hat{\epsilon}_i^* \mid Y \right) \xrightarrow{Pr} \mathcal{N}(0, 1).$$

It is clear to see

$$\mathbb{E} [\hat{\alpha}_i \hat{\epsilon}_i^* \mid Y] = 0 \text{ a.s.} \quad (4.105)$$

Consider $\text{var}(\sum_{i=1}^m \hat{\alpha}_i \hat{\epsilon}_i^* | Y)$, we see

$$\text{var}\left(\sum_{i=1}^m \hat{\alpha}_i \hat{\epsilon}_i^* | Y\right) = \sum_{i=1}^m \hat{s}^{-1} a_j^\top (X_j(m)^\top X_j(m))^{-1} X_{i,j}(m) X_{i,j}(m)^\top (X_j(m)^\top X_j(m))^{-1} a_j \cdot \text{var}(\hat{\epsilon}_i^* | Y) \quad (4.106)$$

$$= \sum_{i=1}^m \hat{s}^{-1} a_j^\top (X_j(m)^\top X_j(m))^{-1} X_{i,j}(m) X_{i,j}(m)^\top (X_j(m)^\top X_j(m))^{-1} a_j \hat{\sigma}^2 \quad (4.107)$$

$$= \hat{s}^{-1} \hat{\sigma}^2 a_j^\top (X_j(m)^\top X_j(m))^{-1} a_j \quad (4.108)$$

$$= \left(a_j^\top (X_j(m)^\top X_j(m))^{-1} a_j\right)^{-1} a_j^\top (X_j(m)^\top X_j(m))^{-1} a_j = 1. \quad (4.109)$$

The $\text{var}(\hat{\epsilon}_i^* | Y) = \hat{\sigma}^2$ because the bootstrap resampling of residuals from $\hat{\beta}$ will have the variance of $\hat{\beta}$, which has limiting distribution $\hat{s}^{-1/2} a^\top (\hat{\beta} - \beta) \xrightarrow{d} \mathcal{N}(0, 1)$.

Therefore, we have $\sum_{i=1}^m \text{var}(\sum_{i=1}^m \hat{\alpha}_i \hat{\epsilon}_i^* | Y) \xrightarrow{Pr} 1$.

Up to now, we have shown $\mathbb{E}[\hat{\alpha}_i \hat{\epsilon}_i^* | Y]$ and $\text{var}(\hat{\alpha}_i \hat{\epsilon}_i^* | Y)$ converges in probability to 0 and 1 respectively. Therefore, to apply the Lindeberg-Feller central limit theorem as stated in Lemma 9, we only have to verify the Lindeberg condition to show the law converges to standard normal distribution in probability.

Notice first from the assumption, $\max_{1 \leq i \leq m} |\hat{\alpha}_i| = o_P(1)$. We could see this by

$$\max_{1 \leq i \leq m} \hat{\alpha}_i^2 = \max_{1 \leq i \leq m} \left\{ \frac{a_j^\top (X_j(m)^\top X_j(m))^{-1} X_{i,j}(m) X_{i,j}(m)^\top (X_j(m)^\top X_j(m))^{-1} a_j}{\hat{\sigma}^2 a_j^\top (X_j(m)^\top X_j(m))^{-1} a_j} \right\} \quad (4.110)$$

$$\leq \frac{\max_{1 \leq i \leq m} \|X_{i,j}\|^2}{n \hat{\sigma}^2 \rho_1} = O_P\left(\frac{p}{n \rho_1}\right) = o_P(1). \quad (4.111)$$

Therefore, we see that

$$\max_{1 \leq i \leq m} \mathbb{E} \left[|\hat{\epsilon}^*|^2 \mathbb{1} \{ |\hat{\alpha}_i \hat{\epsilon}^*| \geq \delta \} \mid Y \right] \quad (4.112)$$

$$\leq \mathbb{E} \left[\hat{\epsilon}^{*2} \mathbb{1} \left\{ |\hat{\epsilon}^*| \max_{1 \leq i \leq m} \hat{\alpha}_i \geq \delta \right\} \mid Y \right] \quad (4.113)$$

$$= \mathbb{E} \left[\hat{\epsilon}^{*2} o_P(1) \mid Y \right] \quad (4.114)$$

$$= o_P(1). \quad (4.115)$$

Compute $\sum_{i=1}^m \hat{\alpha}_i^2$ under event $\mathcal{E}$, we have

$$\sum_{i=1}^m \hat{\alpha}_i^2 = \sum_{i=1}^m \hat{s}^{-1} a_{\hat{j}}^\top (X_{\hat{j}}(m)^\top X_{\hat{j}}(m))^{-1} X_{i,\hat{j}}(m) X_{i,\hat{j}}(m)^\top (X_{\hat{j}}(m)^\top X_{\hat{j}}(m))^{-1} a_{\hat{j}} \quad (4.116)$$

$$= \frac{1}{\sigma^2}. \quad (4.117)$$

Therefore, we know $\sum_{i=1}^m \hat{\alpha}_i^2 \rightarrow \frac{1}{\sigma^2} = O(1)$.

Summarizing the results above, we have the Lindeberg condition that

$$\sum_{i=1}^m \mathbb{E} \left[|\hat{\alpha}_i \hat{\epsilon}_i^*|^2 \mathbb{1} \{ |\hat{\alpha}_i \hat{\epsilon}_i^*| \geq \delta \} \mid Y \right] \xrightarrow{P} 0 \text{ for all } \delta > 0. \quad (4.118)$$

From $\mathbb{E} [\hat{\alpha}_i \hat{\epsilon}_i^* \mid Y] \xrightarrow{P} 0$ and $\text{var} (\hat{\alpha}_i \hat{\epsilon}_i^* \mid Y) \xrightarrow{P} 1$ and the verified Lindeberg condition, we conclude that $\mathcal{L} (\sum_{i=1}^m \hat{\alpha}_i \hat{\epsilon}_i^* \mid Y) \xrightarrow{P} \mathcal{N}(0, 1)$.

□

Proof Theorem 7

We require the same set of conditions as in Thm. 6.

Proof. From Thm. 2 in [137], since our setting implies $\lambda_n/n \rightarrow 0$, we see $\tilde{\beta}_{SSL} = \bar{\beta}$ asymptotically.

Thus, from Lemma 2.2, we have

$$\mathcal{L}(\tilde{\beta}_{SSL} - \beta) \xrightarrow{d} \mathcal{N}(0, \bar{\sigma}^2 (X^\top X)^{-1}). \quad (4.119)$$

From Thm. 6, we see from the pivot of Bootstrap replicates

$$\mathcal{L}(\hat{\beta}^* - \hat{\beta}) \xrightarrow{d} \mathcal{N}\left(0, \hat{\sigma}_m^{*2} \left(X_{J_\beta}^\top X_{J_\beta}\right)^{-1}\right) \quad (4.120)$$

Further, from Thm. 2.1 [109], we see

$$\mathcal{L}(\hat{\beta} - \beta) \xrightarrow{d} \mathcal{N}\left(0, \hat{\sigma}^2 \left(X_{J_\beta}^\top X_{J_\beta}\right)^{-1}\right). \quad (4.121)$$

To show the convergence in the squared Wasserstein distance, we first recall the definition

$$\mathcal{W}_2^2(\mu, \nu) = \inf_{\pi \in \Pi(\mu, \nu)} \mathbb{E}_{(x, y) \sim \pi} [\|x - y\|_2^2]. \quad (4.122)$$

The Wasserstein distance considers the infimum of all push-forward mappings that assigns μ to ν . We first show the following result on the ℓ_2 norm between two vectors $x, y, z \in \mathbb{R}^p$ that

$$\|x - z\|_2^2 \leq (\|x - y\|_2 + \|y - z\|_2)^2 \quad (4.123)$$

$$= \|x - y\|_2^2 + \|y - z\|_2^2 + 2\|x - y\|_2 \|y - z\|_2 \quad (4.124)$$

$$\leq \|x - y\|_2^2 + \|y - z\|_2^2 + 2\left(\frac{1}{2}\|x - y\|_2^2 + \frac{1}{2}\|y - z\|_2^2\right) \quad (4.125)$$

$$= 2\left(\|x - y\|_2^2 + \|y - z\|_2^2\right). \quad (4.126)$$

Therefore, placing this result into the squared of Wasserstein-2 distance, suppose ρ is also

a probability measure, we have

$$\inf_{\pi \in \Pi(\mu, \rho, \nu)} \mathbb{E}_{(x, y, z) \sim \pi} [\|x - y\|_2^2] \leq 2 \inf_{\pi \in \Pi(\mu, \rho, \nu)} \mathbb{E}_{(x, y, z) \sim \pi} [\|x - y\|_2^2 + \|y - z\|_2^2] \quad (4.127)$$

$$= 2 \left(\inf_{\pi \in \Pi(\mu, \rho)} \mathbb{E}_{(x, y) \sim \pi} [\|x - y\|_2^2] + \inf_{\pi \in \Pi(\rho, \nu)} \mathbb{E}_{(y, z) \sim \pi} [\|y - z\|_2^2] \right) \quad (4.128)$$

$$= 2 (\mathcal{W}_2^2(\mu, \rho) + \mathcal{W}_2^2(\rho, \nu)). \quad (4.129)$$

We then see the following

$$\mathcal{W}_2^2(\mathcal{L}(\hat{\beta}^*), \mathcal{L}(\tilde{\beta}_{SSL})) \leq 2 \left(\mathcal{W}_2^2(\mathcal{L}(\hat{\beta}^*), \mathcal{L}(\hat{\beta})) + \mathcal{W}_2^2(\mathcal{L}(\hat{\beta}), \delta_\beta) + \mathcal{W}_2^2(\mathcal{L}(\tilde{\beta}_{SSL}), \delta_\beta) \right) \quad (4.130)$$

$$\leq 2 \left(\mathbb{E} [\|\hat{\beta}^* - \hat{\beta}\|^2] + \mathbb{E} [\|\hat{\beta} - \beta\|^2] + \mathbb{E} [\|\tilde{\beta}_{SSL} - \beta\|^2] \right). \quad (4.131)$$

In the following inequality, we consider a trivial push-forward mappings that assigns μ to ν over the product measure (joint distribution) $d\mu d\nu$. For example, it 'earth-moves' the distribution of $\hat{\beta}$ towards the point mass at the target β by changing the probability assignment at each specific locations (or for all possible events). In this way, it simply considers the difference of two laws. This trivial push-forward mapping is certainly an upper bound of the Wasserstein distance from the definition.

Then, since all the pivot distributions are asymptotically normal. We could apply the multivariate sub-Gaussian tail results that

$$\mathcal{W}_2^2(\mathcal{L}(\hat{\beta}^*), \mathcal{L}(\tilde{\beta}_{SSL})) \leq 2 \left(\mathbb{E} [\|\hat{\beta}^* - \hat{\beta}\|^2] + \mathbb{E} [\|\hat{\beta} - \beta\|^2] + \mathbb{E} [\|\tilde{\beta}_{SSL} - \beta\|^2] \right) \quad (4.132)$$

$$= 2 \left(\hat{\sigma}_m^{*2} \text{tr} \left(\left(X_{J_\beta}^\top X_{J_\beta} \right)^{-1} \right) + \hat{\sigma}_m^{*2} \text{tr} \left(\left(X_{J_\beta}^\top X_{J_\beta} \right)^{-1} \right) + \bar{\sigma}^2 \text{tr} \left(\left(X^\top X \right)^{-1} \right) \right) \quad (4.133)$$

$$[\text{Largest eigenvalue}] \leq 2 \frac{p}{n\rho_1} \left(\hat{\sigma}^{*2} + \hat{\sigma}_m^{*2} + \bar{\sigma}^2 \right) \quad (4.134)$$

$$= 2 \frac{p}{n\rho_1} \left(3\sigma^{*2} + o_P(1) \right) \quad (4.135)$$

$$[\text{from (5.1.b)}] = 0. \quad (4.136)$$

From the sub-Gaussian results, we see the Wasserstein distance is upper bounded by $\frac{p}{n\rho_1}$ times a constant factor. From the asymptotic assumption (5.1.b), we could conclude the convergence in the Wasserstein distance.

□

Technical Lemmas

Lemma 9 (Lindeberg-Feller Central Limit Theorem). *For each n , let $\{X_{ni}\}_{i=1}^n$ be an independent collection of random variables. Suppose the means and variances exist and are finite. Let $\sigma_n^2 \equiv \sum_{i=1}^n \sigma_{ni}^2 = \sum_{i=1}^n \text{var}(X_{ni}) > 0$ holds for all n . Define $Y_{n,i} \equiv (X_{ni} - \mathbb{E}[X_{ni}])/\sigma_n$. Here, if the following condition holds:*

$$\forall \epsilon > 0, \quad \sum_i^{k_n} \mathbb{E} [\|Y_{n,i}\|^2 > \epsilon] \rightarrow 0 \quad [\text{Lindeberg condition}], \quad (4.137)$$

then we have asymptotic normality on $\sum_{i=1}^n Y_{ni}$ that

$$\sum_{i=1}^n Y_{ni} \Rightarrow \mathcal{N}(0, 1). \quad (4.138)$$

Lemma 10 (Slutsky's Theorem). *Say $\{X_n\}$ is a $\mathbb{R}^d$ -valued sequence of random variables and X_n weakly converges to X . We know if $\{Y_n\}$ is a $\mathbb{R}^d$ -valued sequence of random variables s.t. $Y_n \xrightarrow{P} c$ where $c \in \mathbb{R}$, then*

- (i) $X_n + Y_n$ weakly converges to $X + c$.
- (ii) $X_n \cdot Y_n$ weakly converges to cX .
- (iii) $\frac{X_n}{Y_n}$ weakly converges to $\frac{X}{c}$ when $c \neq 0$.

Lemma 11. *From the definition of a sub-Gaussian random vector with parameter (μ, σ^2) that*

$$\mathbb{E} [e^{\lambda(X-\mu)}] \leq e^{\sigma^2 \lambda^2 / 2}. \quad (4.139)$$

We have

$$\mathbb{P}(|X - \mu| \geq t) \leq 2e^{-\frac{t^2}{2\sigma^2}}, \quad (4.140)$$

from the Chernoff bound.

Lemma 12. Let $X \sim \text{Binomial}(n, p)$, $\forall \delta \in (0, 1)$, we know the upper tail is bounded via

$$\mathbb{P}(X \geq (1 + \delta)np) \leq \exp\left(-\frac{\delta^2 np}{3}\right), \quad (4.141)$$

and the lower tail is bounded via

$$\mathbb{P}(X \leq (1 - \delta)np) \leq \exp\left(-\frac{\delta^2 np}{2}\right). \quad (4.142)$$

Implementation of Out-of-bag weighted bagging inclusion probability selection

Algorithm 4 Out-of-bag inclusion probability thresholding least-squares estimation

Input: covariate matrix X , target vector y , thresholding constant γ , cross-validation proportion c , inclusion probability threshold p , number of bagging trials B .

1: **function** BIPVARIABLESELECTION(X, y, γ, c, p, B) :

2: $\text{MSEs} \leftarrow \text{List}()$

3: $\text{IncludeList} \leftarrow \text{List}()$

4: **For** b in $1, 2, \dots, B$:

5: $X_{\text{tr}}^b, X_{\text{te}}^b, y_{\text{tr}}^b, y_{\text{te}}^b \leftarrow \text{CROSSVALIDATE}(X, y, c)$

6: $\beta^b \leftarrow \text{LEASTSQUARES}(X_{\text{tr}}^b, y_{\text{tr}}^b)$

7: $\tau_{1:d}^b \leftarrow \sigma \sqrt{\frac{\gamma}{\text{diag}((X_{\text{tr}}^b)^\top X_{\text{tr}}^b)}}$

8: $\text{include}_{1:d}^b \leftarrow \left(|\beta_{1:d}^b| > \tau_{1:d}^b \right)$

9: $\text{MSE}^b \leftarrow \frac{1}{cn} \|y_{\text{te}}^b - X_{\text{te}}^b \beta^b\|_2^2$

10: $\text{IncludeList.append}(\text{include}^b)$

11: $\text{MSEs.append}(\text{MSE}^b)$

12: $w_b \leftarrow \frac{\exp(-\text{MSE}^b)}{\sum_{\ell=1}^B \exp(-\text{MSE}^\ell)}, \quad b = 1, \dots, B$

13: $\hat{\pi}_{1:d} \leftarrow \sum_{b=1}^B w_b \text{include}_{1:d}^b$

14: $\hat{\mathcal{J}}_{1:d} \leftarrow (\hat{\pi}_{1:d} > p)$

Output: selected support set $\hat{\mathcal{J}}$.

Additional plots

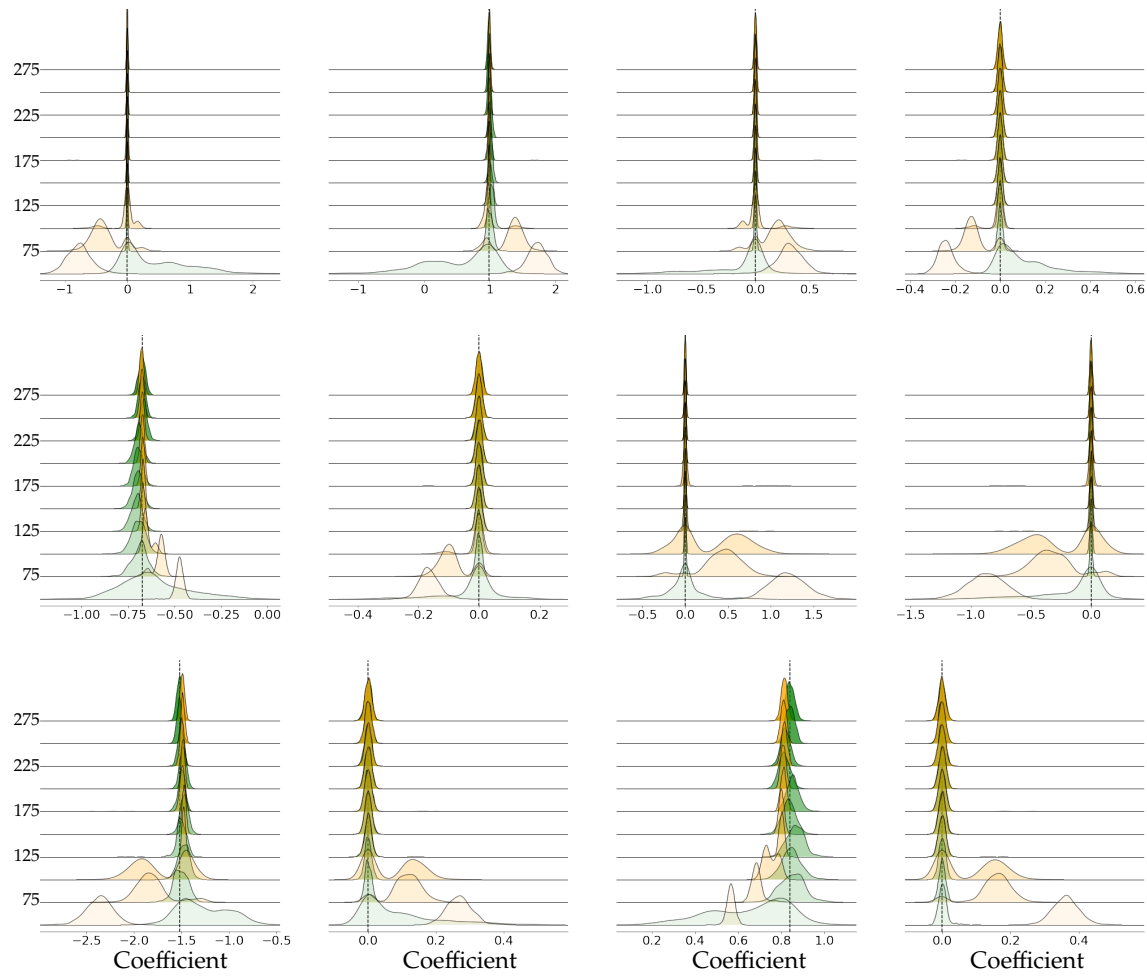


Figure 4.7: Extended plottings of Lotka–Volterra experiment with E-SINDy and Bayesian SINDy.

5

Mesh-Free Sparse Identification of Nonlinear Dynamics

Partial differential equations (PDEs) are one of the most fundamental building blocks of science and engineering. Through a descriptive symbolic formulation, PDEs can accurately predict spatio-temporal dynamics of different phenomena such as climate, fluid flow, optimal control, or population dynamics, among others. A-priori knowledge of the system dynamics is required to numerically solve the governing differential equations and obtain an accurate prediction of the system's future state. However, the explicit form of the governing dynamics can be unknown, so identifying the dynamics of the system is a critical first step for scientific data modeling and prediction [32, 234]. In this context, the sparse identification of nonlinear dynamics (SINDy) method [32] has shown great success in a wide range of applications, and has been extended to improve noise robustness [89] and to provide uncertainty quantification [100]. Despite the existence of black-box methods [185, 172], the most accurate and scientifically meaningful understanding still comes from recovering the explicit form of the governing equation [158, 301]. Still, its applicability is constrained to space-time structured data due to the reliance on traditional numerical discretization schemes (e.g. finite-difference methods).

In this paper, we present the mesh-free SINDy algorithm, a novel approach to identify the governing PDE from arbitrary sensor placements. As illustrated in Fig. 5.1, mesh-free SINDy is based on neural network (NN) approximation and auto-differentiation (AD) to discover the system dynamics from random sensor placement (ie. unstructured data). The mesh-free SINDy method first uses a NN as a surrogate model of a given unstructured dataset, ie.

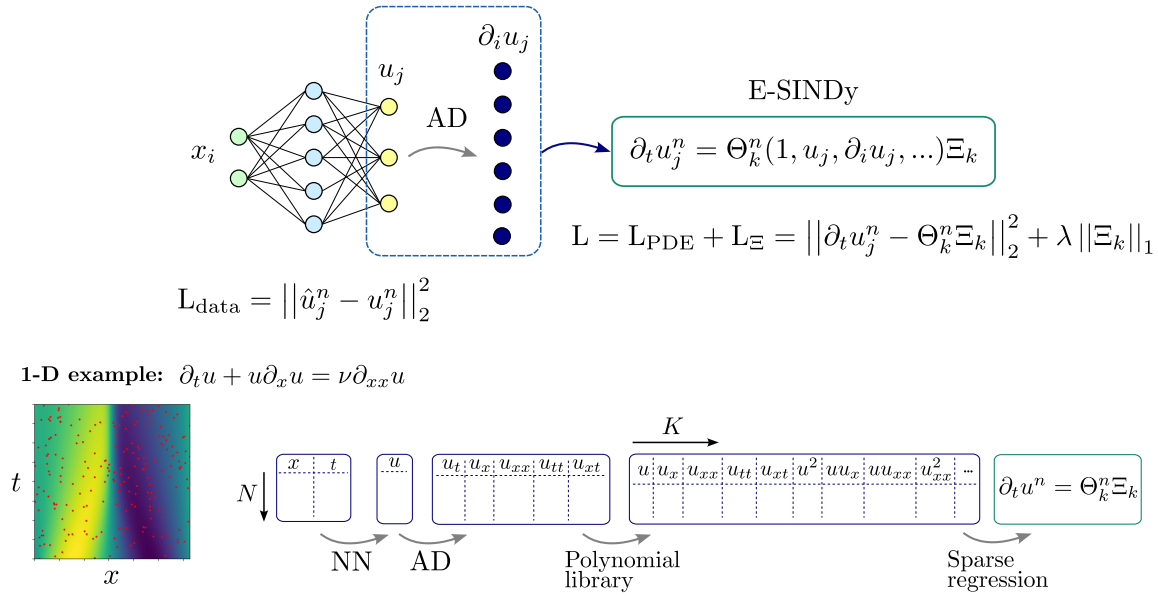


Figure 5.1: Illustration of the architecture of mesh-free SINDy. From N sensor measurements at random space-time points x_i^n , the neural network approximates the system state $u_j^n(x_i^n)$. As shown in the 1-D example, the auto-differentiation framework produces $\partial_t u$ (also noted u_t), and a library of K candidate terms composed of spatial or mixed derivatives which will be used in the sparse regression.

$f_{\text{NN}} : x_i \rightarrow u_j$, where x_i is the vector of independent variables and u_j is the NN-approximated system state. Once the NN is trained according to a loss function $L(u_j^n, \hat{u}_j^n)$ on N randomly sampled collocation points, a candidates library of K potential terms, Θ_k^n , is generated via AD by producing point-wise estimates of spatial, temporal, and mixed derivatives that may be part of the system dynamics. Then, the sparse regression step uses ensemble SINDy (E-SINDy) to find the dominant terms of the system dynamics from the library of candidates.

Unlike previous methods which rely on multi-stage training and a combined loss function, we propose to train the NN decoupled from the sparse regression step. This separation reduces the computational cost of the algorithm while also being robust to high noise levels and small datasets. We note that competing terms in a loss function pose a challenging optimization process which is particularly sensitive to the weight assign into each term. Additionally, we reduce the number of hyper-parameters by not having to set the weight of each objective in the loss function. For example, [201] assess the accuracy of an autoencoder combined with a neural ordinary differential equation (ODE) model for the autoencoder latent space to predict nonlinear temporal dynamics. They show that decoupling the loss

of the autoencoder and the neural ODE yields improved performance compared to training both models under a single combined loss function.

In the experiments, we demonstrate the algorithm’s efficacy on linear and non-linear dynamical systems of different complexity, comparing our results with previous methods. Mesh-free SINDy achieves successful model discovery with drastically reduced training cost and minimal hyperparameter tuning, all without sacrificing any performance. The contribution of our paper is three-fold:

- We introduce an AD-based SINDy framework to discover the underlying system dynamics from scarce and randomly sampled data with uncertainty quantification. A shallow neural network is trained as a response surface for the given dataset, and candidate terms of the dynamical system are computed via automatic differentiation.
- By decomposing the training loop into (i) surrogate learning and (ii) model discovery phases, it enables at least 10x faster execution compared to the traditional workflow, while requiring minimal hyperparameter tuning for model identification. This simplified discovery procedure does not compromise accuracy and can even enhance noise robustness in several cases.
- We conduct extensive experimental studies on four fundamental PDEs: the Burgers’, heat, Korteweg–De Vries, and advection-diffusion equations. Mesh-free SINDy consistently achieves robust discovery from scarce data, even in low-data and high-noise regimes. Detailed experimental setups are provided in Section 4 for full reproducibility.

5.1 Related works

The use of NN and AD to approximate the response of dynamic systems from unstructured data was initially proposed by Both et al. [29] through the *DeepMod* framework. Similarly to the present method, derivative terms are generated via AD from the NN response. DeepMod uses a multi-objective loss function to optimize the models’ weights based on (i) the error from NN system state prediction, $L_{\text{data}} = \|u_j^n - \hat{u}_j^n\|_2^2$, (ii) the error from the Lasso-based reconstructed dynamics, $L_{\text{PDE}} = \|\partial_t u_j^n - \Theta_k^n \Xi_k\|_2^2$, where Ξ_k is the coefficients vector for

each term in Θ_k , and (iii) the regularization term $L_{\Xi} = \lambda \|\Xi_k\|_1$ to promote sparsity, yielding $L = L_{\text{data}} + L_{\text{PDE}} + L_{\Xi}$. The overall training process, which optimizes both the NN trainable parameters and the Ξ_k coefficients, is performed twice. The first training includes the regularization term and, after training converges, the Ξ_k vector is reduced to a smaller number of terms according to an arbitrary threshold. The second training removes the regularization term and is aimed at fine-tuning the remaining Ξ_k coefficients.

Chen et al. [59] proposed a similar approach to DeepMod, namely *PINN-SR*, which combines both the data loss and the sparse regression of the Ξ_k coefficients in a single loss function. In that work, authors used an alternating direction optimization (ADO) algorithm instead of the double-training strategy implemented in DeepMod, and the Ridge algorithm instead of Lasso for finding the optimal Ξ_k coefficients. Recently, Stephany and Earils [257] proposed *PDE-LEARN*, a similar framework which also uses a combined loss function of the form $L = L_{\text{data}} + L_{\text{PDE}} + L_{\Xi}$. The novelty in PDE-LEARN relies on the use of rational NNs instead of classic multilayer perceptron architecture, and a triple-stage training strategy consisting of (i) burn-in stage to optimize and prune candidates based on data and PDE losses, (ii) sparsification stage which includes regularization term, and (iii) fine-tuning stage to optimize the coefficients for the surviving terms.

5.2 Mesh-free sparse identification of nonlinear dynamics

A neural network is a nonlinear modeling technique to approximate arbitrary functions. Given a dataset $\{(x^n, t^n, u^n)\}_{n=1}^N$ where x^n is the spatial location, t^n is the time, and $\hat{u}^n$ is the state variable of the PDE at (x^n, t^n) , the L -layer of a neural network can be defined as

$$u(x, t) := \sigma_L(W_k \sigma_{k-1}(W_{k-1} \cdots \sigma_1(W_1(x, t)))) , \quad (5.1)$$

where $\theta = (W_1, \dots, W_L)$ with $W_1 \in \mathbb{R}^{d \times p}$, $W_i \in \mathbb{R}^{p \times p}$, $W_L \in \mathbb{R}^{p \times 1}$, and $\sigma_i(\cdot)$ represents the activation function. Following prior work [92], we utilize a feed-forward neural network to approximate the PDE field.

Mesh-free PDE data PDEs utilize first principles to model physical and natural phenomena. Given spatial location x and temporal information t , the state variable u is a function of the independent spatial and temporal variables, i.e. $u(x, t)$. Typically, high-quality data collection under fixed spatial grids with regular temporal sampling is required to obtain partial derivative estimates from (e.g.) finite difference methods, which requires $x \in \mathbb{R}^d$ and $t \in \mathbb{R}$ to be structured grids (e.g. $x = [-0.5, -0.4, -0.3, \dots, 0.4, 0.5]$ and $t = [0, 0.1, 0.2, \dots, 1.0]$).

However, real-world data collection often results in non-uniformly sampled data points in space and time. For instance, uniformly spaced sensors for sea surface temperature data collection are not possible [225]. Hence, a non-uniform dataset of N samples can be defined as $\{(x^n, t^n; \hat{u}^n)\}_{n=1}^N$, where samples are randomly collected within a given spatio-temporal domain.

Automatic differentiation The automatic differentiation framework is a powerful tool in modern deep learning. Suppose we model the neural network as $f_\theta(x, t)$, where θ are the trainable parameters of the neural network. We can compute the spatial and temporal derivatives of the neural network with respect to the input (x, t) using the back-propagation algorithm [236]. From the computational graph of the back-propagation algorithm, we can estimate spatial, temporal, and mixed derivatives such as $\frac{\partial u}{\partial t}, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x \partial t}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^3 u}{\partial x^3}$, and so on.

Partial derivative estimation from mesh-free data Compared to traditional numerical differentiation methods, AD can obtain direct space-time derivatives of a non-uniform dataset from a fitted analytical response surface, such as a neural network. This mitigates interpolation errors that otherwise arise when projecting non-uniformly sampled data on structured grids. Adapting from the better estimated derivatives, we then perform sparse regression to discover the governing PDE via SINDy and its variants [32, 89, 20, 98].

We can now perform PDE discovery with arbitrary sensor placements with irregularly-sampled time series [230], which can relax the complexity of data collection and reduce the cost of data acquisition [138], as presented in Alg. 5.

Algorithm 5 MESH-FREE SINDy for sparse identification of nonlinear dynamics

Input: data $\mathcal{D} = \{(x^{(n)}, t^{(n)}, \hat{u}^{(n)})\}_{n=1}^N$, SINDy dictionary $\Theta(\cdot)$, sparsity threshold λ .

- 1: **function** MESHFREE SINDy($\mathcal{D}, \Theta, \lambda$) :
- 2: Initialize neural surrogate $f_\theta : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}$ with random weights
- 3: $\theta \leftarrow \arg \min_{\theta} \frac{1}{N} \sum_{n=1}^N \|\hat{u}^{(n)} - f_\theta(x^{(n)}, t^{(n)})\|_2^2$ $\triangleright$ MSE training convergence n in $1, 2, \dots, N$:
- 4: $u^{(n)} \leftarrow f_\theta(x^{(n)}, t^{(n)})$ $\triangleright$ forward pass
- 5: $\partial_t u^{(n)} \leftarrow \partial_t f_\theta(x^{(n)}, t^{(n)})$ $\triangleright$ auto-differentiation
- 6: $\partial_x^k u^{(n)} \leftarrow \partial_x^k f_\theta(x^{(n)}, t^{(n)})$ for required spatial derivative orders k
- 7: $\Theta \leftarrow \left[\Theta(u^{(n)}, \partial_x u^{(n)}, \partial_x^2 u^{(n)}, \dots) \right]_{n=1}^N$ $\triangleright$ build dictionary
- 8: $\Xi \leftarrow \arg \min_{\Xi} \frac{1}{N} \|\partial_t u - \Theta \Xi\|_2^2 + \lambda \|\Xi\|_1$ $\triangleright$ sparse regression

Output: sparse coefficient matrix Ξ .

Quantifying PDE discovery uncertainty from automatic differentiation Uncertainty quantification is an essential step in scientific machine learning and model discovery as it provides confidence of discovery as well as safe predictive inference for the discovered model. This is particularly important for critical applications such as health-care, climate modeling, or aerodynamic design, among others.

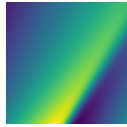
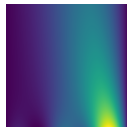
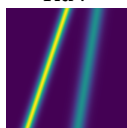
To enable uncertainty estimate for mesh-free SINDy, we perform non-parametric bootstrap via the following procedure. Firstly, we randomly subsample $m < N$ data points from the original dataset $\mathcal{D} = \{\Theta, \mathbf{u}_t\}$ without replacement, which we denote this data as $\tilde{\mathcal{D}}$, where $\Theta = \Theta(u^n, \partial_x u^n, \partial_x^2 u^n, \dots)$ and $\mathbf{u}_t = \partial_t u_\theta(x^{(n)}, t^{(n)})$. Then, the algorithm runs SINDy on the bootstrapped dataset $\tilde{\mathcal{D}}$ to perform sparse regression and obtain the PDE coefficients $\tilde{\Xi}$. Finally, repeating the above procedure M times to obtain M bootstrapped PDE coefficients $\{\tilde{\Xi}^m\}_{m=1}^M$.

The Bootstrapping algorithm above can obtain an empirical distribution $\{\tilde{\Xi}^m\}_{m=1}^M$, which enables uncertainty quantification for mesh-free SINDy. This algorithm can be implemented via ensemble SINDy [89], which is theoretically grounded to provide asymptotically correct distributional estimates [100]. In the experiments, we provide the uncertainty quantification results for the discovered PDEs via mesh-free SINDy.

5.3 Numerical experiments

In the following sections, we present case studies across a range of fundamental PDEs, including the Burgers' equation, the heat equation, the Korteweg–De Vries equation, and the 2D advection-diffusion equation. We summarize and highlight the experimental results in Tab. 5.1. Further details about each experiment are provided in the subsequent subsections.

Table 5.1: Performance of Mesh-free SINDy on tested PDEs

PDE	Sample	Noise	Success	Time (s)	Recovered Equation
Burgers'  $u_t + uu_x = 0.5u_{xx}$	100	1%	66.7%	18	$u_t + 1.03uu_x = 0.59u_{xx}$
	1000	1%	100.0%	20	$u_t + 0.98uu_x = 0.47u_{xx}$
	2500	1%	100.0%	20	$u_t + 1.00uu_x = 0.49u_{xx}$
	5000	1%	100.0%	21	$u_t + 1.00uu_x = 0.49u_{xx}$
	100	10%	50.0%	25	$u_t + 0.98uu_x = 0.41u_{xx}$
	1000	10%	100.0%	25	$u_t + 0.94uu_x = 0.39u_{xx}$
	2500	50%	80.0%	25	$u_t + 0.79uu_x = 0.37u_{xx}$
	5000	75%	80.0%	25	$u_t + 0.73uu_x = 0.37u_{xx}$
Heat  $u_t = u_{xx}$	500	1%	8.3%	29	$u_t = 0.35u_{xx}$
	1000	1%	91.7%	29	$u_t = 0.59u_{xx}$
	2500	1%	91.7%	30	$u_t = 0.75u_{xx}$
	4000	1%	100.0%	30	$u_t = 0.85u_{xx}$
	500	5%	8.3%	29	$u_t = 0.27u_{xx}$
	1000	10%	91.7%	29	$u_t = 0.48u_{xx}$
	2500	25%	8.3%	30	$u_t = 0.43u_{xx}$
	4000	25%	50.0%	30	$u_t = 0.33u_{xx}$
KdV  $u_t + 6uu_x + u_{xxx} = 0$	100	5%	8.3%	48	$u_t + 1.2uu_x + 0.18u_{xxx} = 0$
	500	5%	25.0%	52	$u_t + 5.75uu_x + 0.98u_{xxx} = 0$
	1000	5%	75.0%	56	$u_t + 5.38uu_x + 0.84u_{xxx} = 0$
	2000	5%	83.3%	69	$u_t + 5.89uu_x + 1.00u_{xxx} = 0$
	100	10%	0.0%	48	$u_t + 2.34uu_x + 1.06u_{xxx} = 0$
	500	10%	50.0%	52	$u_t + 5.31uu_x + 0.79u_{xxx} = 0$
	1000	25%	41.6%	56	$u_t + 5.03uu_x + 0.73u_{xxx} = 0$
	2000	50%	66.7%	40	$u_t + 5.02uu_x + 0.84u_{xxx} = 0$
Advection-Diffusion  $\partial_t u = -\nabla \cdot (\vec{c}u) + 0.5\nabla^2 u$	500	10%	100.0%	31	$u_t + 0.24u_x + 0.49u_y = 0.49u_{xx} + 0.49u_{yy}$
	2000	10%	100.0%	33	$u_t + 0.25u_x + 0.50u_y = 0.49u_{xx} + 0.49u_{yy}$
	4000	10%	100.0%	36	$u_t + 0.25u_x + 0.51u_y = 0.48u_{xx} + 0.49u_{yy}$
	6000	10%	100.0%	38	$u_t + 0.25u_x + 0.50u_y = 0.50u_{xx} + 0.48u_{yy}$
	500	25%	8.3%	31	$u_t + 0.27u_x + 0.50u_y = 0.45u_{xx} + 0.48u_{yy}$
	2000	25%	100.0%	33	$u_t + 0.24u_x + 0.49u_y = 0.47u_{xx} + 0.47u_{yy}$
	4000	50%	91.6%	36	$u_t + 0.24u_x + 0.51u_y = 0.45u_{xx} + 0.48u_{yy}$
	6000	75%	25.0%	38	$u_t + 0.28u_x + 0.48u_y = 0.49u_{xx} + 0.45u_{yy}$

5.3.1 Burgers' equation

The Burgers' equation is a fundamental nonlinear PDE that is used as a simplified model for turbulence and viscous fluid flow [17, 35] in 1D with the combination of nonlinear advection and diffusion.

The governing equation is given by

$$u_t + uu_x = \nu u_{xx}, \quad (5.2)$$

where u is the velocity, ν is the viscosity, u_t is the temporal derivative, and u_x, u_{xx} are first and second order spatial derivatives, respectively.

In the experiments, we set the viscosity $\nu = 0.5$ with a periodic boundary conditions for the domain $x \in [0, 2\pi]$ and $t \in [0, 1]$ where

$$u(0, t) = u(2\pi, t). \quad (5.3)$$

We simulate the PDE with random sensor placement $\{(x^n, t^n)\}_{n=1}^N$ where N is the sample size. To further simulate the PDE, we utilize the following closed-form solution of the viscous Burgers' equation

$$\hat{u}(x, t) = 4 - 2\nu \frac{\frac{d}{dx}\phi(x, t)}{\phi(x, 0)}, \quad (5.4)$$

where $\phi(x, t) = \exp\left(-\frac{(x-4t)^2}{(4\nu(t+1))^2}\right) + \exp\left(-\frac{(x-4t-2\pi)^2}{(4\nu(t+1))^2}\right)$.

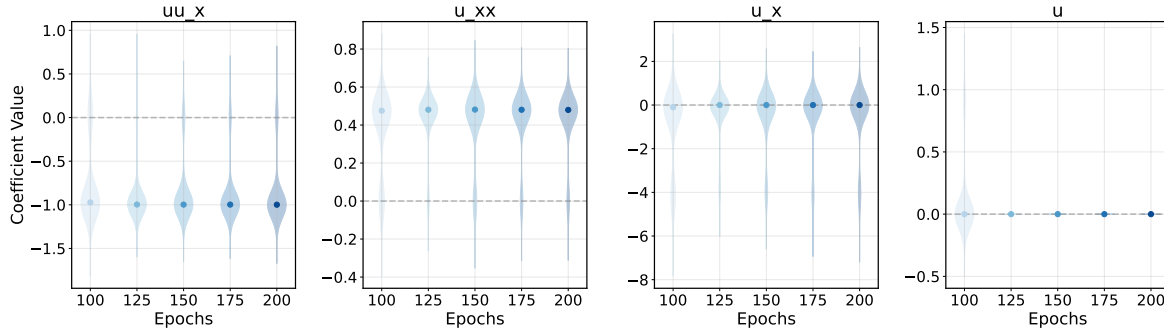
We setup a fully-connected two-layer neural network with 32 neurons per layer. For continuous differentiation, we utilize the tanh activation function. The neural network is trained with the AdamW optimizer with a learning rate of 0.0005, weight decay of 0.01, and a batch size of 20. We set the training epoch to 200 which approximately takes 25 seconds to run without GPU acceleration. To identify the system dynamics with SINDy, we set the interaction order to 2, which includes 19 polynomial terms including

$$1, u, u_x, u_{xx}, u_{tt}, u_{xt}, u^2, uu_x, uu_{xx}, uu_{tt}, uu_{xt}, u^3, u_x u_{xx}, u_x u_{tt}, u_x u_{xt}, u_x u^2, u_{tt} u_{xt}, u_{tt} u^2, u_{xt} u^2.$$

We utilize an ensemble sequentially thresholding least-squares algorithm with threshold 0.14 and ℓ_2 regularization 0.05. We note here that the hyperparameter setting is fixed among all sample sizes and noise levels. These hyper-parameters are kept constant during the

Table 5.2: PDE discovery success rate (%) under different sample sizes and noise levels for the Burgers' equation.

Samples	1% Noise	10% Noise	25% Noise	50% Noise	75% Noise
100	66.7	50.0	16.7	16.7	0.0
1000	100.0	100.0	50.0	25.0	0.0
2500	100.0	100.0	100.0	80.0	40.0
5000	100.0	100.0	100.0	100.0	80.0

**Figure 5.2:** Uncertainty quantification and epoch-wise evaluation of the discovered SINDy model for the Burgers' equation.

experiments when varying dataset size and noise level.

The training convergence as well as the neural network extrapolation are included in 5.5.2. We show in Tab. 5.2 that the mesh-free SINDy algorithm achieves robust discovery with up to 50% noise using 5,000 samples, with a 100% success rate. With 75% injected noise, we achieve an 80% success rate with 5,000 samples. In the low-data limit, we achieve a 66.7% success rate with 100 samples and 1% noise, and a 50% success rate with 10% noise, which is notably robust when testing non-uniformly sampled data.

Fig. 5.2 presents the uncertainty estimate of mesh-free SINDy, where ensemble SINDy discovery is performed every 25 epochs of training the neural network which provides the derivatives estimate. This figure demonstrates the distribution estimation of mesh-free SINDy with an increasing number of training epochs. It is worth noting that insufficient initial training can lead to incorrect model discovery, but the model will converge to the correct physics with more training epochs. In turn, it demonstrates that longer training periods leads to more accurate derivative estimation in results in a higher success rate of PDE discovery.

5.3.2 1D heat equation

The heat equation is a fundamental PDE that characterizes the heat transfer via a diffusion process. In 1D, the heat equation can be written as

$$u_t = \nu u_{xx}, \quad (5.5)$$

where u is the temperature and $\nu = 1.0$ is the diffusion coefficient. We set up the initial condition to be $u(x, 0) = x^2 \sin(x) \approx \sum_q D_q \sin(qx)$ in a $x \in [0, \pi]$ periodic domain, i.e. $u(0, t) = u(\pi, t) = 0$.

In the experiments, we use a neural network with four fully-connected layers of 128 neurons each, again using tanh as the activation function. A batch size of 20 samples is considered, and the learning rate is set to $2e^{-4}$ for 300 training epochs. The MSE loss function between target and predicted data is applied. Following the previous example, we set the interaction order to 3 thus including 12 polynomial terms

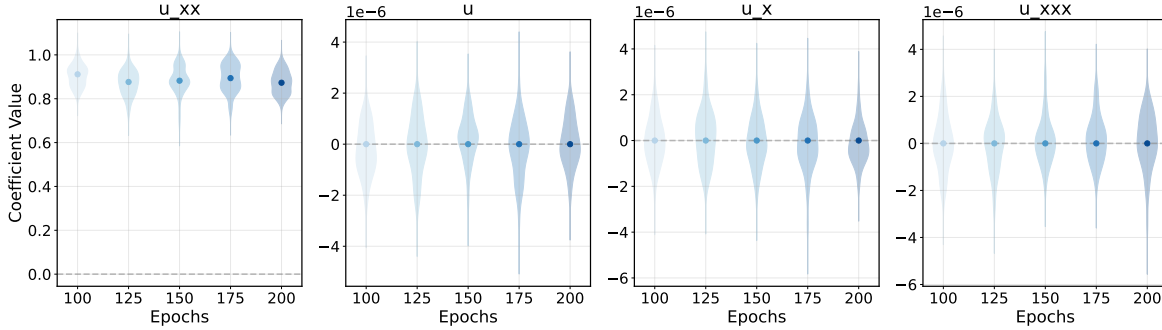
$$1, u, u^2, u_x, u_{xx}, u_{xxx}, uu_x, uu_{xx}, uu_{xxx}, u^2 u_x, u^2 u_{xx}, u^2 u_{xxx}.$$

We utilize mixed-integer optimizer to perform sparse regression with ℓ_0 regularization with $\ell_2 = 0.05$. These hyper-parameters are kept constant during the experiments when varying dataset size and noise level.

The convergence of training and the result of neural network extrapolation are in 5.5.3. In Tab. 5.3, mesh-free SINDy achieves robust discovery with up to 1% noise using 4,000 samples, with a 100% success rate. With 25% injected noise, we achieve a 50% success rate with 4,000 samples. In the low-data limit, we achieve a 91.7% success rate with 1000 samples and 1% noise. Remarkably, the model can still successfully discover the heat equation with only 500 samples under a noisy environment. Fig. 5.3 presents the uncertainty estimate of mesh-free SINDy, where ensemble SINDy discovery is performed every 25 epochs. The correct term u_{xx} is robustly discovered with statistical significance, while other terms are concentrated around zero.

Table 5.3: PDE discovery success rate (%) under different sample sizes and noise levels for the heat equation.

Samples	1% Noise	5% Noise	10% Noise	25% Noise
500	8.3	8.3	8.3	0.0
1000	91.7	83.3	91.7	0.0
2500	91.7	91.7	61.7	8.3
4000	100.0	75.0	66.7	50.0

**Figure 5.3:** Uncertainty quantification and epoch-wise evaluation of the discovered SINDy model for the heat equation.

5.3.3 Korteweg–De Vries equation

The Korteweg–De Vries (KdV) equation is a mathematical model of waves on shallow water surfaces. The equation is a third-order nonlinear PDE which can be written as

$$u_t + uu_x + u_{xxx} = 0, \quad (5.6)$$

where u is the wave height and u_x and u_t are the spatial and temporal derivatives.

In the experiments, we apply the two-soliton solution of the KdV equation with the initial condition being a superposition with functions of the form

$$\hat{u}(x, t) = \frac{c}{2} \text{sech}^2 \left(\frac{\sqrt{c}}{2} (x - ct - a) \right), \quad (5.7)$$

with offset centers and different amplitudes following [234].

In the experiments, we apply use a fully-connected neural network with four layers and 30 neurons per layer. The tanh activation function is employed. We use Adam op-

timizer with learning rate $1e^{-3}$ and apply full-batch gradient descent as the mesh-free dataset is small. The training time for 500 samples and 200 epochs is around 50 seconds using the MSE loss. The convergence of training and the result of neural network extrapolation are in 5.5.4. Following the previous example, we set the following library for discovery: $1, u, u^2, u_x, u_{xx}, u_{xxx}, uu_x, uu_{xx}, uu_{xxx}, u^2u_x, u^2u_{xx}, u^2u_{xxx}$. We utilize mixed-integer optimizer to perform sparse regression with ℓ_0 regularization with $\ell_2 = 0.05$. These hyper-parameters are kept constant during the experiments when varying dataset size and noise level.

Mesh-free SINDy achieves robust discovery under 1% noise using 2,000 samples with a 100% success rate. In Tab. 5.4, with 50% injected noise, we achieve a 66.7% success rate with 2,000 samples. In the low-data limit, we achieve a 50% success rate with 500 samples and 5% noise. Remarkably, the model can still successfully discover the KdV equation with only 100 samples under 5% noise. Fig. 5.4 presents the uncertainty estimate of mesh-free SINDy, where ensemble SINDy discovery is performed every 2500 epochs. The correct terms, u_{xxx} and uu_x , are robustly discovered with uncertainty estimates, while other terms are consistently evolve into a distribution centered at zero.

Table 5.4: PDE discovery success rate (%) under different sample sizes and noise levels for the KdV equation.

Samples	5% Noise	10% Noise	25% Noise	50% Noise
100	8.3	0.0	0.0	0.0
500	50.0	25.0	16.6	8.3
1000	75.0	50.0	41.6	33.3
2000	83.3	75.0	83.3	66.7

5.3.4 2D advection-diffusion equation

The two-dimensional advection-diffusion equation combines advection and diffusion processes in two spatial dimensions. Advection-diffusion equations are widely used to describe many linear processes in the physical sciences, including heat transfer, pollutant transport,

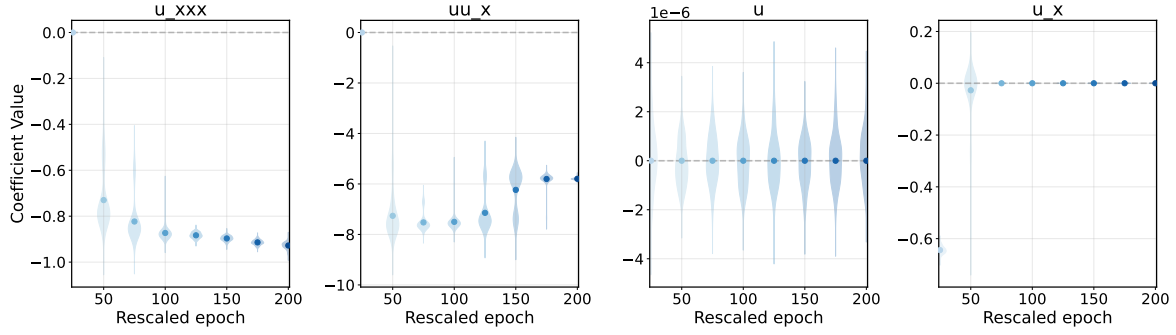


Figure 5.4: Uncertainty quantification and epoch-wise evaluation of the discovered SINDy model for the KdV equation.

Table 5.5: PDE discovery success rate (%) under different sample sizes and noise levels for the 2D advection-diffusion equation.

Samples	1% Noise	10% Noise	25% Noise	50% Noise	75% Noise
500	100.0	100.0	8.3	0.0	0.0
2000	100.0	100.0	100.0	16.7	0.0
4000	100.0	100.0	100.0	91.6	8.3
6000	100.0	100.0	100.0	91.6	25.0

and fluid dynamics. The equation is given by

$$\partial_t u = -\nabla \cdot (\vec{c}u) + K\nabla^2 u, \quad (5.8)$$

where u is the concentration, $\vec{c} = (0.25, 0.50)$ is the constant velocity vector, and $K = 0.50$ is the diffusion coefficient. We set the initial condition as a Gaussian distribution. In the experiments, we use neural network with eight fully-connected layers and 60 neurons per layer. The tanh activation function is employed. We use Adam optimizer with learning rate $1e^{-3}$ and apply full-batch gradient descent as the mesh-free dataset is small. The training time for 500 samples and 10,000 epochs is around 30 seconds under the MSE loss. Following the previous examples, we set the following library for discovery: $1, u, u^2, u^3, u_x, u_y, u_{xx}, u_{yy}, uu_x, uu_y$. Compared to all prior examples, this library fits into the 2D PDE discovery context, which includes ten library functions that are commonly applied. We utilize mixed-integer optimizer to perform sparse regression with ℓ_0 regularization with $\ell_2 = 0.0005$. These hyper-parameters are kept constant during the experiments when varying dataset size and noise level.

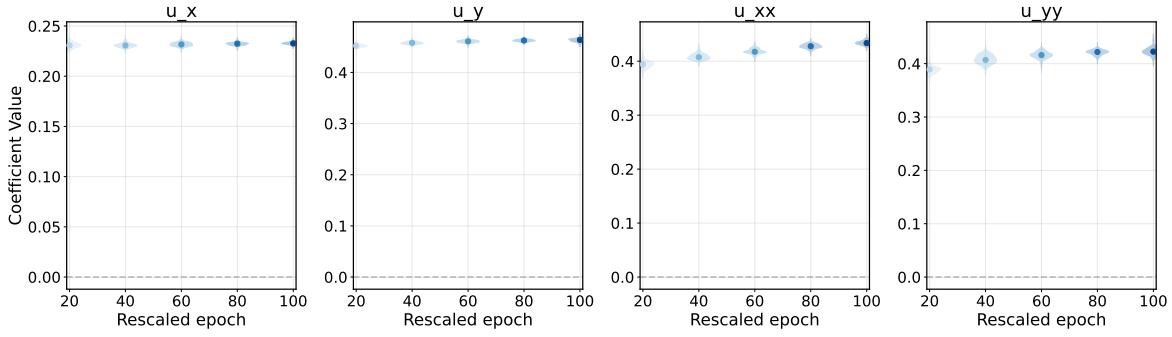


Figure 5.5: Uncertainty quantification and epoch-wise evaluation of the discovered SINDy model for the 2D advection-diffusion equation.

The convergence of training and the result of neural network extrapolation are in 5.5.5. Mesh-free SINDy achieves robust discovery under 25% noise using 6,000 samples with a 100% success rate. In Tab. 5.5, with 75% injected noise, we achieve a 25% success rate with 6,000 samples. In the low-data limit, we achieve a 100% success rate with 500 samples and 10% noise. Remarkably, the model can still successfully discover the complex advection-diffusion equation with only 500 samples up to 25% noise. Fig. 5.5 presents the uncertainty estimate of mesh-free SINDy, where ensemble SINDy discovery is performed every 2000 epochs. The correct terms are robustly discovered with uncertainty estimates.

5.3.5 Comparison with previous works

The performance of mesh-free SINDy is compared with previous methods using the Burgers' equation as a baseline. Other methods have been run according to the documentation and best-practices available in their open-source repositories. All methods are tested under the same GPU hardware.

Tab. 5.6 summarizes the results for each method. In general, mesh-free SINDy offers the simplest implementation with fast computation and minimal tuning, while maintaining high robustness to noise, especially in low-sample scenarios. Ensemble Weak (EW-)SINDy [89, 194] can tolerate very high noise levels, but it requires grid-based data collection with a large sample size, which is not always feasible in real-world applications. PINN-SR and PDE-LEARN both have relatively slow training times and require careful tuning of hyper-parameters. This can be particularly challenging in the context of deep learning, which adds extra complexity on

Table 5.6: Comparison of mesh-free SINDy with previous methods for the Burgers’ equation experiment.

Method	Cost	Noise level	Samples	Tuning	Success rate	Data Collection
Mesh-free SINDy	30s	100%	100	Low	High	Arbitrary
EW-SINDy [89]	1s	100%	25,000	Low	High	Grid
PINN-SR [59]	20m	10%	800	Medium	Medium	Arbitrary
PDE-LEARN [257]	30m	100%	250	High	Medium	Arbitrary
DeepMOD [29]	25m	75%	100	High	Medium	Arbitrary

top of the neural network tuning required for optimal training. In addition, reproducibility is also hindered by the fine-tuning procedure, which can be nontrivial and biased towards prior knowledge. DeepMOD has a fast training time and is robust to noise under very small sample sizes. However, the tuning effort required is high, as one potentially needs to tune the threshold for different levels of noise. The periodic thresholding scheduler also requires careful manipulation, which can be challenging in practice.

On the other hand, mesh-free SINDy has encountered limitations to discover very high-order PDEs, such as the Kuramoto—Sivashinsky equation, which contains a 4th-order derivative in space. Higher order terms may contain errors as a result of recursive (nested) first-order AD [216, 146]. Other methods enforce strong priors during the early pruning of the candidate library as part of a multi-stage training approach, which helps focus the sparse regression step on a reduced set of terms. Hence, future work will be focused on improving the AD step to produce better estimates of high-order derivatives.

5.4 Conclusion

In this paper we introduce mesh-free SINDy to discover governing equations from non-uniformly sampled and scarce data. This is a complex problem in general, as all partial derivative computation would require a grid for all difference-based methods. Utilizing the power of neural network learning and automatic differentiation, mesh-free SINDy demonstrates a remarkable procedure to uncover the hidden governing physical laws under noisy environments. In particular, we test the algorithm for the following systems: 1D viscous Burgers’ equation, 1D heat equation, Korteweg–De Vries equation, and 2D advection-diffusion

equation. In all computational examples, mesh-free SINDy is able to discover the correct form of the governing equation with a small number of samples. Under the high-noise limit, the algorithm can also identify the governing equation with high success rate with increasing number of sample sizes.

First, the present approach provides a data-driven way to compute partial derivatives for mesh-free data, which is a difficult task since all of the numerical differentiation algorithms are defined on a grid. Mesh-free SINDy shows that it is sufficient for a surrogate model to estimate partial derivatives via automatic differentiation methods. In contrast to prior works that require library setting and tuning under the correct hyperparameter schedule within the surrogate model training, as demonstrated in all of the computational examples, as long as the surrogate model $u(x, t)$ can approximate u well enough, the estimate of spatial and temporal derivatives is accurate. This hugely relaxes the efforts required for prior works to set up the correct physics loss during training, which requires the model to retrain from scratch once the library of functions or sparse regression parameters is changed. This provides a tool for practitioners to compute partial derivatives from mesh-free data.

Additionally, the AD estimate of partial derivatives is accurate enough to perform equation discovery automatically, and it further has the capability to smooth out the injected noise from the experiments. During the learning procedure, the shallow neural network can work as a noise filter that only captures information that is relevant to physics. The more training samples we have, the better the AD estimation of partial derivatives will be. This will open the black-box of prior works that identify the physics during the surrogate model training. Mesh-free SINDy provides a simple, clear, and interpretable procedure by only using a surrogate model to estimate partial derivatives. With the partial derivative estimation, we apply SINDy algorithms [32, 89, 234] algorithms to identify the governing equation.

All of the prior claims are verified with our computational examples. In our experimental study, we find that mesh-free SINDy is not only more interpretable but also improves the discovery success rate in noisy environments compared to prior jointly trained methods. It is both robust to noise and achieves a higher success rate on smaller datasets, while also substantially reducing overall training cost. We conclude mesh-free SINDy has better

interpretability with lower cost, which potentially works better in high-noise and low-data regimes for complex real-world scarce data.

5.5 Experimental details

5.5.1 Error Metrics

We evaluate our SINDy model using five complementary error metrics:

- **PDE Conservation Error** (E_{PDE}): Measures how well the discovered model satisfies the underlying partial differential equation conservation laws.

$$E_{\text{PDE}} = \frac{1}{N} \sum_{n=1}^N \left\| \frac{\partial \hat{u}_j(x^{(n)}, t^{(n)})}{\partial t} - \mathcal{F}_{\text{SINDy}}(u_j^{(n)}, \nabla u_j^{(n)}, \nabla^2 u_j^{(n)}, \dots) \right\|_2^2 \quad (5.9)$$

where $\mathcal{F}_{\text{SINDy}}$ represents the identified sparse PDE model and $\frac{\partial \hat{u}_j^{(n)}}{\partial t}$ the true system temporal derivative.

- **Neural Network Residual** (E_{NN}): Quantifies the fitting accuracy of the neural network to the training data.

$$E_{\text{NN}} = \frac{1}{N} \sum_{n=1}^N \left\| u_j^{(n)} - \hat{u}_j(x^{(n)}, t^{(n)}) \right\|_2^2 \quad (5.10)$$

where u is the neural network approximation of the state and $\hat{u}$ is the true (measured) system state.

- **Temporal Derivative Error** ($E_{\text{du dt}}$): Evaluates the accuracy in predicting the time derivatives.

$$E_{\text{du dt}} = \frac{1}{N} \sum_{n=1}^N \left\| \frac{\partial u_j^{(n)}}{\partial t} - \frac{\partial \hat{u}_j(x^{(n)}, t^{(n)})}{\partial t} \right\|_2^2 \quad (5.11)$$

where $\frac{\partial u}{\partial t}$ is the time derivative computed from the neural network.

- **SINDy Prediction Error** (E_{SINDy}): Measures how well the sparse identified model approxi-

mates the right-hand side dynamics.

$$E_{\text{SINDy}} = \frac{1}{N} \sum_{n=1}^N \left\| \frac{\partial u_j^{(n)}}{\partial t} - \mathcal{F}_{\text{SINDy}}(u_j^{(n)}, \nabla_x u_j^{(n)}, \nabla_x^2 u_j^{(n)}, \dots) \right\|_2^2 \quad (5.12)$$

comparing the neural network time derivatives with the SINDy model predictions.

- **State Reconstruction Error** (E_{field}): Quantifies the accuracy in reconstructing the solution field.

$$E_{\text{field}} = \frac{1}{N} \sum_{n=1}^N \left\| u_j^{(n)} - \hat{u}_j(x^{(n)}, t^{(n)}) \right\|_2^2 \quad (5.13)$$

where $\hat{u}_j$ represents the ground truth solution vector, potentially from high-fidelity simulations or experimental data.

5.5.2 Burgers' equation

In the training of Burgers' equation, the convergence of training and testing loss is shown in Fig. 5.6. For all noise levels, the training and testing loss converge synchronously, indicating that the model is performing similarly well on both training and testing datasets. We further visualize the predicted solution of the Burgers' equation in Fig. 5.7. It is observable that even with limited scarce data, the shallow neural network can accurately predict the solution of the Burgers' equation. Table 5.7 and Table 5.8 presents the error evolution of mesh-free SINDy from two different perspectives. From Table 5.7, we discover that with larger sample sizes, the error of mesh-free SINDy uniformly decreases and the model becomes more robust to noise. In Table 5.8, we provide more evaluation metrics to assess the performance of mesh-free SINDy with increasing noise levels.

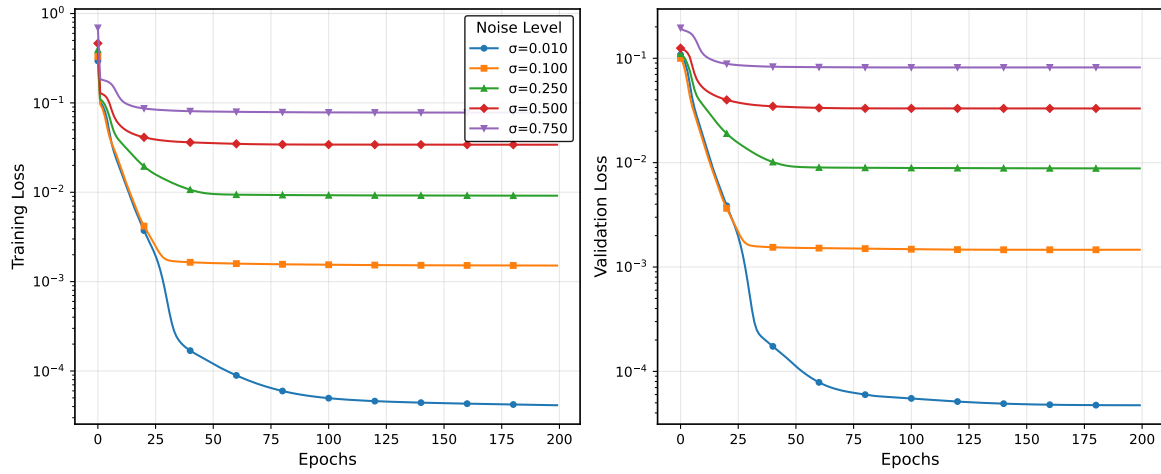


Figure 5.6: Epoch-wise training and validation loss convergence with different noise levels for the Burgers' equation.

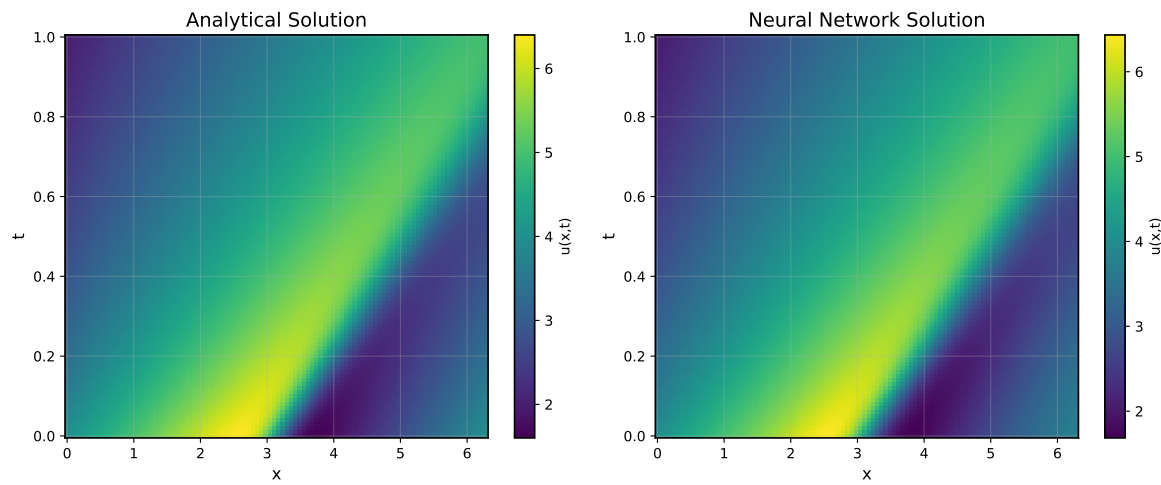


Figure 5.7: Real (left) vs. predicted (right) solution of the Burgers' equation.

Table 5.7: Loss of mesh-free SINDy under different sample sizes and noise levels for the Burgers' equation.

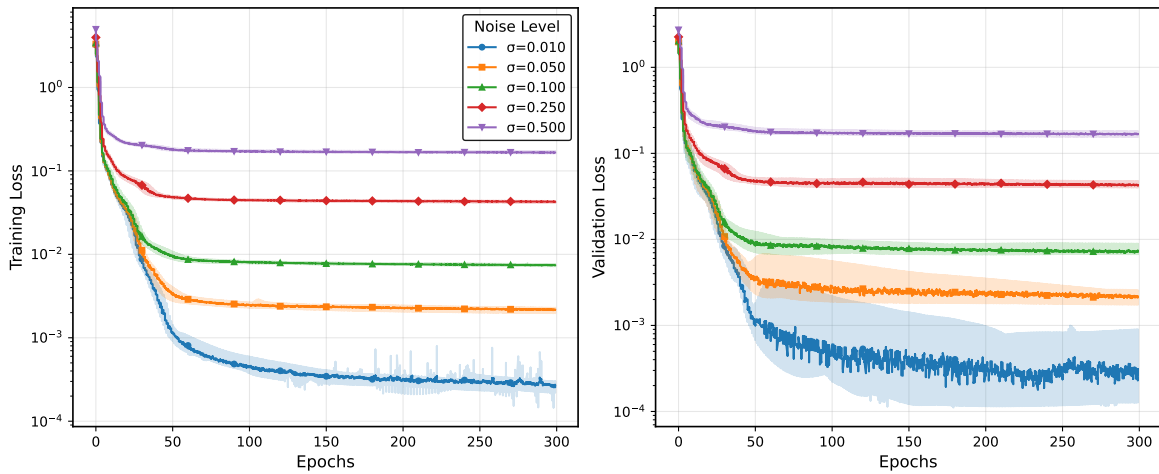
Samples	1% Noise	10% Noise	25% Noise	50% Noise	75% Noise
100	0.6162	0.8487	1.2749	1.9100	2.2014
1000	0.2663	0.3753	0.5485	0.8724	1.0492
2500	0.1379	0.2822	0.4027	0.5942	0.7665
5000	0.1011	0.2214	0.3747	0.5030	0.6842

Table 5.8: Error metrics of SINDy under different noise levels with 5000 samples for the Burgers' equation.

Metrics	1% Noise	10% Noise	25% Noise	50% Noise	75% Noise
E_{PDE}	4.76e-16	4.70e-16	4.88e-16	4.94e-16	4.75e-16
E_{NN}	0.0970	0.2912	0.5296	1.0727	1.4731
E_{dudt}	0.1095	0.2625	0.3242	0.4995	0.5658
E_{SINDy}	0.0966	0.2787	0.3001	0.5018	0.3995
E_{field}	0.1017	0.3318	0.5983	1.1713	1.4587

5.5.3 Heat equation

In the training of the heat equation, the convergence of training and testing loss is shown in Fig. 5.8. For all noise levels, the training and testing loss converge synchronously, indicating that the model is performing similarly well on both training and testing datasets. However, the testing loss demonstrates larger instability during training, which is likely due to the learning rate setting and choice of optimizer. We further visualize the predicted solution of the heat equation in Fig. 5.9. The solution of heat equation is nicely predicted via neural network. Table 5.9 and Table 5.10 presents the error evolution of mesh-free SINDy. From Table 5.9, we discover that with larger sample sizes, the error of mesh-free SINDy uniformly decreases and the model becomes more robust to noise. In Table 5.10, we provide more evaluation metrics to assess the performance of mesh-free SINDy with increasing noise levels.

**Figure 5.8:** Epoch-wise training and validation loss convergence with different noise levels for the heat equation.

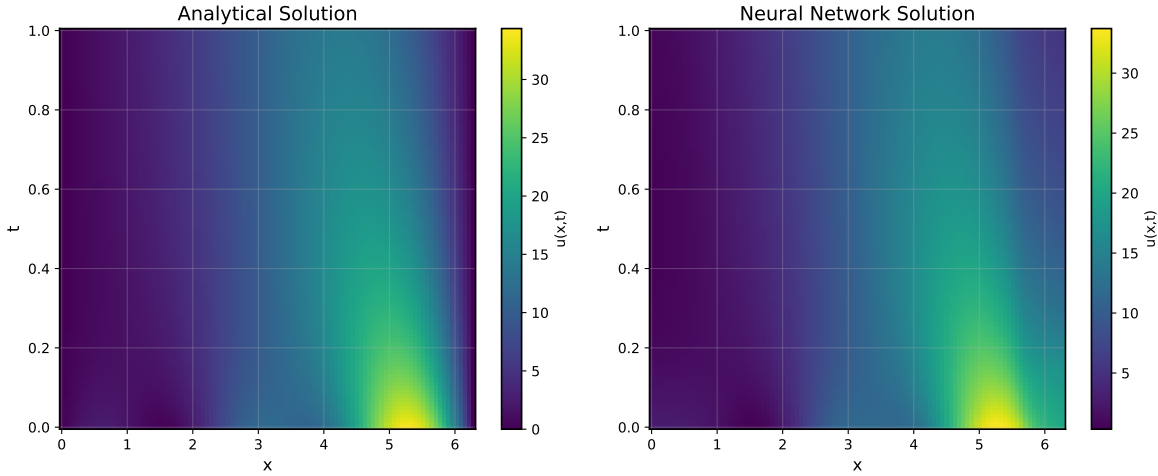


Figure 5.9: Real (left) vs. predicted (right) solution of the heat equation.

Table 5.9: Loss of mesh-free SINDy under different sample sizes and noise levels for the heat equation.

Samples	1% Noise	5% Noise	10% Noise	25% Noise
500	5.1618	5.1329	5.2472	5.5408
1000	3.3018	3.6175	3.6939	5.2842
2500	2.6697	3.2724	4.7077	5.2627
4000	2.2628	3.7658	4.4749	5.0642

Table 5.10: Error metrics of SINDy for under different noise levels with 5000 samples for the heat equation.

Metrics	1% Noise	5% Noise	10% Noise	25% Noise	50% Noise
E_{PDE}	5.53e-07	5.56e-07	5.64e-07	5.63e-07	5.56e-07
E_{NN}	1.9749	4.3219	4.8943	11.7437	12.9245
E_{dudt}	0.4288	1.0015	0.9983	1.9481	2.9653
E_{SINDy}	2.3853	3.8856	5.1131	5.1594	5.3305
E_{field}	5.17e+04	6.09e+04	6.03e+04	8.99e+04	9.27e+04

5.5.4 KdV equation

In the training of the KdV equation, the convergence of training and testing loss is shown in Fig. 5.10. For all noise levels, the training and testing loss converge synchronously, indicating that the model is performing similarly well on both training and testing datasets. Unlike the previous two equations, both training and testing loss demonstrates larger instability during training, which is likely due to the learning rate setting and choice of optimizer. We

further visualize the predicted solution of the KdV equation in Fig. 5.11. The solution of KdV equation is nicely predicted via neural network. Table 5.11 and Table 5.12 presents the error evolution of mesh-free SINDy. From Table 5.11, we discover that with larger sample sizes, the error of mesh-free SINDy uniformly decreases and the model becomes more robust to noise. In Table 5.12, we provide more evaluation metrics to assess the performance of mesh-free SINDy with increasing noise levels.

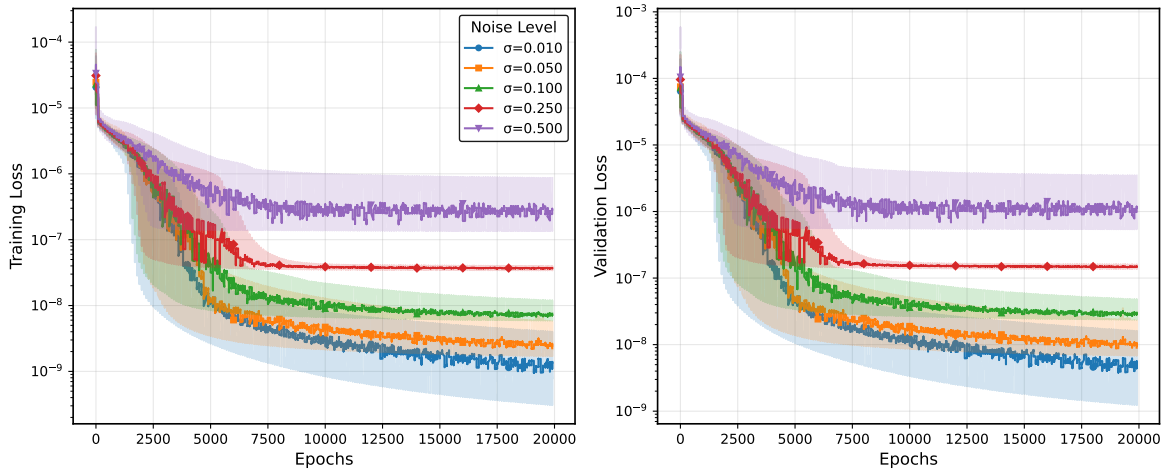


Figure 5.10: Epoch-wise training and validation loss convergence with different noise levels for the KdV equation.

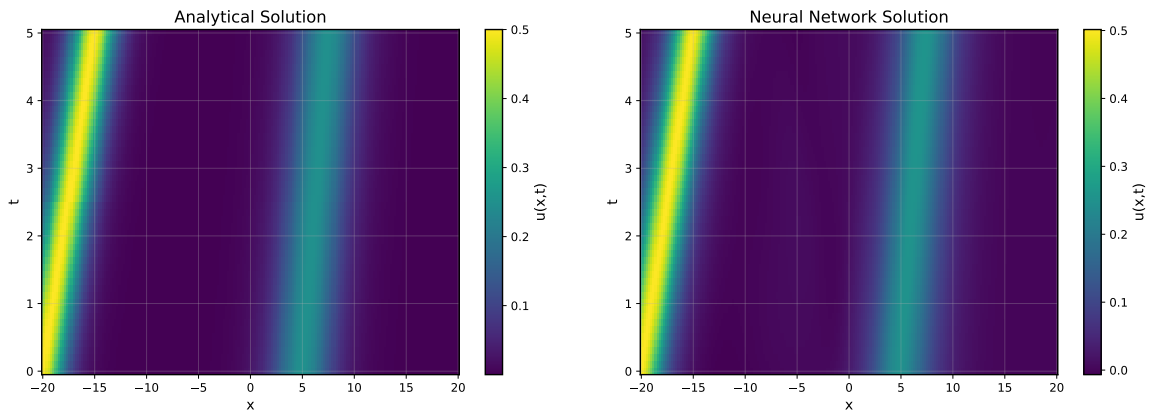


Figure 5.11: Real (left) vs. predicted (right) solution of the KdV equation.

Table 5.11: Loss of mesh-free SINDy under different sample sizes and noise levels for the KdV equation.

Samples	1% Noise	5% Noise	10% Noise	25% Noise
100	83.69	96.37	119.89	121.28
500	59.00	59.67	59.00	59.54
1000	60.09	59.30	59.22	59.62
2000	61.13	59.60	58.92	58.49

Table 5.12: Error metrics of SINDy under different noise levels with 1000 samples for the KdV equation.

Metrics	1% Noise	5% Noise	10% Noise	25% Noise	50% Noise
E_{PDE}	1.76e-03	1.72e-03	1.68e-03	1.77e-03	1.65e-03
E_{NN}	41.9907	44.2752	42.6268	43.8201	44.7633
E_{dudt}	55.6690	56.4196	57.4353	58.6451	63.4641
E_{SINDy}	58.2436	56.9515	57.6004	59.1942	63.6559
E_{field}	43.1152	45.3267	43.6592	44.6639	45.4288

5.5.5 Advection-diffusion equation

In the training of the advection-diffusion equation, the convergence of training and testing loss is shown in Fig. 5.12. For all noise levels, the training and testing loss converge synchronously, indicating that the model is performing similarly well on both training and testing datasets. Since we uniformly set the learning rate for all experiments, the training and testing loss for 1% noise level demonstrates minor instability. We further visualize the predicted solution of the advection-diffusion equation in Fig. 5.13. The solution of advection-diffusion equation is nicely predicted via neural network. Table 5.13 and Table 5.14 presents the error evolution of mesh-free SINDy. From Table 5.13, we discover that with larger sample sizes, the error of mesh-free SINDy uniformly decreases and the model becomes more robust to noise. In Table 5.14, we provide more evaluation metrics to assess the performance of mesh-free SINDy with increasing noise levels.

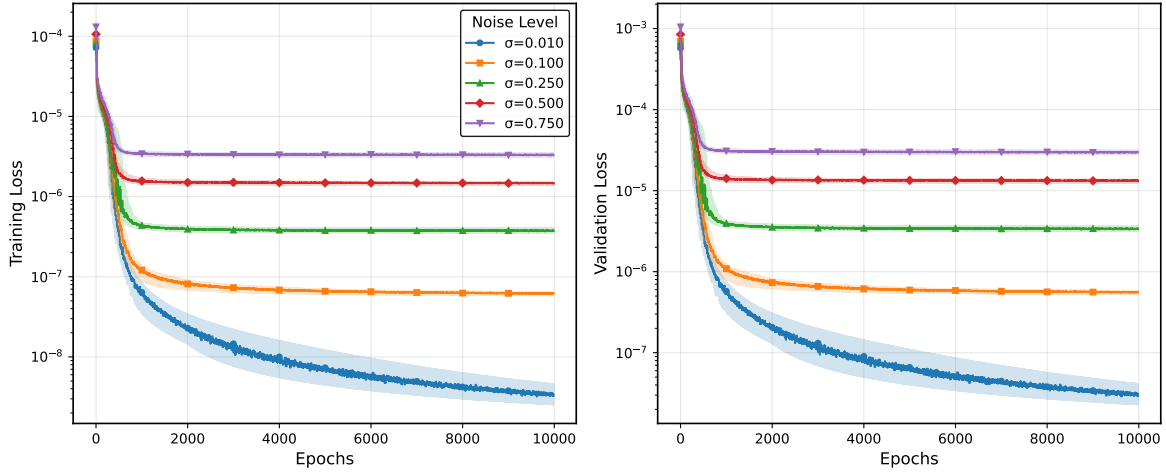


Figure 5.12: Epoch-wise training and validation loss convergence with different noise levels for the advection-diffusion equation.

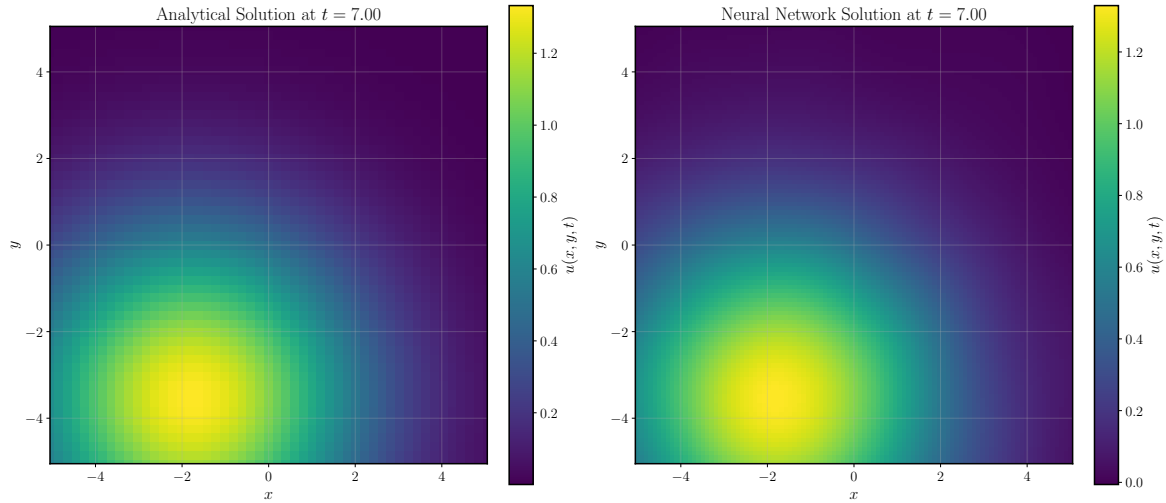


Figure 5.13: Real (left) vs. predicted (right) solution of the advection-diffusion equation.

Table 5.13: Loss of mesh-free SINDy under different sample sizes and noise levels for the advection-diffusion equation.

Samples	1% Noise	10% Noise	25% Noise	50% Noise
500	905.87	919.62	920.59	1003.33
2000	916.12	916.99	923.89	922.10
4000	913.72	924.95	951.24	951.34
6000	929.11	951.23	906.92	934.35

Table 5.14: Error metrics of SINDy under different noise levels with 6000 samples for the advection-diffusion equation.

Metrics	1% Noise	10% Noise	25% Noise	50% Noise	75% Noise
E_{PDE}	1.42e+03	1.45e+03	1.38e+03	1.40e+03	1.39e+03
E_{NN}	0.1896	0.1915	0.1834	0.1889	0.1905
E_{dudt}	931.10	951.35	904.72	936.53	926.19
E_{SINDy}	929.11	951.23	906.92	934.35	917.25
E_{field}	4.03e+03	4.08e+03	4.03e+03	4.26e+03	4.40e+03

6

Uncertainty Quantification of Shallow Recurrent Decoder Networks for Sparse Sensing via Engression

6.1 Introduction

Reconstructing high-dimensional spatiotemporal fields from limited and sparse sensor measurements is a fundamental problem across the physical, biological and engineering sciences. In each of these settings, the reconstruction problem is highly underdetermined, as the spatial dimensionality vastly exceeds the number of available sensor measurements. Data-driven approaches to sparse sensing have advanced considerably in recent years [190], with deep learning architectures demonstrating strong empirical performance on high-dimensional reconstruction tasks [86, 85, 310]. The SHallow REcurrent Decoder (SHRED) network [292] takes a distinct approach by leveraging the temporal history of a hyper-sparse number of sensor measurements into a reduced-order latent space, which is grounded in Takens' delay embedding theorem [264]. SHRED thus enables sparse spatial sensing from as few as three sensors placed anywhere in the dynamics. This hyper-sparse sampling and sensor agnostic placement strategy have been demonstrated on a number of scientific applications such as fluid dynamics [293], seismology [206], neuroscience [232], plasma physics [161], and remote sensing [316]. The SHRED architecture is flexible and can also be modified for model discovery [103] and data assimilation [11]. Challenges remain however, especially as point estimations are fundamentally limited for downstream tasks such as risk assessment, anomaly detection, and decision-making under uncertainty. Therefore, it is essential to extend the

SHRED architecture to quantify the uncertainty with validity in the sensor-to-state mapping, enabling uncertainty-aware analysis for reliable and safety-critical downstream scientific applications. Despite its importance, obtaining reliable uncertainty estimates in this setting is highly nontrivial. The difficulties come from the fact that (i) the use of blackbox learning architectures introduces nonlinear mappings, (ii) recurrent units accumulate and amplify uncertainty over time, and (iii) many existing approaches to distributional learning require substantial data and computational resources (e.g. ensembling) while often lacking formal statistical guarantees on the resulting predictive distribution.

In this paper, we introduce UQ-SHRED, a distributional learning framework for sparse sensing that enables valid uncertainty quantification by learning the full conditional distribution of the spatial states given observed sensor measurements. It is achieved by concatenating Gaussian noise vectors to the input and training the model using an energy score loss to directly optimize a proper scoring rule for distributional predictions inspired by Shen and Meinshausen [245]. Since the noise is injected at the input, it propagates through the temporal unit and shallow decoder, allowing uncertainty to be modeled throughout the full mapping from sparse sensors to spatial states without architectural modification. At inference time, UQ-SHRED generates samples from the conditional predictive distribution by drawing samples of the input noise distribution and propagating these samples through the network. The resulting empirical distribution enables estimation of statistical quantities of interests, including the conditional mean, variance, and prediction intervals. Crucially, UQ-SHRED requires only a single trained network which avoids the computational cost of training multiple separate units for distributional learning. The method further admits theoretical guarantees on the derived conditional distribution and the quantile estimates.

We evaluate our method on five complex real-world datasets including sea-surface temperature data from NOAA [225], isotropic turbulent flow from JHADB [170], neural data from the Allen Institute [247], solar activity from NASA Solar Dynamics Observatory [217], and propulsion physics [12]. These datasets provide a broad testbed for uncertainty quantification in sparse recovery, spanning different spatial structures, sensing regimes, and system dynamics. We validate the UQ-SHRED architecture and its robustness across these scientific

applications. Furthermore, we present an ablation study on the SST dataset to investigate the sensitivity of calibration quality to temporal lag, sensor count, noise dimension, Monte Carlo sample size, and training duration. The ablation study further illustrates the behavior of the uncertainty estimates produced by UQ-SHRED under these changing, potentially non-ideal hyperparameter settings. To summarize, our UQ-SHRED contributions are as follows:

1. We develop a distributional learning framework for sparse sensing that provides valid uncertainty quantification. Inspired by engression, UQ-SHRED combines noise injection and energy score minimization for learning the predictive distribution across the temporal unit and shallow decoder of the SHRED architecture, while preserving the computational efficiency of a single-network approach.
2. We establish the theoretical validity of UQ-SHRED under regularity conditions, showing that the learned conditional distribution supports valid estimation of common statistical quantities of interest. This distinguishes UQ-SHRED from purely heuristic uncertainty estimates and provides theoretical support for its use in uncertainty-aware sparse recovery.
3. We verify the effectiveness of UQ-SHRED on a wide range of scientific problems including sea-surface temperature, turbulent flows, neural activity, solar activity, and propulsion physics. This demonstrates that UQ-SHRED remains effective across diverse applications with remarkably different physics.

The manuscript is organized as follows: In Sec. 6.2, we detail the two critical mathematical structures of UQ-SHRED and highlight some of the current methods being used for UQ in deep learning. This is followed in Sec. 6.3 with the development of the UQ-SHRED architecture along with theoretical estimates related to the methodology. Section 6.4 deploys the UQ-SHRED architecture across a diversity of real-world data sets, explicitly demonstrating the quality of reconstructions and the accuracy of the theoretical estimates derived. An outlook and review of the findings is given in Sec. 6.5.

6.2 Preliminaries

Before presenting our UQ-SHRED architecture, the two mathematical architectures required are detailed. Moreover, we also give a brief review of other existing methods for extracting UQ metrics using deep learning.

6.2.1 The SHRED Architecture

As noted earlier, reconstructing high-dimensional spatiotemporal states from sparse sensor measurements is a core problem in scientific computing and engineering [158]. The SHRED architecture [293, 307] addresses this by exploiting the temporal structure at sensor locations to infer the full spatial field. The approach draws on Takens' delay embedding theorem [264], which guarantees that, under generic conditions, a sufficiently long time-history of a scalar observable provides a diffeomorphic representation of the underlying dynamical attractor. Concretely as it relates to SHRED, given p sensors at fixed spatial locations, a sliding window of L consecutive measurements $\mathbf{S}_{t-L+1:t} \in \mathbb{R}^{L \times p}$ is fed into a multi-layer LSTM [128] that produces a latent code $\mathbf{z}_t \in \mathbb{R}^{d_z}$ from its final hidden state. A shallow decoder then maps $\mathbf{z}_t$ to the full spatial field $\hat{\mathbf{u}}_t \in \mathbb{R}^n$ with $n \gg p$.

Two properties of this architecture are particularly relevant for uncertainty quantification. First, the sensor-to-state mapping is inherently underdetermined: p sensor values do not uniquely specify n spatial degrees of freedom, and the LSTM-based structure imposes a finite-dimensional information bottleneck through d_z . Any deterministic SHRED model selects a single point estimate from the set of states consistent with the observed sensors, discarding information about the remaining ambiguity. Second, the temporal unit processes the lag window sequentially, meaning its hidden state serves as a compressed summary of the recent dynamical history. Injecting controlled stochasticity into this process—as we propose in Section 6.3.1—transforms the deterministic point estimate into a distribution over plausible reconstructions, with the spread reflecting the sensor-to-state ambiguity at each spatiotemporal location.

Several extensions of SHRED have been developed for specific modeling challenges,

including latent-space dynamics identification for general purpose scientific discovery [103], reduced-order modeling [270], and transformer-based temporal units [308]. DA-SHRED [11] addresses the simulation-to-reality gap by updating the latent representations of SHRED using real sensor observations, enabling data assimilation without access to full-field ground-truth data. In this work, we effectively regularize SHRED with the engression framework for reconstruction tasks from sparse observation, though the approach could be compatible with any variant that admits noise injection at the input layer, including multiscale extensions for settings with heterogeneous fields.

6.2.2 Engression and the Energy Score

Engression [245] is a neural network-based distributional learning method that aims to learn the full conditional law $\mathcal{L}(Y \mid X = x)$ rather than only the conditional mean $\mathbb{E}[Y \mid X = x]$. Engression models the conditional distribution through a general conditional generative model of the form

$$Y = g_\theta(X, \varepsilon), \quad (6.1)$$

where $\varepsilon \sim P_\varepsilon$ and $\varepsilon \perp X$. Engression fits a conditional sampler $g_\theta(x, \varepsilon)$ which induces a predictive distribution $P_\theta(\cdot \mid x)$.

For a distribution P on r and observation $z \in r$, let $Z, Z' \stackrel{i.i.d.}{\sim} P$ be two independent random vectors drawn from P . The proper scoring rule used in engression is the Energy Score

$$ES(P, z) = \frac{1}{2} \mathbb{E}_P [\|Z - Z'\|_2] - \mathbb{E}_P [\|Z - z\|_2], \quad Z, Z' \stackrel{i.i.d.}{\sim} P, \quad (6.2)$$

which is a multivariate extension of the continuous ranked probability score (CRPS), and is strictly proper under mild conditions [262, 263]. Plugging in $P = P_\theta(\cdot \mid X)$ yields the population objective:

$$\min_{\theta} \mathbb{E} \left[\|Y - g_\theta(X, \varepsilon)\|_2 - \frac{1}{2} \|g_\theta(X, \varepsilon) - g_\theta(X, \varepsilon')\|_2 \right], \quad \varepsilon, \varepsilon' \stackrel{i.i.d.}{\sim} P_\varepsilon, \quad (6.3)$$

which enables likelihood-free distribution learning for multivariate Y using only samples from g_θ by drawing Monte Carlo samples from $g_\theta(x, \varepsilon_j)$ with $\varepsilon_j \stackrel{i.i.d.}{\sim} P_\varepsilon$.

6.2.3 Uncertainty quantification methods in deep learning

Uncertainty quantification (UQ) for deep learning models is an active area of research with a broad and diverse range of methods for extracting UQ [121, 297]. Bayesian neural networks [324, 202, 27, 271, 191] place prior distributions over network weights and approximate the posterior via variational inference or sampling, providing a principled but often computationally demanding framework. MC-dropout [97] reinterprets dropout at test time as approximate Bayesian inference, offering a practical alternative that requires no architectural changes but lacks formal consistency guarantees for the resulting predictive distribution. Deep ensembles [163, 116] train multiple independent networks with different random initializations and aggregate their predictions; while empirically effective, this approach multiplies the training and inference cost by the ensemble size and does not optimize a proper scoring rule. Heteroscedastic regression models [149] parameterize the output variance as a function of the input, but typically assume a fixed distributional form (e.g., Gaussian) that may be misspecified for complex spatiotemporal systems. More recently, generative approaches including conditional normalizing flows [212, 272] and diffusion models [127, 253] can represent flexible conditional distributions without parametric assumptions on the output, but normalizing flows require invertible network architectures that can limit expressiveness in higher-dimensional output spaces, and diffusion models incur significant computational cost from the iterative denoising procedure required during both training and inference.

6.3 Method: UQ-SHRED

The deterministic SHRED model (introduced in Section 6.2.1) returns a single point estimate $\hat{y}_t$ for each sensor window x_t . However, in regimes characterized by stochastic dynamics, high-frequency content, or insufficient data coverage, portions of the spatiotemporal field cannot be reconstructed with high confidence. In such settings, the quantity of interest is not

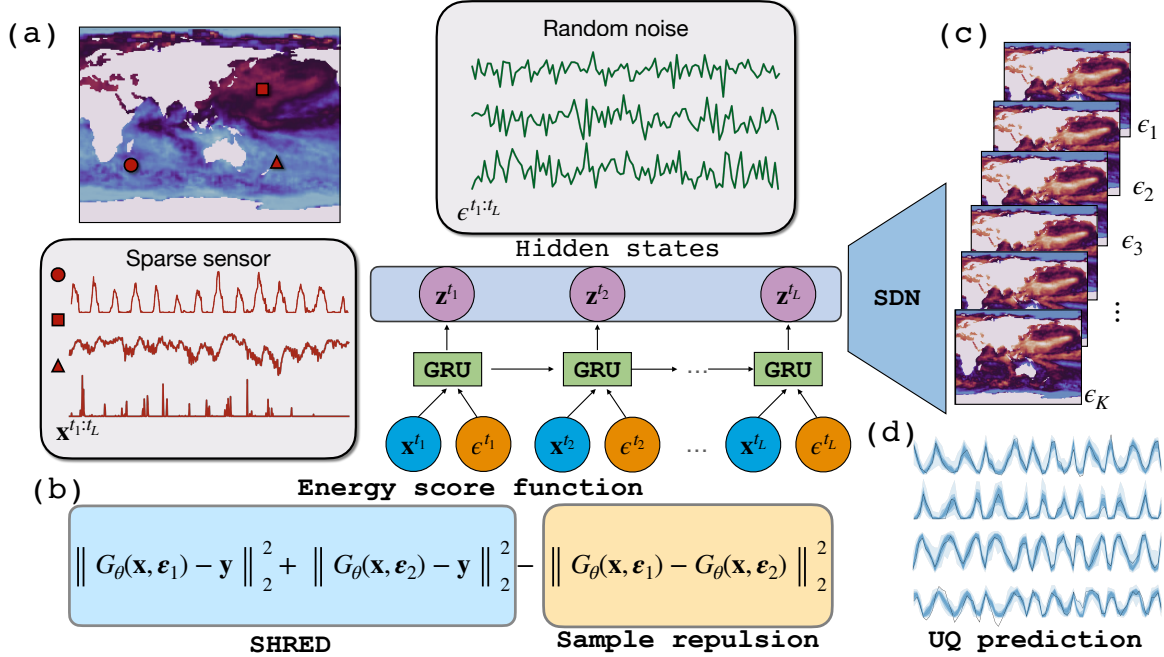


Figure 6.1: (a) UQ-SHRED architecture. The noise vector ϵ is concatenated to the sensor input at each lag step and kept constant across the temporal window. The random noises are concatenated to the sensor input to the GRU, as stated in Eqn. 6.4. (b) The function $G(\cdot)$ represents the SHRED architecture with temporal units and shallow decoder networks. Two independent forward passes with different noise draws produce $G_\theta(x, \epsilon_1)$ and $G_\theta(x, \epsilon_2)$, which define the energy score loss (6.7). (c) At inference, K forward passes with independently resampled noise yield an empirical predictive distribution from which quantile-based confidence intervals are extracted. (d) Uncertainty estimation from the Monte Carlo samples for test sensor locations. Black: ground truth, blue: 95% uncertainty bounds.

just a best-guess reconstruction but the full conditional distribution $P(y_t | x_t)$, from which one can derive prediction intervals, assess risk, and propagate uncertainty into downstream analyses. This section develops UQ-SHRED, which leverages the engression framework 6.2.2 to learn the predictive distribution of the spatial state conditioned on the sparse sensor history.

6.3.1 Architecture

We use engression to enable uncertainty quantification for SHRED with the following architecture. In Figure 6.1 (a), we demonstrate the architecture design of UQ-SHRED. Recall that SHRED maps a lagged sensor window $x_t \in \mathbb{R}^{L \times p}$ to the full state $y_t \in \mathbb{R}^m$ through a composition of a temporal unit (GRU network) and a shallow decoder. To apply engression, at each forward pass, a noise vector $\epsilon \sim \mathcal{N}(0, \mathbf{I}_{d_\epsilon})$ is sampled i.i.d. as part of the input. The augmented input $\tilde{x}_t = [x_t, \epsilon]$ is fed into the temporal unit. The temporal unit produces a

latent representation, and the decoder maps it to the full state:

$$G_\theta(\mathbf{x}, \boldsymbol{\varepsilon}) = f_{\mathcal{D}}(f_{\text{LSTM}}([\mathbf{x}, \boldsymbol{\varepsilon}]); \theta). \quad (6.4)$$

This design is simple but effective which follows the engression algorithm. The training data is injected with random gaussian noise, and the training objective is the following energy score function over the dataset. From (6.3), the energy score objective is

$$\mathcal{R}(\theta) = \mathbb{E} \left[\|\mathbf{y} - G_\theta(\mathbf{x}, \boldsymbol{\varepsilon})\|_2 - \frac{1}{2} \|G_\theta(\mathbf{x}, \boldsymbol{\varepsilon}) - G_\theta(\mathbf{x}, \boldsymbol{\varepsilon}')\|_2 \right], \quad (6.5)$$

where $\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}' \sim \mathcal{N}(0, \mathbf{I}_{d_\varepsilon})$.

The energy score function above in Eqn. 6.5 can be estimated via finite-sample approximation [245] by drawing samples of $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \mathbf{I}_{d_\varepsilon})$ during training. In practice, we approximate this expectation using two i.i.d. noise draws $\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2$ and obtain

$$\hat{\mathbf{y}}_i = f_{\mathcal{D}}(f_{\text{LSTM}}([\mathbf{x}, \boldsymbol{\varepsilon}_i]); \theta). \quad (6.6)$$

Then, we can compute the training loss

$$\mathcal{L}_{\text{ES}}(\theta; \mathbf{x}, \mathbf{y}) = \frac{1}{2} (\|\hat{\mathbf{y}}_1 - \mathbf{y}\| + \|\hat{\mathbf{y}}_2 - \mathbf{y}\|) - \frac{1}{2} \|\hat{\mathbf{y}}_1 - \hat{\mathbf{y}}_2\|. \quad (6.7)$$

The overall objective is

$$\min_{\theta} \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}} \mathcal{L}_{\text{ES}}(\theta; \mathbf{x}, \mathbf{y}). \quad (6.8)$$

Training optimizes a single network that not only reconstructs the target from noise-augmented inputs but, through the repulsive term in the energy score, encourages diverse predictions under different noise realizations. In this way, the model will address the injected noise by maximizing the distance between different predicted variants, which discourages mode collapse and drives the model to approximate the true conditional distribution of the target.

6.3.2 Inference

During inference time, UQ-SHRED can generate samples of $Y \mid X = x$ simply by generating Monte Carlo samples of the noise distribution. For a test input sparse sensor trajectory $\mathbf{x}'$, we have the following steps:

1. Draw K samples from the noise distribution $\boldsymbol{\varepsilon} \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \mathbf{I}_{d_\varepsilon})$ with $\boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_K$.
2. Then $\hat{\mathbf{y}}_k = \mathcal{D}(f_{\text{LSTM}}(\mathbf{x}', \boldsymbol{\varepsilon}_k)), k = 1, \dots, K$ is a sample of the estimated distribution of $Y \mid X = x$.
3. We further obtain estimators through empirical samples $\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_K$. For example, we can obtain the conditional mean from $\mathbb{E}_\varepsilon [\mathcal{D}(f_{\text{LSTM}}(\mathbf{x}', \boldsymbol{\varepsilon}))]$ and conditional α -quantile $\mathbb{Q}_\alpha(\mathcal{D}(f_{\text{LSTM}}(\mathbf{x}', \boldsymbol{\varepsilon})))$. Both quantities are estimated via plug-in estimators from the Monte Carlo samples of $\boldsymbol{\varepsilon}_k$.

Figure 6.1 (c) illustrates this process of probabilistic inference procedure for uncertainty quantification.

6.3.3 Theoretical guarantee

In this subsection, we study theoretical guarantee of UQ-SHRED on distributional learning, conditional mean estimation, and quantile learning.

Theorem 8. *Suppose we have the following:*

(1.a) *The sensor-state pair $(\mathbf{x}, \mathbf{y})$ admits a joint distribution, and for almost every $\mathbf{x} \in \mathbb{R}^{L \times p}$, the conditional law $\mathcal{L}(\mathbf{y} \mid \mathbf{x})$ is well-defined.*

(1.b) *The state has finite first moment $\mathbb{E} [\|\mathbf{y}\|_2] < \infty$.*

(1.c) *The UQ-SHRED architecture $G_\theta(\mathbf{x}, \boldsymbol{\varepsilon})$ induces a conditional law $P_\theta(\cdot \mid \mathbf{x})$, and the model class is well-specified that there exists θ^* such that for almost every $\mathbf{x}$, we have $P_{\theta^*}(\cdot \mid \mathbf{x}) = \mathcal{L}(\mathbf{y} \mid \mathbf{x})$.*

(1.d) *For almost every $\mathbf{x} \in \mathbb{R}^{L \times p}$, each induced conditional law $P_\theta(\cdot \mid \mathbf{x})$, $\theta \in \Theta$, has finite first moment.*

Then, the population minimizer $\hat{\theta} \in \arg \min_{\theta} \mathcal{R}(\theta)$ (defined in (6.5)) satisfies

$$P_{\hat{\theta}}(\cdot | \mathbf{x}) = \mathcal{L}(\mathbf{y} | \mathbf{x}) \quad \forall \mathbf{x} \text{ a.e.} \quad (6.9)$$

Theorem 8 demonstrates that UQ-SHRED is a valid distributional learning method for this sparse sensing problem. Condition (1.a) assumes that we have sufficient sensor history from the Takens's theorem to reconstruct the spatial states. Condition (1.b) ensures the state is finite. Condition (1.c) assumes that there exists a minimizer within the model class that can represent the true distributional law $\mathcal{L}(\mathbf{y} | \mathbf{x})$. Condition (1.d) ensures that the model-induced conditional laws have finite first moments, so that the energy score is well defined and strictly proper.

Proof. Conditioning on a fixed $\mathbf{x}$, from the law of total expectation, the population risk can be written as

$$\mathcal{R}(\theta) = \mathbb{E}_{\mathbf{x}} [\mathcal{R}_{\mathbf{x}}(\theta)] \quad (6.10)$$

$$= \mathbb{E}_{\mathbf{x}} \left[\|\mathbf{y} - G_{\theta}(\mathbf{x}, \boldsymbol{\varepsilon})\|_2 - \frac{1}{2} \|G_{\theta}(\mathbf{x}, \boldsymbol{\varepsilon}) - G_{\theta}(\mathbf{x}, \boldsymbol{\varepsilon}')\|_2 \middle| \mathbf{x} \right] \quad (6.11)$$

Conditional on $X = \mathbf{x}$, the random variable $G_{\theta}(\mathbf{x}, \boldsymbol{\varepsilon})$ follows the distribution $P_{\theta}(\cdot | \mathbf{x})$, while $\mathbf{y}$ follows the true conditional law $\mathcal{L}(\mathbf{y} | X = \mathbf{x})$. Therefore $\mathcal{R}_{\mathbf{x}}(\theta)$ coincides with the energy-score risk of the predictive distribution $P_{\theta}(\cdot | \mathbf{x})$ relative to the true conditional distribution $\mathcal{L}(\mathbf{y} | X = \mathbf{x})$.

By condition (1.d), for almost every $\mathbf{x}$, all $P_{\theta}(\cdot | \mathbf{x})$ have finite first moments. Hence the energy score is strictly proper on this pair of conditional laws [262, 263]. As the energy score is a strictly proper scoring rule, the energy-score risk satisfies

$$\mathcal{R}_{\mathbf{x}}(Q) \geq \mathcal{R}_{\mathbf{x}}(P_{\mathbf{x}}) \quad (6.12)$$

for any predictive distribution Q , with equality iff $Q = P_{\mathbf{x}}$. Taking $Q = P_{\theta}(\cdot | \mathbf{x})$ therefore

implies that

$$\mathcal{R}_x(\theta) \tag{6.13}$$

is minimized if and only if

$$P_\theta(\cdot | x) = \mathcal{L}(y | X = x). \tag{6.14}$$

It means that given x , we must have $P_\theta(\cdot | x) = \mathcal{L}(y | X = x)$ to be the minimizer.

We then show further this holds for almost every x by contradiction.

By assumption (1.c), there exists θ^* such that

$$P_{\theta^*}(\cdot | x) = \mathcal{L}(y | X = x), \text{ for almost every } x, \tag{6.15}$$

so θ^* minimizes the population risk.

Now suppose $\hat{\theta}$ is a population minimizer and

$$P_{\hat{\theta}}(\cdot | x) \neq \mathcal{L}(y | X = x) \tag{6.16}$$

on a set of x with positive measure, then strict propriety implies that

$$\mathcal{R}_x(\hat{\theta}) > \mathcal{R}_x(\theta^*) \tag{6.17}$$

on that set. Taking expectation over X yields

$$\mathcal{R}(\hat{\theta}) > \mathcal{R}(\theta^*). \tag{6.18}$$

Then this is no longer a population minimizer. Contradiction!

Therefore, we have

$$P_{\hat{\theta}}(\cdot | x) = \mathcal{L}(y | X = x), \text{ for almost every } x. \tag{6.19}$$

□

Remark 1. *Distributional approximation of UQ-SHRED under model misspecification.*

Because the energy score is a strictly proper scoring rule, the population minimizer under misspecification satisfies

$$\hat{\theta} \in \arg \min_{\theta \in \Theta} \mathcal{R}(\theta) = \arg \min_{\theta \in \Theta} \mathbb{E}_{\mathbf{x}} [\text{ES}(P_{\theta}(\cdot | \mathbf{x}), \mathcal{L}(\mathbf{y} | \mathbf{x}))], \quad (6.20)$$

where $\text{ES}(Q, P)$ denotes the expected energy score of the predictive distribution Q evaluated against draws from P . Since strict propriety implies that $\text{ES}(Q, P) \geq \text{ES}(P, P)$ with equality if and only if $Q = P$, the population minimizer $P_{\hat{\theta}}(\cdot | \mathbf{x})$ is the *closest* element of $\mathcal{P}_{\Theta}$ to the true conditional law in the sense of minimizing the expected energy score divergence

$$d_{\text{ES}}(Q, P) := \text{ES}(Q, P) - \text{ES}(P, P) = \frac{1}{2} \mathbb{E} [\|Z - Z'\|_2] - \mathbb{E} [\|Z - W\|_2] + \frac{1}{2} \mathbb{E} [\|W - W'\|_2], \quad (6.21)$$

where $Z, Z' \stackrel{i.i.d.}{\sim} Q$ and $W, W' \stackrel{i.i.d.}{\sim} P$, which is nonnegative and equals zero if and only if $Q = P$.

Thus, even when condition (1.c) does not hold exactly, UQ-SHRED returns the best distributional approximation within its model class as measured by the energy score. The practical consequence is that the predictive distribution may not achieve exact calibration, but it remains the optimal approximation available to the architecture, and the degree of miscalibration is controlled by the approximation capacity of the model class $\mathcal{P}_{\Theta}$.

Corollary 1. *Suppose that conditions (1.a)-(1.d) of Theorem 8 hold. Additionally, we assume*

(1.e) *The state has finite second moment $\mathbb{E} [\|\mathbf{y}\|_2^2] < \infty$.*

Then any population minimizer $\hat{\theta}$ in Theorem 8 satisfies

$$\mathbb{E}_{P_{\hat{\theta}}} [\mathbf{y} | \mathbf{x}] = \mathbb{E} [\mathbf{y} | \mathbf{x}] \quad \text{for almost every } \mathbf{x},$$

and

$$\text{Cov}_{P_{\hat{\theta}}}(\mathbf{y} \mid \mathbf{x}) = \text{Cov}(\mathbf{y} \mid \mathbf{x}) \quad \text{for almost every } \mathbf{x}.$$

Proof. This is a direct result of Thm. 8. We know under θ^* that we obtain the true law $\mathcal{L}(\mathbf{y} \mid \mathbf{x})$. Therefore, the expectation and covariance will be the same as the expectation and covariance given the true law, from condition (1.e) with finite second moment. $\square$

Theorem 9. For a test input $\mathbf{x} \in \mathbb{R}^{L \times p}$, a coordinate index $j \in \{1, \dots, m\}$, and a level $\alpha \in (0, 1)$. We generate K Monte Carlo samples $\hat{\mathbf{y}}^{(k)} = f_{\mathcal{D}}(f_{\text{LSTM}}([\mathbf{x}, \boldsymbol{\varepsilon}_k]); \theta)$, $\boldsymbol{\varepsilon}_k \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_{d_{\varepsilon}})$. Let $F_{\theta,j}(\cdot \mid \mathbf{x})$ denote the conditional CDF of the j -th coordinate under the law $P_{\theta}(\cdot \mid \mathbf{x})$. Define the coordinate-wise predictive α -quantile by

$$q_{\theta,j,\alpha}(\mathbf{x}) = \inf\{t \in \mathbb{R} : F_{\theta,j}(t \mid \mathbf{x}) \geq \alpha\}.$$

Suppose we have

(2.a) The distribution function $F_{\theta,j}(\cdot \mid \mathbf{x})$ is continuous at $q_{\theta,j,\alpha}(\mathbf{x})$ and strictly increasing in a neighborhood of $q_{\theta,j,\alpha}(\mathbf{x})$.

Define the empirical distribution function and the empirical coordinate-wise α -quantile

$$\hat{F}_{K,j}(t \mid \mathbf{x}) = \frac{1}{K} \sum_{k=1}^K \mathbf{1}\{\hat{Y}_j^{(k)} \leq t\}, \quad \hat{q}_{K,j,\alpha}(\mathbf{x}) = \inf\{t \in \mathbb{R} : \hat{F}_{K,j}(t \mid \mathbf{x}) \geq \alpha\}. \quad (6.22)$$

Then, we have

$$\hat{q}_{K,j,\alpha}(\mathbf{x}) \xrightarrow{a.s.} q_{\theta,j,\alpha}(\mathbf{x}), \quad (6.23)$$

as $K \rightarrow \infty$.

Theorem 9 demonstrates that the Monte Carlo inference algorithm UQ-SHRED is a statistically valid quantile learning method. The theorem is based on conditions that $\boldsymbol{\varepsilon}$ samples are i.i.d. and the CDF is continuous.

Proof. The random variables $\hat{\mathbf{y}}_j^{(1)}, \dots, \hat{\mathbf{y}}_j^{(K)}$ are i.i.d. draws given $\mathbf{x}$ from the one-dimensional predictive distribution with CDF $F_{\theta,j}(\cdot \mid \mathbf{x})$ as $\boldsymbol{\varepsilon}_i$'s are i.i.d. By Glivenko-Cantelli theorem [273],

we have

$$\sup_{t \in \mathbb{R}} \left| \widehat{F}_{K,j}(t \mid \mathbf{x}) - F_{\theta,j}(t \mid \mathbf{x}) \right| \xrightarrow{a.s.} 0. \quad (6.24)$$

For all $\delta > 0$, from (2.a), we have both

$$F_{\theta,j}(q_{\theta,j,\alpha}(\mathbf{x}) - \delta \mid \mathbf{x}) < \alpha, \quad F_{\theta,j}(q_{\theta,j,\alpha}(\mathbf{x}) + \delta \mid \mathbf{x}) > \alpha \quad (6.25)$$

for sufficiently small $\delta > 0$. Since $\widehat{F}_{K,j} \xrightarrow{a.s.} F_{\theta,j}$, we know there exists sufficiently large K that

$$\widehat{F}_{K,j}(q_{\theta,j,\alpha}(\mathbf{x}) - \delta \mid \mathbf{x}) < \alpha, \quad \widehat{F}_{K,j}(q_{\theta,j,\alpha}(\mathbf{x}) + \delta \mid \mathbf{x}) > \alpha, \quad a.s. \quad (6.26)$$

By the definition of the empirical quantile, we know

$$q_{\theta,j,\alpha}(\mathbf{x}) - \delta \leq \widehat{q}_{K,j,\alpha}(\mathbf{x}) \leq q_{\theta,j,\alpha}(\mathbf{x}) + \delta, \quad a.s. \quad (6.27)$$

Since $\delta > 0$ can be arbitrary small, under $K \rightarrow \infty$, we have

$$\widehat{q}_{K,j,\alpha}(\mathbf{x}) \xrightarrow{a.s.} q_{\theta,j,\alpha}(\mathbf{x}). \quad (6.28)$$

□

6.4 Experiments

6.4.1 Evaluation metrics

We assess the quality of the predictive distribution using three complementary metrics.

Calibration. For each nominal confidence level $\alpha \in (0, 1)$, we compute the fraction of ground-truth values that fall within the corresponding α -level prediction interval, averaged over all spatiotemporal locations. We denote this observed coverage $\hat{\alpha}$. A well-calibrated model satisfies $\hat{\alpha} \approx \alpha$ for all levels, which we visualise as a calibration diagram plotting $\hat{\alpha}$

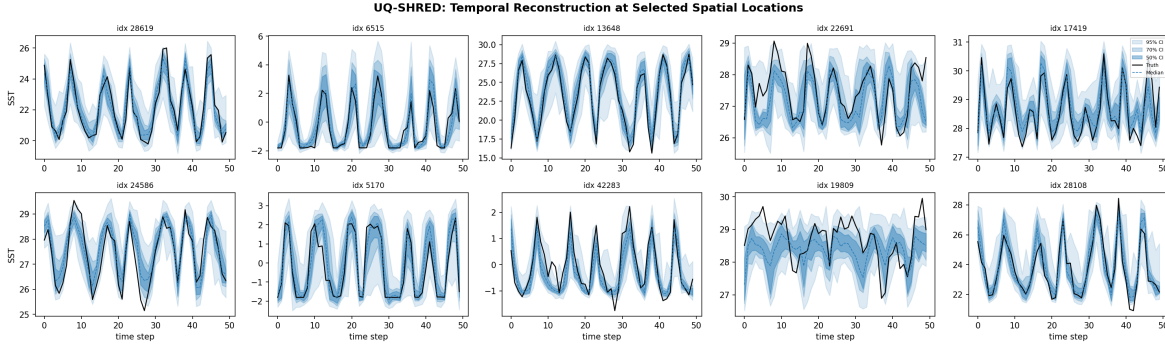


Figure 6.2: UQ-SHRED reconstruction of the SST dataset. Shaded regions: 50%, 70%, and 95% confidence intervals; blue dashed: median; black solid: ground truth.

versus α ; the diagonal represents ideal calibration.

Continuous ranked probability score. The CRPS [111] is a proper scoring rule that jointly assesses calibration and sharpness. For a set of predictive samples $\{\hat{y}^{(k)}\}_{k=1}^K$ and an observation y , the CRPS admits the representation

$$\text{CRPS} = \frac{1}{K} \sum_{k=1}^K |\hat{y}^{(k)} - y| - \frac{1}{2K^2} \sum_{k=1}^K \sum_{j=1}^K |\hat{y}^{(k)} - \hat{y}^{(j)}|, \quad (6.29)$$

which we average over all spatial locations and time steps. Lower values indicate better probabilistic forecasts.

Sharpness. The average width of the prediction interval at each confidence level, measuring the concentration of the predictive distribution. Conditional on correct calibration, narrower intervals (sharper forecasts) are preferred.

6.4.2 Sea-surface temperature data

Dataset The Sea-surface temperature (SST) dataset contains 1,400 snapshots of weekly averaged sea surface temperature from 1992 to 2019 reported by NOAA [225]. The dataset is a reanalysis dataset through interpolation combining in-situ buoy, ship, and satellite 429 observations and every snapshot includes a 180×360 grid with 44,219 grid points

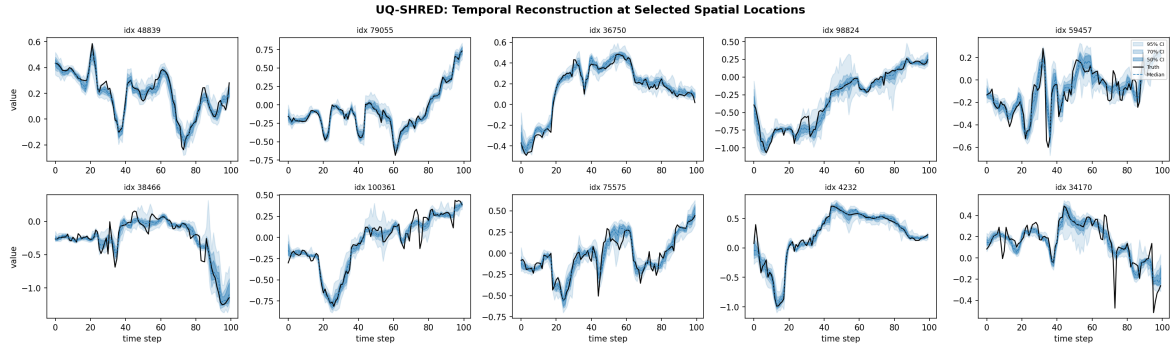


Figure 6.3: UQ-SHRED reconstruction of the isotropic turbulent flow. Shaded regions: 50%, 70%, and 95% confidence intervals; blue dashed: median; black solid: ground truth.

corresponding to the sea-surface.

We train UQ-SHRED with global SST data from only 3 random sensors, with temporal lag $L = 52$ (approximately one year of history), noise dimension $d_\varepsilon = 1000$, and 200 Monte Carlo samples at inference to estimate the quantiles. Figure 6.2 presents reconstruction from randomly selected sensors and uncertainty quantification on test data. The median prediction tracks the ground truth (black) across both fields. The confidence bands widen at the global temperature shifts, where the rate of change is highest and sparse sensors provide less constraint on the global field.

6.4.3 Isotropic turbulent flow

Dataset The isotropic turbulent flow dataset is a direct numerical simulation of forced isotropic turbulence from the Johns Hopkins Turbulence Database (JHTDB) [170]. It is generated from the incompressible Navier-Stokes equation using pseudo-spectral method with periodic boundary conditions on a 1024^3 grid. We extract 2D 350×350 spatial slices of the pressure field, yielding 1,667 temporal samples.

We train UQ-SHRED with 3 random sensors, $L = 100$, noise dimension $d_\varepsilon = 100$, and 50 Monte Carlo samples at inference. Figure 6.3 presents reconstruction and uncertainty quantification on test data. The confidence bands widen during rapid local pressure fluctuations, reflecting increased reconstruction ambiguity when the flow undergoes violent spatiotemporal changes.

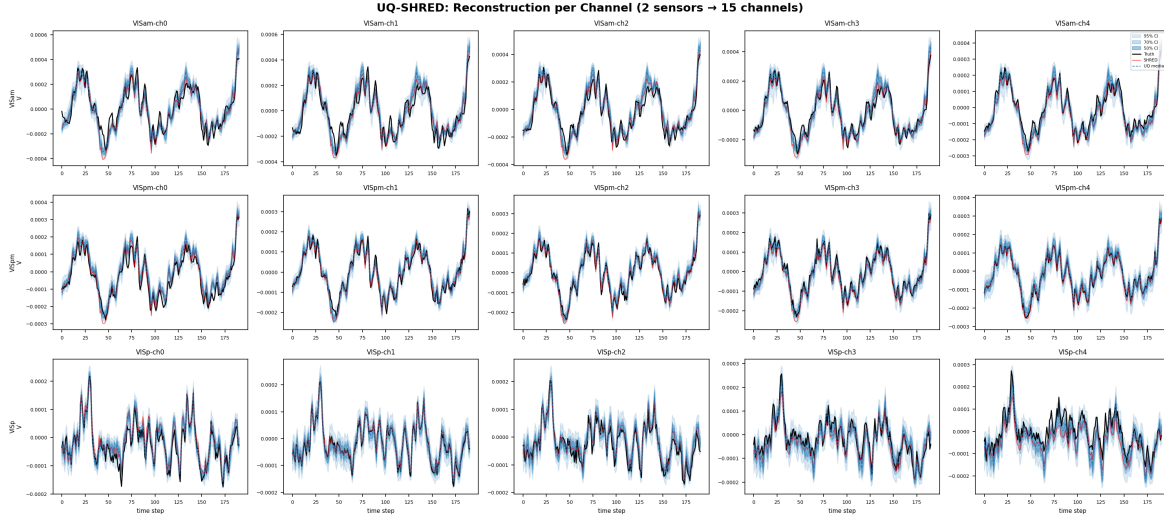


Figure 6.4: UQ-SHRED reconstruction of the neural activity dataset. Shaded regions: 50%, 70%, and 95% confidence intervals; blue dashed: median; black solid: ground truth.

6.4.4 Neural activity data

Dataset The neural data was obtained from the Allen Institute Visual Coding—Neuropixels project [247]. This dataset contains extracellular electrophysiology recordings from the mouse visual cortex and thalamus during a variety of visual stimuli. We analyzed 0.5 kHz local field potential (LFP) recordings from the anteromedial (VISam), posteromedial (VISpm), and primary (VISp) visual cortex during presentation of a 2-second 180° drifting grating stimulus. Rather than recording activity from a single unit, LFPs measure population activity from ensembles of neurons [141]. While LFPs are valuable for studying network dynamics, they require invasive recording techniques and are susceptible to noise and interference, which introduce additional ambiguity into the signals [81]. Our aim is to train UQ-SHRED on this dataset to reconstruct neural population activity from a limited number of sensors while providing valid uncertainty quantification.

We train UQ-SHRED on LFP data with 2 randomly selected channels, $L = 100$, noise dimension $d_\epsilon = 50$, and 100 Monte Carlo samples at inference. Figure 6.4 presents reconstruction and uncertainty quantification on test data. The confidence bands widen at high-frequency local perturbations in the neural signal, consistent with the increased ambiguity introduced by noise and interference at finer temporal scales.

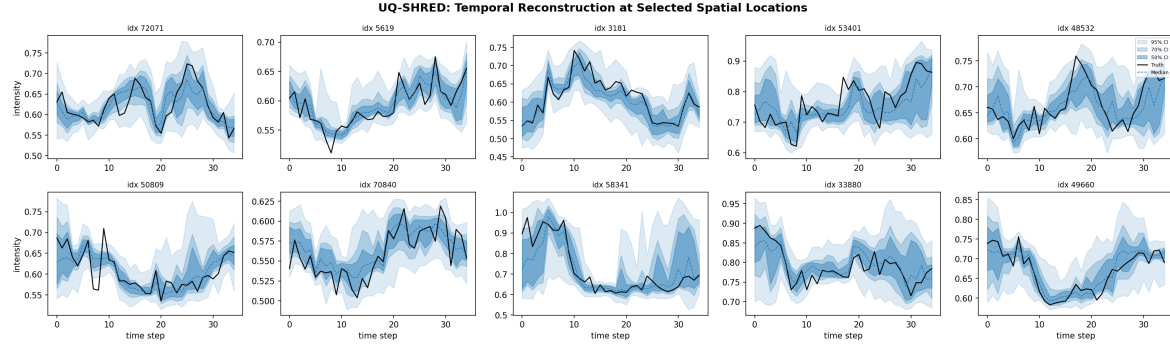


Figure 6.5: UQ-SHRED reconstruction of the solar activity dataset. Shaded regions: 50%, 70%, and 95% confidence intervals; blue dashed: median; black solid: ground truth.

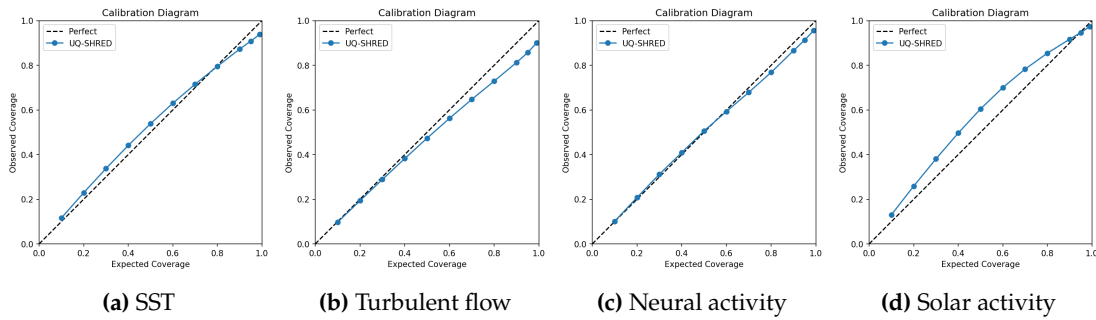


Figure 6.6: Calibration diagrams for UQ-SHRED across all four datasets. Dashed line: perfect calibration. Blue: UQ-SHRED empirical coverage.

6.4.5 Solar activity data

Dataset The solar activity data describes active physical and chemical reactions on the sun. We obtain the solar data from publicly available dataset from NASA. The Solar Dynamics Observatory collects observations of the sun, and we particularly focus on the wavelength 171 Ångstroms, which is known to be the most active and commonly used part of the sun. It specifically highlights a spectral line emitted by iron atoms that have lost 8 electrons at temperatures of 600,000 K. We collected 274 frames of an active area of the sun with 30 minutes temporal gap of each frame.

We train UQ-SHRED with 3 randomly selected sensors, $L = 75$, noise dimension $d_\epsilon = 200$, and 100 Monte Carlo samples at inference. Figure 6.5 presents reconstruction and uncertainty quantification on test data. The confidence bands widen during rapid changes in solar irradiance, particularly around transient flare events.

Table 6.1: UQ-SHRED metrics across the four experimental datasets. Calibration reports observed coverage (%) at each nominal confidence level. CRPS represents the continuous ranked probability score. Sharpness is the average CI width.

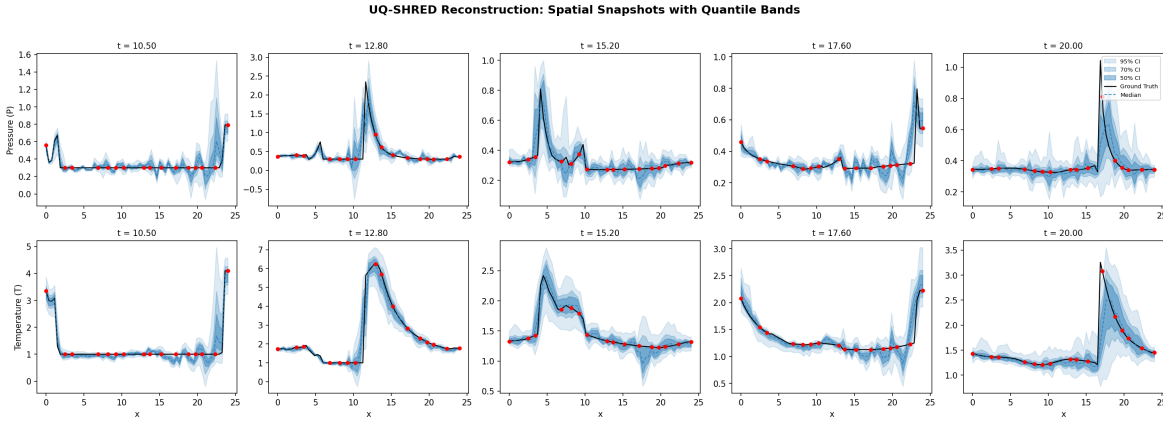
Dataset	RMSE	Calibration % (expected to observed)					Sharpness		
		50%	70%	90%	95%	99%	CRPS	95% CI	50% CI
SST	0.656	53.9	71.5	87.3	90.8	93.9	0.343	2.791	0.890
ISO	0.656	47.3	64.7	81.2	85.7	90.1	0.032	0.209	0.067
Neural	$3.54e^{-5}$	50.6	67.9	86.6	91.3	95.5	$1.8e^{-5}$	$1.14e^{-4}$	$3.9e^{-5}$
Solar	0.029	60.4	78.3	91.7	94.6	97.3	0.016	0.139	0.051

Summary across datasets. Table 6.1 reports the calibration, CRPS, and sharpness metrics for all four datasets. Across all experiments, the median prediction of UQ-SHRED tracks the ground truth closely. The observed coverages at the 50%, 70%, 90%, 95%, and 99% nominal levels are consistent with well-calibrated predictive distributions, as shown in the calibration diagrams in Figure 6.6. In each case, the confidence intervals are not uniformly wide but vary spatially and temporally: they widen where reconstruction ambiguity is highest and narrow where sensor data sufficiently constrains the state. Additional spatial reconstruction visualizations are provided in Appendix 6.6.

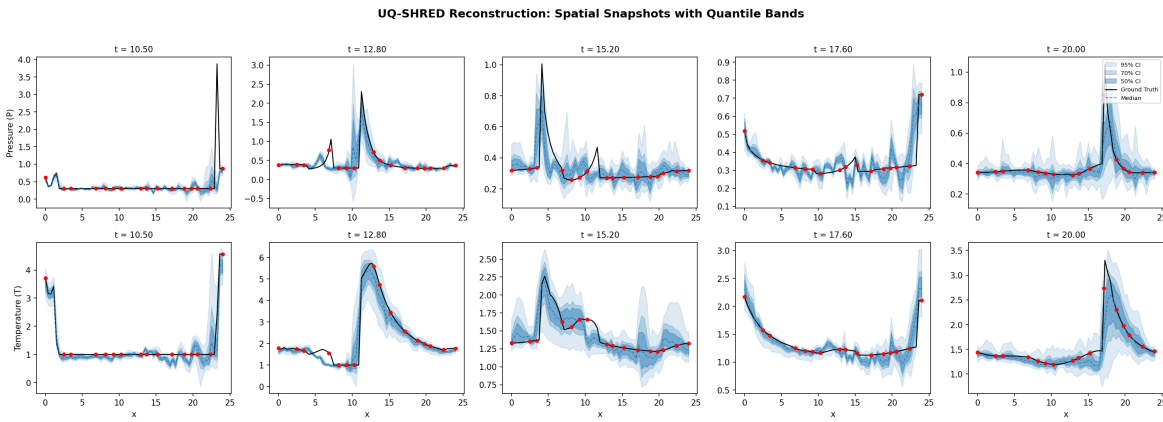
6.4.6 1D RDE transient Stage

Dataset We consider the ignition transient of a one-dimensional rotating detonation engine (RDE) generated by Koch’s reduced-order model [153], which approximates detonation-wave propagation in an annular combustor via the reactive Euler equations with source terms for injection, mixing, and finite-rate chemistry. The transient stage—in which the flow transitions from a quiescent premixed state to sustained detonation—produces sharp, rapidly evolving pressure and temperature fronts whose positions and amplitudes vary across initial conditions. We reconstruct both the pressure $P(x, t)$ and temperature $T(x, t)$ fields over the temporal window $t \in [10.0, 20.0]$ ($T = 101$ frames) on a downsampled spatial grid of $m_x = 64$ points.

We train UQ-SHRED from scratch on one ground-truth trajectory using the energy score loss, with $p = 16$ uniformly spaced sensors, noise dimension $d_\epsilon = 50$, and 200 Monte Carlo



(a) Closer trajectory (Run 0, small IC perturbation from training data).



(b) Further trajectory (Run 1, larger IC perturbation).

Figure 6.7: UQ-SHRED reconstruction of the 1D RDE transient stage at selected time instances. Top rows: pressure; bottom rows: temperature. Shaded regions: 50%, 70%, and 95% confidence intervals; red dashed: median; black solid: ground truth; red dots: sensor locations.

samples at inference. We evaluate on two held-out simulation trajectories that differ in the magnitude of their initial-condition perturbation from the training trajectory: a *closer* trajectory (Run 0, small perturbation) and a *further* trajectory (Run 1, larger perturbation). This pair tests how the learned uncertainty responds to distributional shift.

Figure 6.7 presents spatial snapshots at selected time instances for both cases. The median prediction tracks the ground truth across both fields, while the 50%, 70%, and 95% confidence bands widen at the detonation wave fronts—where sparse sensors provide the least constraint on the full-field reconstruction—and narrow in the quiescent post-detonation regions where the state is well-determined. Table 6.2 reports the calibration, CRPS, and sharpness metrics for both test cases alongside the deterministic SHRED baseline RMSE. Additional figs are

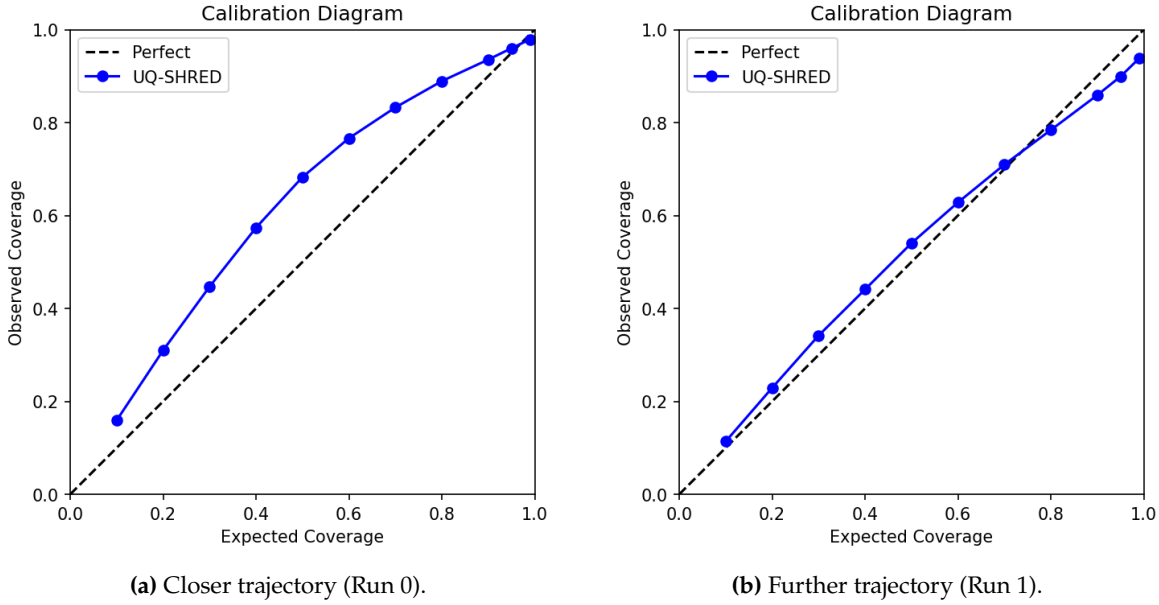


Figure 6.8: Calibration diagrams for UQ-SHRED on the 1D RDE transient stage. Dashed line: perfect calibration. Blue: UQ-SHRED empirical coverage.

Table 6.2: UQ-SHRED metrics on the 1D RDE transient stage for two test trajectories with different initial-condition perturbation magnitudes. Calibration reports observed coverage (%) at each nominal confidence level. CRPS represents the continuous ranked probability score. Sharpness is the average CI width.

Test case	RMSE (median)		Calibration % (expected to observed)					Sharpness		
	P	T	50%	70%	90%	95%	99%	CRPS	95% CI	50% CI
Closer (Run 0)	0.156	0.307	69.5	83.5	93.6	95.9	98.0	0.070	0.556	0.185
Further (Run 1)	0.194	0.398	56.9	72.9	88.0	91.5	95.1	0.096	0.569	0.190
<i>Deterministic SHRED baseline:</i>										
Closer (Run 0)	0.145	0.274					—	—	—	—
Further (Run 1)	0.198	0.389					—	—	—	—

provided in Appendix 6.6.

To further investigate the stability of UQ-SHRED under temporal extrapolation, we conduct an experiment in which both UQ-SHRED and deterministic SHRED are trained on the first portion of the ground-truth trajectory ($t \leq 15.0$) and evaluated on the remaining unseen portion ($t > 15.0$). Both methods utilize the same SHRED-architecture with number of LSTM layers varying from 1 to 4, with 10 independent random initializations for each architecture. Figure 6.9 presents the visualizations: both methods produce comparable reconstructions within the training window, but beyond the training boundary (dashed vertical line), the

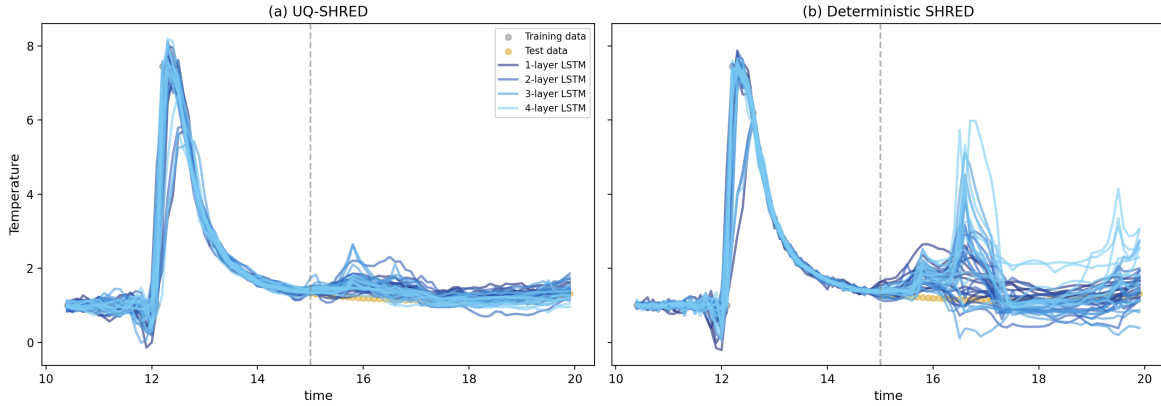


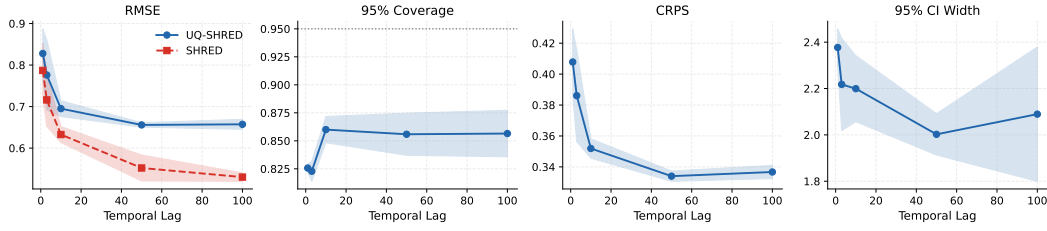
Figure 6.9: Stability comparison between UQ-SHRED and deterministic SHRED under temporal extrapolation on the 1D RDE data. Visualizations are presented for temporal traces at a selected spatial location $x = 14.50$ on the temperature field.

deterministic SHRED predictions diverge substantially across initializations—the variance is on average $3.30\times$ larger than that of UQ-SHRED. This behavior parallels the findings of Shen and Meinshausen [245], who demonstrated that engression produces stable out-of-support predictions where standard regression with identical architectures yields highly variable extrapolations. In the present setting, the energy score loss regularizes the learned mapping so that different initializations converge to consistent predictive distributions, whereas the MSE objective—which is unconstrained beyond the training support—permits each initialization to extrapolate differently. The result suggests that engression not only provides calibrated uncertainty estimates but also confers improved robustness when extrapolating beyond the training time window.

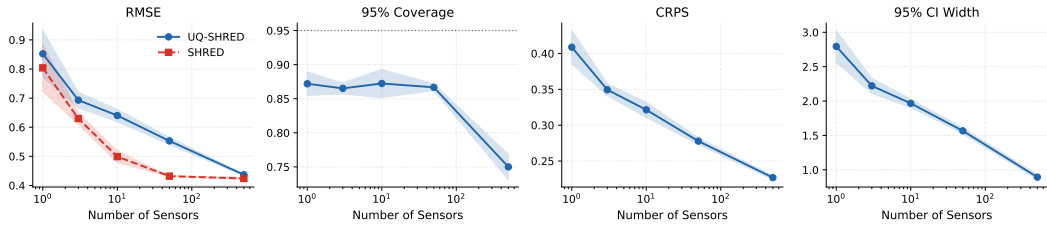
Figure 6.8 shows the calibration diagrams with physically interpretable behavior: the model’s learned uncertainty bandwidth, calibrated on data near the training distribution, becomes slightly insufficient when wavefront positions shift substantially.

6.4.7 Ablation study on SST data

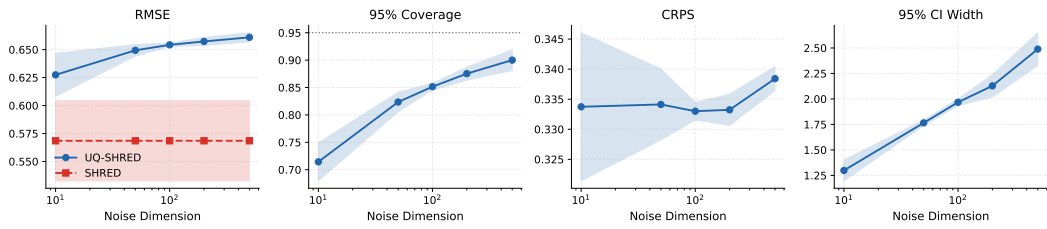
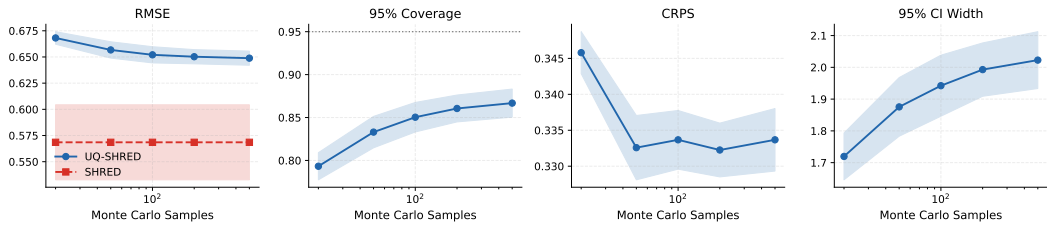
As is common in evaluating neural network architectures, ablation studies are important in order to help understand the contribution of individual components to the model’s overall performance. The ablation study performed here is critical for building interpretability and trust in the UQ-SHRED architecture while allowing for more efficient model design by



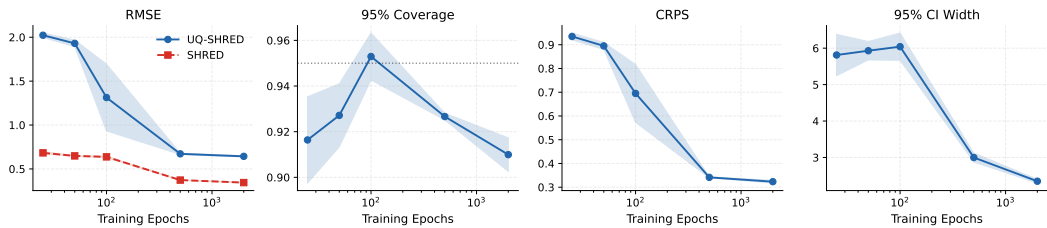
(a) Effect of temporal lag on UQ-SHRED.



(b) Effect of number of sensors on UQ-SHRED.

(c) Effect of the dimension of d_ϵ on UQ-SHRED.

(d) Effect of Monte Carlo sample size on UQ-SHRED.



(e) Effect of the number of training epochs on UQ-SHRED.

revealing what can be simplified without sacrificing accuracy.

Temporal lag parameter In Figure 6.10a, we demonstrate how the temporal lag parameter will affect UQ-SHRED training and uncertainty quantification. Takens' embedding theorem guarantees that a sufficiently long temporal history provides a faithful embedding of the

underlying attractor. Consistent with this, both SHRED and UQ-SHRED exhibit clear RMSE reduction as the lag increases, confirming the importance of temporal history for sparse sensing. We further observe that uncertainty coverage and confidence interval width are relatively insensitive to the temporal lag, particularly beyond $L = 10$.

Sensor number In Figure 6.10b, we demonstrate how the number of sensors will affect UQ-SHRED's performance. For sparse sensing problems, it is clear that with more sensors, the spatial prediction problem will be less challenging. This is supported by the numerical experiments that the testing loss is significantly reduced with more sensors. However, it is also important to notice from the 95% coverage plot that the coverage performance is not improving much with additional sensors; it is likely that we need to increase the noise dimension d_ε for larger input dimensions to address the uncertainty in a higher-dimensional space.

Noise dimension In Figure 6.10c, we show how noise dimension can affect UQ-SHRED coverages. In general, a larger noise dimension helps the model better capture the uncertainty. In practice, we suggest the users to specify the noise dimension d_ε scales with the size of input data. Having a significantly large noise dimension could help, but will also significantly increase the computational burden.

Monte Carlo sample size In Figure 6.10d, we show how Monte Carlo sample size during inference can affect UQ-SHRED coverages. In theory, having more Monte Carlo samples will reduce the bias of the estimate of the quantiles, which is empirically verified here in the 95% coverage plot. A common guideline is to have at least 100 Monte Carlo samples for valid quantile estimation.

Epochs In Figure 6.10e, we show an important observation that UQ-SHRED is able to quantify uncertainty for both overfitted and underfitted models. When the number of training epochs are small, the model is not learning enough information of the system. UQ-SHRED is still able to provide statistically valid uncertainty estimates which has an average of 91.8%

empirical coverage for 95% quantile. When the model is well-trained, it is still able to provide a valid estimate with an average of 91.6%. Figure 6.10e also suggests that applying early stopping methods can be a helpful technique for uncertainty quantification, which essentially avoids the model to overfit onto the noise.

6.5 Conclusion

In this paper, we introduce UQ-SHRED, a distributional learning framework for sparse sensing problems that enables valid uncertainty estimation. This stochastic modeling of uncertainty amends the deterministic reconstruction framework for regimes where the system is partially observed, stochastic, and/or contains high-frequency dynamics that are not captured by the available sensor data. Inspired by engression [245], we concatenate a resampled Gaussian noise vector to the sensor input, and incorporate the energy score loss during training. The resulting model retains the computational efficiency while producing a distributional estimate at each spatiotemporal location through Monte Carlo resampling. Because the energy score is a strictly proper scoring rule, the population optimum is attained when the predictive distribution matches the true conditional distribution under technical assumptions in Thm. 8, and the quantile estimation converges to the true quantiles with large Monte Carlo samples (as demonstrated in Thm. 9).

The experimental study spans five datasets drawn from distinct scientific domains including climate science, computational fluid dynamics, neuroscience, astrophysics, and propulsion physics. These systems differ substantially in their spatiotemporal complexity, dimensionality, noise characteristics, and the degree to which sparse sensors constrain the full-field reconstruction. Across these diverse settings, UQ-SHRED consistently produces well-calibrated confidence intervals while maintaining reconstruction accuracy comparable to the deterministic SHRED baseline. The calibration diagrams show that the observed coverages closely follow the diagonal across all target confidence levels. More importantly, the confidence intervals exhibit spatially and temporally varying width corresponding to the true underlying complexity of the physical interactions. Specifically, the uncertainty estimates

typically widen in regions of higher reconstruction ambiguity such as detonation wave fronts in the RDE, turning points in the SST seasonal cycle, and rapid transients in neural recordings. The uncertainty estimation tends to be narrow where the sensor data sufficiently constrains the reconstruction. These observations all indicate that the learned predictive distribution faithfully reflects the information provided by the sparse sensor measurements.

The ablation study on SST data (Section 6.4.7) reveals further insights in practice. Temporal lag primarily affects reconstruction accuracy through Takens’s embedding theorem but has limited influence on calibration once a minimal embedding dimension is reached. The noise dimension d_ϵ should scale with the input dimensionality to ensure sufficient expressive capacity for the stochastic mapping; a very small noise dimension will lead to underdispersed predictions, while excessively large dimensions increase computational cost without proportional benefit. Most notably, UQ-SHRED maintains valid uncertainty estimates across a wide range of training epochs, including both undertrained and overtrained regimes, suggesting that the energy score loss provides robustness to model misspecification that could be absent from standard point-estimate training.

We note several limitations of the current framework, which can be improved by future works. First, the calibration guarantees of the energy score hold at the population optimum; with finite training data, the learned distribution may exhibit conservative or overconfident behavior depending on the dataset size and complexity, as observed in the RDE experiment where training on a single trajectory could lead to mild miscalibration under strong distributional shift. Additionally, the current framework does not provide coverage guarantees for finite samples, which can be corrected by conformal calibration methods [167, 13, 136, 108].

Code and Data Availability

Our codebase is available at https://github.com/gaoliyao/uq_shred.

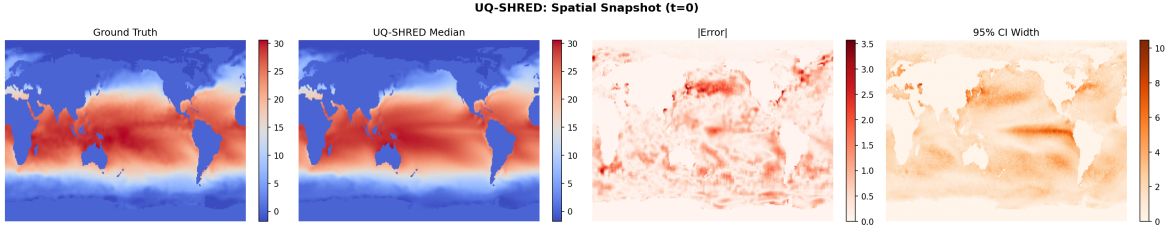


Figure 6.11: Spatial reconstruction of UQ-SHRED on SST data. Columns: ground truth, median reconstruction, absolute error, and 95% CI width.

6.6 Additional figs

Spatial reconstruction visualizations figs 6.11–6.15 present spatial reconstruction visualizations for the first four experimental datasets. Each figure displays, from left to right: the ground-truth field, the UQ-SHRED median reconstruction, the absolute error, and the 95% CI width, at selected time steps. In all cases, the spatial distribution of the 95% CI width closely tracks the spatial distribution of reconstruction error, indicating that UQ-SHRED assigns higher uncertainty where the reconstruction is less accurate.

1D RDE transient stage: additional figs Figure 6.16 presents temporal traces at selected spatial locations for both the closer and further RDE test trajectories. The confidence bands widen during detonation-wave passage and narrow during quiescent intervals, consistent with the spatial snapshots shown in Figure 6.7.

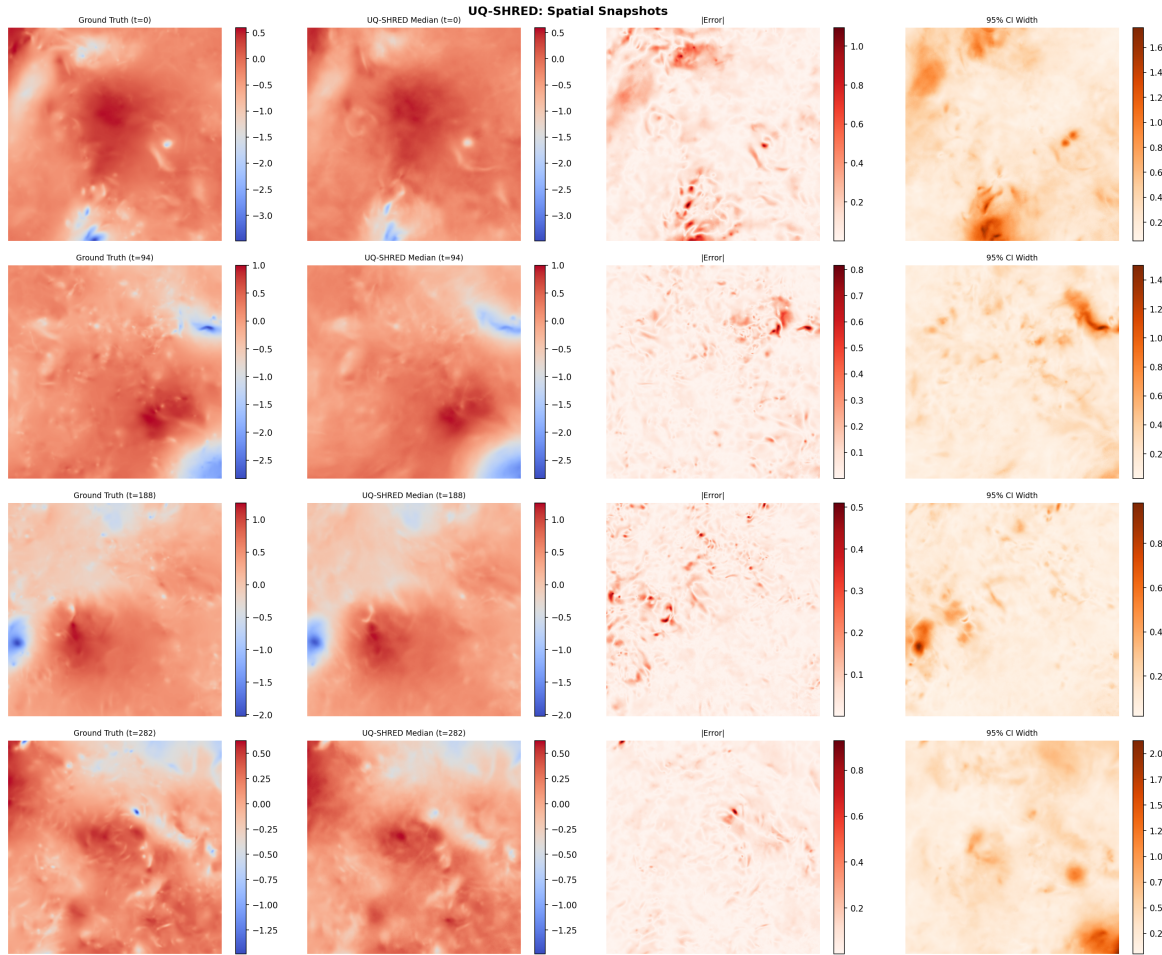


Figure 6.12: Spatial reconstruction of UQ-SHRED on isotropic turbulent flow data. Columns: ground truth, median reconstruction, absolute error, and 95% CI width.

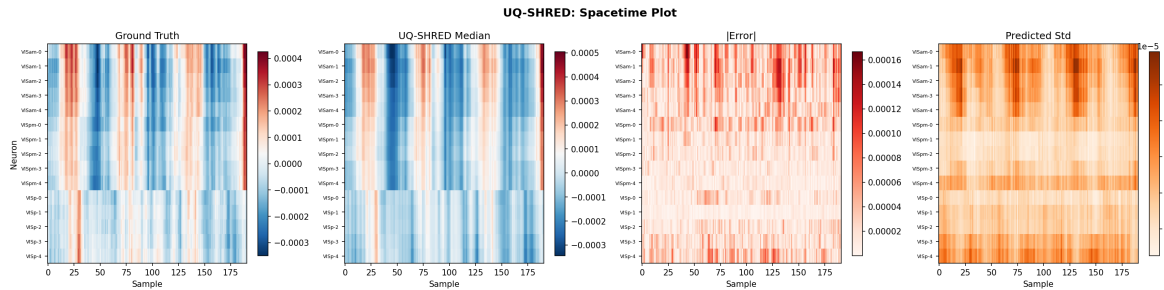


Figure 6.13: Spatial reconstruction of UQ-SHRED on neuro activity data. Columns: ground truth, median reconstruction, absolute error, and 95% CI width.

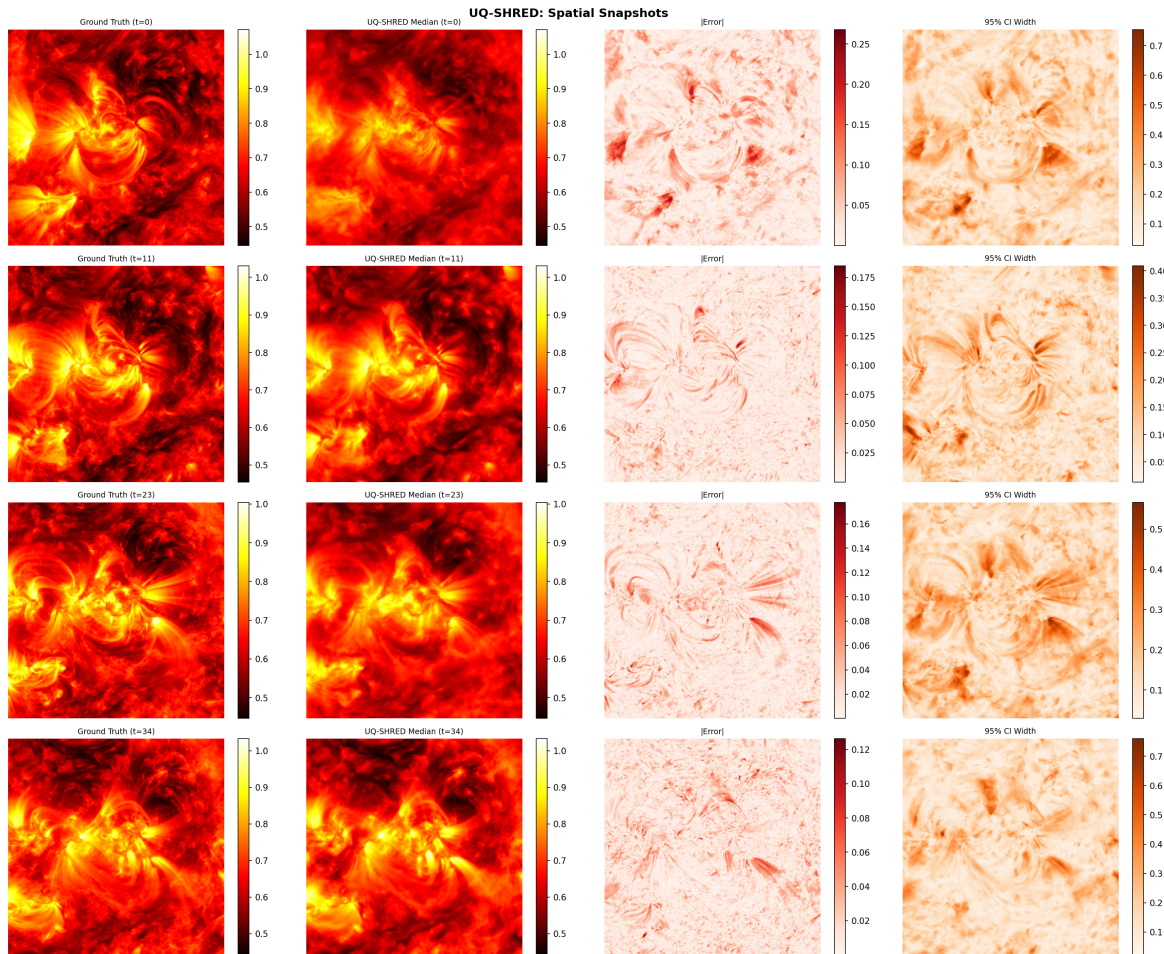


Figure 6.14: Spatial reconstruction of UQ-SHRED on solar activity data. Columns: ground truth, median reconstruction, absolute error, and 95% CI width.

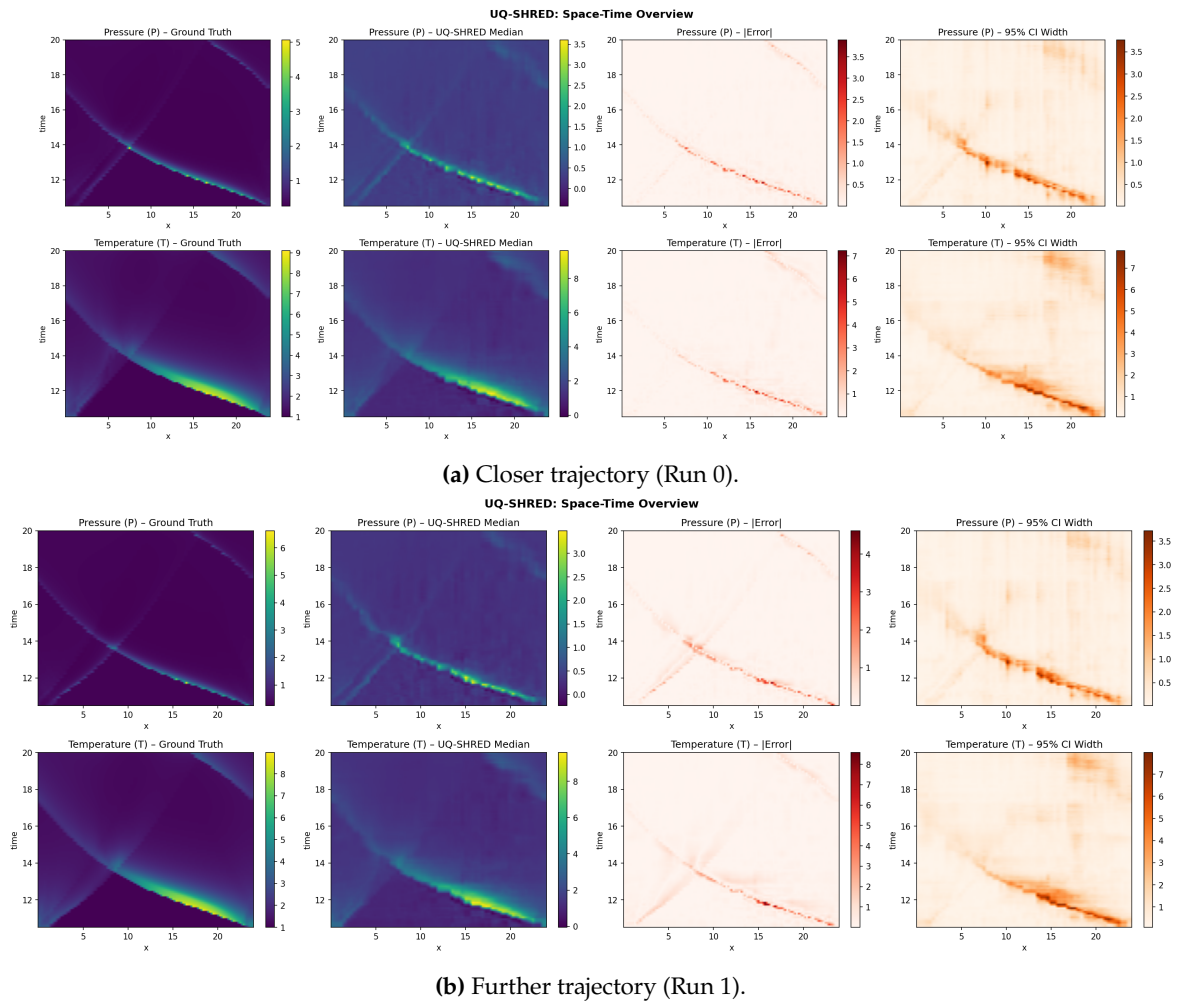


Figure 6.15: Spacetime overview for UQ-SHRED on the 1D RDE transient ignition stage. Columns: ground truth, median reconstruction, absolute error, and 95% CI width. Top rows: pressure; bottom rows: temperature. The 95% CI width concentrates along the detonation fronts, matching the spatial distribution of reconstruction error.

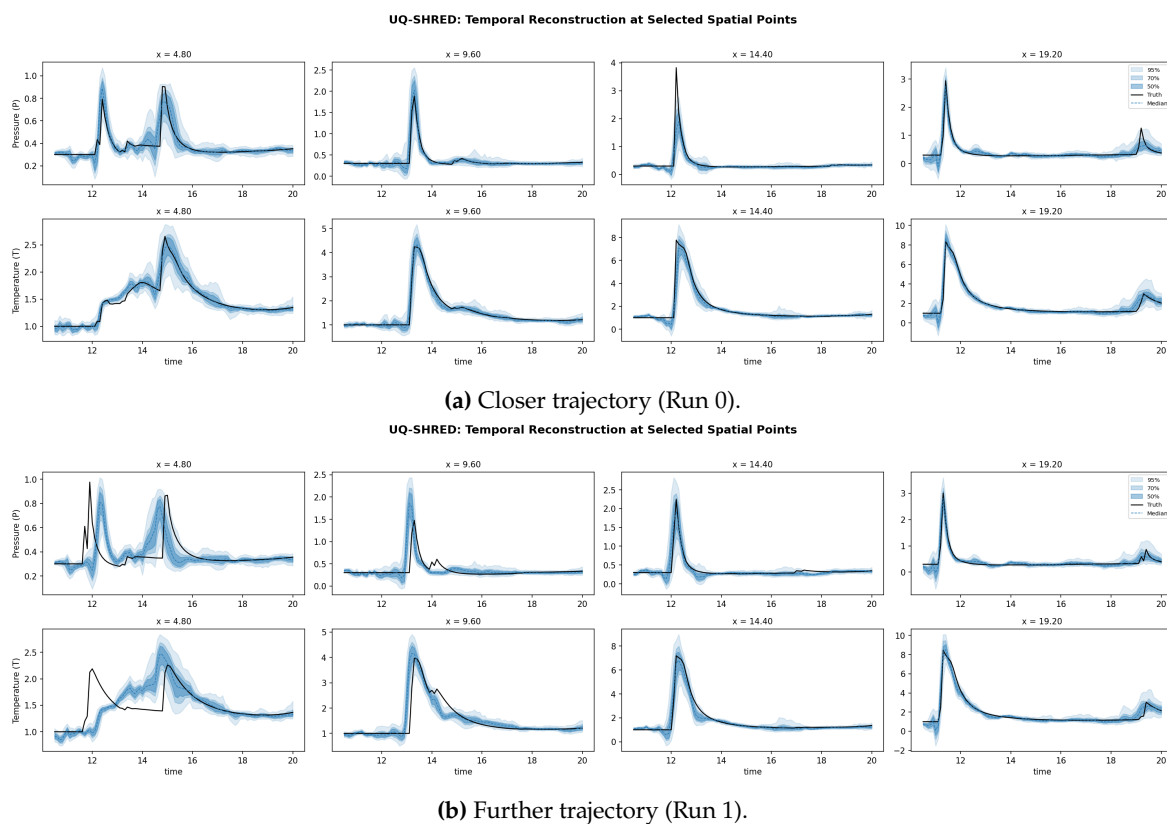


Figure 6.16: Temporal traces at selected spatial locations for UQ-SHRED on the 1D RDE transient ignition stage. Confidence bands widen during detonation-wave passage and contract during quiescent intervals.

7

Conclusion

This dissertation has investigated how machine learning can be used not only as a tool for prediction, but as a framework for discovering interpretable governing structure from scientific data. The motivating premise throughout this work is that accurate approximation, while essential, is not sufficient for scientific understanding. In scientific and engineering applications, the objective is often not merely to reproduce observed measurements, but to identify the variables, equations, uncertainties, and structural principles that govern the underlying system. This distinction is central: a predictive model may interpolate successfully within the distribution on which it was trained, yet provide limited insight into why a system behaves as it does or how it may behave under new conditions. Governing laws, by contrast, provide a compact and interpretable representation of mechanism.

The work presented in this dissertation develops this idea across several increasingly realistic data regimes. Beginning with the problem of latent physical law discovery, the thesis first considers settings in which the intrinsic coordinates of a system are hidden behind high-dimensional observations. Bayesian SINDy autoencoders were developed to jointly infer latent coordinates, sparse governing equations, and physical constants, while also providing uncertainty estimates for the discovered model. This work demonstrates that interpretable dynamics can be recovered even when the measured data are not themselves the natural variables of the system.

The dissertation then extends governing law discovery to high-dimensional spatiotemporal systems observed through sparse sensors. SINDy-SHRED and Koopman-SHRED show that sensor histories can encode sufficient information to reconstruct full fields and identify

parsimonious latent dynamics. This provides a bridge between sparse sensing, reduced-order modeling, and interpretable dynamical systems discovery. Rather than learning a black-box map from sensor measurements to future states, these methods impose a low-dimensional dynamical structure that can be inspected, simulated, and interpreted.

A central concern in any discovery problem is reliability. Sparse regression can produce compact models, but compactness alone does not guarantee correctness. Noise, limited data, numerical differentiation error, and library mismatch can all introduce spurious terms. For this reason, this dissertation also studies uncertainty quantification for sparse model discovery. Ensemble and Bayesian approaches provide a way to assess which terms and coefficients are stable under perturbations of the data and modeling procedure. This perspective is essential for scientific machine learning: a discovered equation should not only be sparse, but accompanied by evidence for which parts of the model are reliable.

The thesis further addresses the practical limitations of classical sparse identification methods by removing the assumption of structured-grid measurements. Mesh-free SINDy uses neural network surrogates and automatic differentiation to discover partial differential equations from scarce, noisy, and irregularly sampled data. This broadens the applicability of sparse PDE discovery to experimental settings where uniform grids, dense measurements, and accurate finite-difference derivatives are unavailable. In doing so, it moves sparse law discovery closer to the conditions under which real scientific data are often collected.

Finally, this dissertation considers uncertainty in sparse sensing and field reconstruction. UQ-SHRED extends deterministic sparse-sensor reconstruction to distributional prediction by learning conditional distributions over high-dimensional states from limited sensor histories. This is important because sparse measurements may not uniquely determine the full system state, particularly in stochastic, high-frequency, or partially observed regimes. In such cases, a single deterministic reconstruction can be misleading. A distributional model can instead represent ambiguity, express uncertainty, and indicate where the available measurements are insufficient.

Taken together, these contributions advance a unified view of scientific machine learning. The methods developed in this dissertation use machine learning to learn representations, but

constrain those representations with sparse dynamics, uncertainty estimates, and physical structure. Across latent coordinate discovery, sparse sensing, uncertainty quantification, mesh-free PDE discovery, and distributional reconstruction, the common goal is to convert data into interpretable models rather than only into accurate predictions.

Several limitations remain. The success of sparse discovery depends on the expressiveness and appropriateness of the candidate function library. If the true dynamics are not well represented by the chosen basis, the discovered equations may be only approximate, or may contain terms that reflect the limitations of the library rather than the true mechanism. In addition, latent coordinate systems are generally not unique. Different learned coordinates may reconstruct the same observations while producing different apparent equations, which complicates the interpretation of discovered latent dynamics. Uncertainty quantification also remains an open challenge, especially in finite-data and high-dimensional regimes where formal coverage guarantees are difficult to obtain. Finally, many real-world systems involve nonstationarity, multiscale coupling, stochastic forcing, control inputs, or changing regimes, while many current law-discovery frameworks still assume a relatively stable underlying dynamical structure.

These limitations suggest several directions for future work. One direction is to develop richer discovery libraries that include rational functions, nonlocal terms, stochastic dynamics, delay coordinates, multiscale closure models, and neural-symbolic representations. Another is to improve the identifiability of latent coordinates by incorporating symmetries, invariances, conservation laws, causal constraints, or weak forms of physical supervision. A further direction is to integrate law discovery with modern foundation models and large-scale scientific datasets, while preserving the interpretability and reliability required for scientific use. Finally, uncertainty-aware discovery should be developed not only as a diagnostic tool, but as a mechanism for guiding experiment design, active sensing, and model refinement.

The broader contribution of this dissertation is therefore conceptual as well as methodological. It argues that the role of machine learning in science should not be limited to replacing traditional models with black-box predictors. Instead, machine learning can serve as a computational instrument for discovering the coordinates, equations, structures, and

uncertainties that make complex systems understandable. Scientific progress depends not only on predicting what will happen, but on explaining why it happens and under what conditions those explanations remain valid. This dissertation contributes toward that goal by developing interpretable machine learning methods for governing law discovery from realistic, imperfect, and partially observed data.

Bibliography

- [1] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. 2021. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297.
- [2] Melissa Adrian, Jake A Soloff, and Rebecca Willett. 2024. Stabilizing black-box model selection with the inflated argmax. *arXiv preprint arXiv:2410.18268*.
- [3] Jacob Albright. 2017. [Flow visualization in a water channel](#). YouTube video.
- [4] E Paulo Alves and Frederico Fiuza. 2022. Data-driven discovery of reduced plasma physics models from fully kinetic simulations. *Physical Review Research*, 4(3):033192.
- [5] Arash Amini, Ulugbek S Kamilov, and Michael Unser. 2012. The analog formulation of sparsity implies infinite divisibility and rules out bernoulli-gaussian priors. In *2012 IEEE Information Theory Workshop*, pages 682–686. Ieee.
- [6] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. 2023. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15619–15629.
- [7] Francis R Bach. 2008. Bolasso: model consistent lasso estimation through the bootstrap. In *Proceedings of the 25th international conference on Machine learning*, pages 33–40.
- [8] Jincheng Bai, Qifan Song, and Guang Cheng. 2020. Efficient variational inference for sparse deep learning with theoretical guarantee. *Advances in Neural Information Processing Systems*, 33:466–476.
- [9] Joseph Bakarji, Kathleen Champion, J Nathan Kutz, and Steven L Brunton. 2022. Discovering governing equations from partial measurements with deep delay autoencoders. *arXiv preprint arXiv:2201.05136*.

- [10] Xuchan Bao, James Lucas, Sushant Sachdeva, and Roger B Grosse. 2020. Regularized linear autoencoders recover the principal components, eventually. *Advances in Neural Information Processing Systems*, 33:6971–6981.
- [11] Yuxuan Bao and J Nathan Kutz. 2025. Data assimilation and discrepancy modeling with shallow recurrent decoders. *arXiv preprint arXiv:2512.01170*.
- [12] Yuxuan Bao, Jan Zajac, Megan Powers, Venkat Raman, and J Nathan Kutz. 2026. Cheap2rich: A multi-fidelity framework for data assimilation and system identification of multiscale physics–rotating detonation engines. *arXiv preprint arXiv:2601.20295*.
- [13] Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. 2023. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845.
- [14] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. 2024. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*.
- [15] Peter L Bartlett and Shahar Mendelson. 2002. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482.
- [16] Peter L Bartlett, Andrea Montanari, and Alexander Rakhlin. 2021. Deep learning: a statistical viewpoint. *Acta numerica*, 30:87–201.
- [17] Harry Bateman. 1915. Some recent researches on the motion of fluids. *Monthly Weather Review*, 43(4):163–170.
- [18] Sarah Beetham and Jesse Capecelatro. 2020. Formulating turbulence closures using sparse regression with embedded form invariance. *Physical Review Fluids*, 5(8):084611.
- [19] Sarah Beetham, Rodney O Fox, and Jesse Capecelatro. 2021. Sparse identification of multiphase turbulence closures for coupled fluid–particle flows. *Journal of Fluid Mechanics*, 914.
- [20] Dimitris Bertsimas and Wes Gurnee. 2023. Learning sparse nonlinear dynamics via mixed-integer optimization. *Nonlinear Dynamics*, 111(7):6585–6604.

- [21] 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. 2001. Global modeling of tropospheric chemistry with assimilated meteorology: Model description and evaluation. *Journal of Geophysical Research: Atmospheres*, 106(D19):23073–23095.
- [22] Kush Bhatia, Prateek Jain, and Purushottam Kar. 2015. Robust regression via hard thresholding. *Advances in neural information processing systems*, 28.
- [23] Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. 2023. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619(7970):533–538.
- [24] Peter J Bickel, Ya’acov Ritov, and Alexandre B Tsybakov. 2009. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, pages 1705–1732.
- [25] Niloy Biswas, Lester Mackey, and Xiao-Li Meng. 2022. Scalable spike-and-slab. In *International Conference on Machine Learning*, pages 2021–2040. PMLR.
- [26] Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Gutttag. 2020. What is the state of neural network pruning? *Proceedings of machine learning and systems*, 2:129–146.
- [27] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR.
- [28] Lorenzo Boninsegna, Feliks Nüske, and Cecilia Clementi. 2018. Sparse learning of stochastic dynamical equations. *The Journal of chemical physics*, 148(24):241723.
- [29] Gert-Jan Both, Subham Choudhury, Pierre Sens, and Remy Kusters. 2021. [Deepmod: Deep learning for model discovery in noisy data](#). *Journal of Computational Physics*, 428:109985.
- [30] Leo Breiman. 1996. Bagging predictors. *Machine learning*, 24(2):123–140.

- [31] Steven L Brunton, Marko Budišić, Eurika Kaiser, and J Nathan Kutz. 2021. Modern koopman theory for dynamical systems. *arXiv preprint arXiv:2102.12086*.
- [32] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. 2016. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937.
- [33] Peter Bühlmann and Bin Yu. 2002. Analyzing bagging. *The annals of Statistics*, 30(4):927–961.
- [34] Peter Bühlmann, Bin Yu, Yoram Singer, and Larry Wasserman. 2006. Sparse boosting. *Journal of Machine Learning Research*, 7(6).
- [35] Johannes Martinus Burgers. 1948. A mathematical model illustrating the theory of turbulence. *Advances in applied mechanics*, 1:171–199.
- [36] Salva Rühling Cachay, Brian Henn, Oliver Watt-Meyer, Christopher S Bretherton, and Rose Yu. 2024. Probabilistic emulation of a global climate model with spherical dyffusion. *arXiv preprint arXiv:2406.14798*.
- [37] Jared L Callaham, J-C Loiseau, Georgios Rigas, and Steven L Brunton. 2021. Nonlinear stochastic modelling with langevin regression. *Proceedings of the Royal Society A*, 477(2250):20210092.
- [38] Jared L Callaham, Georgios Rigas, Jean-Christophe Loiseau, and Steven L Brunton. 2022. An empirical mean-field model of symmetry-breaking in a turbulent wake. *Science Advances*, 8(19):eabm4786.
- [39] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. 2011. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37.
- [40] Emmanuel J Candès, Justin Romberg, and Terence Tao. 2006. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509.

- [41] Emmanuel J Candes, Justin K Romberg, and Terence Tao. 2006. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(8):1207–1223.
- [42] Emmanuel J Candes and Terence Tao. 2005. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215.
- [43] Claudio Canuto, M Yousuff Hussaini, Alfio Quarteroni, and Thomas A Zang. 2007. *Spectral methods: evolution to complex geometries and applications to fluid dynamics*. Springer Science & Business Media.
- [44] Carlos M Carvalho, Nicholas G Polson, and James G Scott. 2009. Handling sparsity via the horseshoe. In *Artificial Intelligence and Statistics*, pages 73–80. PMLR.
- [45] Carlos M Carvalho, Nicholas G Polson, and James G Scott. 2010. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480.
- [46] Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. 2019. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451.
- [47] Kathleen P Champion, Steven L Brunton, and J Nathan Kutz. 2019. Discovery of nonlinear multiscale systems: Sampling strategies and embeddings. *SIAM Journal on Applied Dynamical Systems*, 18(1):312–333.
- [48] Gary J Chandler and Rich R Kerswell. 2013. Invariant recurrent solutions embedded in a turbulent two-dimensional kolmogorov flow. *Journal of Fluid Mechanics*, 722:554–595.
- [49] Rohitash Chandra, Mahir Jain, Manavendra Maharana, and Pavel N Krivitsky. 2022. Revisiting bayesian autoencoders with mcmc. *IEEE Access*, 10:40482–40495.
- [50] Arindam Chatterjee and S Lahiri. 2010. Asymptotic properties of the residual bootstrap for lasso estimators. *Proceedings of the American Mathematical Society*, 138(12):4497–4509.

- [51] Arindam Chatterjee and Soumendra Nath Lahiri. 2011. Bootstrapping lasso estimators. *Journal of the American Statistical Association*, 106(494):608–625.
- [52] Aoxue Chen and Guang Lin. 2021. Robust data-driven discovery of partial differential equations with time-dependent coefficients. *arXiv preprint arXiv:2102.01432*.
- [53] Boyuan Chen, Kuang Huang, Sunand Raghupathi, Ishaan Chandratreya, Qiang Du, and Hod Lipson. 2021. Discovering state variables hidden in experimental data. *arXiv preprint arXiv:2112.10755*.
- [54] Boyuan Chen, Kuang Huang, Sunand Raghupathi, Ishaan Chandratreya, Qiang Du, and Hod Lipson. 2022. Automated discovery of fundamental variables hidden in experimental data. *Nature Computational Science*, 2(7):433–442.
- [55] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. 2018. Neural ordinary differential equations. *Advances in neural information processing systems*, 31.
- [56] Tianqi Chen, Emily Fox, and Carlos Guestrin. 2014. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pages 1683–1691. PMLR.
- [57] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, et al. 2024. Symbolic discovery of optimization algorithms. *Advances in neural information processing systems*, 36.
- [58] Yuming Chen, Daniel Sanz-Alonso, and Rebecca Willett. 2023. Reduced-order autodifferentiable ensemble kalman filters. *Inverse Problems*, 39(12):124001.
- [59] Zhao Chen, Yang Liu, and Hao Sun. 2021. [Physics-informed learning of governing equations from scarce data](#). *Nature Communications*, 12(1).
- [60] Sheng Cheng, Deqian Kong, Jianwen Xie, Kookjin Lee, Ying Nian Wu, and Yezhou Yang. 2024. Latent space energy-based neural odes. *arXiv preprint arXiv:2409.03845*.
- [61] Victor Chernozhukov, Denis Chetverikov, and Kengo Kato. 2017. [Central limit theorems and bootstrap in high dimensions](#). *The Annals of Probability*, 45(4):2309 – 2352.

- [62] Alexandre J Chorin and Fei Lu. 2015. Discrete approach to stochastic parametrization and dimension reduction in nonlinear dynamics. *Proceedings of the National Academy of Sciences*, 112(32):9804–9809.
- [63] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 2016. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer.
- [64] Paolo Conti, Giorgio Gobat, Stefania Fresca, Andrea Manzoni, and Attilio Frangi. 2023. Reduced order modeling of parametrized systems through autoencoders and sindy approach: continuation of periodic solutions. *Computer Methods in Applied Mechanics and Engineering*, 411:116072.
- [65] Richard Courant et al. 1994. Variational methods for the solution of problems of equilibrium and vibrations. *Lecture notes in pure and applied mathematics*, pages 1–1.
- [66] Miles Cranmer. 2023. Interpretable machine learning for science with pysr and symbolicregression. jl. *arXiv preprint arXiv:2305.01582*.
- [67] Magnus Dam, Morten Brøns, Jens Juul Rasmussen, Volker Naulin, and Jan S Hesthaven. 2017. Sparse identification of a predator-prey system from simulation data of a convection model. *Physics of Plasmas*, 24(2):022310.
- [68] Emmanuel De Bézenac, Arthur Pajot, and Patrick Gallinari. 2019. Deep learning for physical processes: Incorporating prior scientific knowledge. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124009.
- [69] Nan Deng, Bernd R Noack, Marek Morzyński, and Luc R Pastur. 2021. Galerkin force model for transient and post-transient dynamics of the fluidic pinball. *Journal of Fluid Mechanics*, 918.
- [70] Wei Deng, Qi Feng, Liyao Gao, Faming Liang, and Guang Lin. 2020. Non-convex

- learning via replica exchange stochastic gradient mcmc. In *International Conference on Machine Learning*, pages 2474–2483. PMLR.
- [71] Wei Deng, Guang Lin, and Faming Liang. 2020. A contour stochastic gradient langevin dynamics algorithm for simulations of multi-modal distributions. *Advances in neural information processing systems*, 33:15725–15736.
- [72] Wei Deng, Guang Lin, and Faming Liang. 2022. An adaptively weighted stochastic gradient mcmc algorithm for monte carlo simulation and global optimization. *Statistics and Computing*, 32(4):1–24.
- [73] Wei Deng, Xiao Zhang, Faming Liang, and Guang Lin. 2019. An adaptive empirical bayesian method for sparse deep learning. *Advances in neural information processing systems*, 32.
- [74] Lijun Ding, Dmitriy Drusvyatskiy, Maryam Fazel, and Zaid Harchaoui. 2024. Flat minima generalize for low-rank matrix recovery. *Information and Inference: A Journal of the IMA*, 13(2):iaae009.
- [75] Nan Ding, Youhan Fang, Ryan Babbush, Changyou Chen, Robert D Skeel, and Hartmut Neven. 2014. Bayesian sampling using stochastic gradient thermostats. *Advances in neural information processing systems*, 27.
- [76] David L Donoho. 1995. De-noising by soft-thresholding. *IEEE transactions on information theory*, 41(3):613–627.
- [77] David L Donoho and Jain M Johnstone. 1994. Ideal spatial adaptation by wavelet shrinkage. *biometrika*, 81(3):425–455.
- [78] Megan R Ebers, Jan P Williams, Katherine M Steele, and J Nathan Kutz. 2024. Leveraging arbitrary mobile sensor trajectories with shallow recurrent decoder networks for full-state reconstruction. *IEEE Access*.
- [79] Bradley Efron. 1979. Computers and the theory of statistics: thinking the unthinkable. *SIAM review*, 21(4):460–480.

- [80] Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.
- [81] Gaute T Einevoll, Christoph Kayser, Nikos K Logothetis, and Stefano Panzeri. 2013. Modelling and analysis of local field potentials for studying the function of cortical circuits. *Nature Reviews Neuroscience*, 14(11):770–785.
- [82] Benet Eiximeno, Arnau Miró, Beka Begiashvili, Eusebio Valero, Ivette Rodriguez, and Oriol Lehmkhul. 2025. pylom: A hpc open source reduced order model suite for fluid dynamics applications. *Computer Physics Communications*, 308:109459.
- [83] D Elizondao, Emile Fiesler, and Jerzy Korczak. 1995. Non-ontogenic sparse neural networks. In *Proceedings of ICNN'95-International Conference on Neural Networks*, volume 1, pages 290–295. IEEE.
- [84] Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.
- [85] N Benjamin Erichson, Lionel Mathelin, Zhewei Yao, Steven L Brunton, Michael W Mahoney, and J Nathan Kutz. 2020. Shallow neural networks for fluid flow reconstruction with limited sensors. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476(2238).
- [86] N Benjamin Erichson, Vinicius Mikuni, Dongwei Lyu, Yang Gao, Omri Azencot, Soon Hoe Lim, and Michael W Mahoney. 2025. Flex: A backbone for diffusion-based modeling of spatio-temporal physical systems. *arXiv preprint arXiv:2505.17351*.
- [87] Jianqing Fan and Runze Li. 2001. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360.
- [88] Nicola Farenga, Stefania Fresca, Simone Brivio, and Andrea Manzoni. 2024. On latent dynamics learning in nonlinear reduced order modeling. *arXiv preprint arXiv:2408.15183*.
- [89] U. Fasel, J. N. Kutz, B. W. Brunton, and S. L. Brunton. 2022. [Ensemble-sindy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control](#). *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 478(2260).

- [90] Urban Fasel, J Nathan Kutz, Bingni W Brunton, and Steven L Brunton. 2022. 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.
- [91] Chelsea Finn, Ian Goodfellow, and Sergey Levine. 2016. Unsupervised learning for physical interaction through video prediction. *Advances in neural information processing systems*, 29.
- [92] Bernat Font, Gabriel D Weymouth, Vinh-Tan Nguyen, and Owen R Tutty. 2021. Deep learning of the spanwise-averaged navier–stokes equations. *Journal of Computational Physics*, 434:110199.
- [93] George E Forsythe, Michal A Malcolm, and Cleve B Moler. 1977. Computer methods for mathematical computation prentice-hall. *Englewood Cliffs, New Jersey*.
- [94] David A Freedman. 1981. Bootstrapping regression models. *The Annals of Statistics*, 9(6):1218–1228.
- [95] Wenjiang Fu and Keith Knight. 2000. Asymptotics for lasso-type estimators. *The Annals of statistics*, 28(5):1356–1378.
- [96] Kai Fukami, Takaaki Murata, Kai Zhang, and Koji Fukagata. 2021. Sparse identification of nonlinear dynamics with low-dimensionalized flow representations. *Journal of Fluid Mechanics*, 926:A10.
- [97] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.
- [98] L Gao, Urban Fasel, Steven L Brunton, and J Nathan Kutz. 2023. Convergence of uncertainty estimates in ensemble and bayesian sparse model discovery. *arXiv preprint arXiv:2301.12649*.
- [99] L Mars Gao and J Nathan Kutz. 2022. Bayesian autoencoders for data-driven discovery

- of coordinates, governing equations and fundamental constants. *arXiv e-prints*, pages arXiv–2211.
- [100] M. Gao and N. Kutz. 2024. [Bayesian autoencoders for data-driven discovery of coordinates, governing equations and fundamental constants](#). *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 480(2286).
- [101] Mars Liyao Gao, Yuxuan Bao, Amy S Rude, Xinwei Shen, and J Nathan Kutz. 2026. Uq-shred: uncertainty quantification of shallow recurrent decoder networks for sparse sensing via engression. *arXiv preprint arXiv:2604.01305*.
- [102] Mars Liyao Gao, J Nathan Kutz, and Bernat Font. 2025. Mesh-free sparse identification of nonlinear dynamics. *arXiv preprint arXiv:2505.16058*.
- [103] Mars Liyao Gao, Jan P Williams, and J Nathan Kutz. 2025. Sparse identification of nonlinear dynamics and Koopman operators with shallow recurrent decoder networks. *arXiv preprint arXiv:2501.13329*.
- [104] Mars Liyao Gao, Jan P Williams, and J Nathan Kutz. 2026. Sparse identification of nonlinear dynamics and koopman operators with shallow recurrent decoder networks. *Proceedings of the National Academy of Sciences*, 123(16):e2508144123.
- [105] Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z Li. 2022. Simvp: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3170–3180.
- [106] Patrick Gelß, Stefan Klus, Jens Eisert, and Christof Schütte. 2019. Multidimensional approximation of nonlinear dynamical systems. *Journal of Computational and Nonlinear Dynamics*, 14(6).
- [107] Edward I George and Robert E McCulloch. 1997. Approaches for bayesian variable selection. *Statistica sinica*, pages 339–373.
- [108] Isaac Gibbs and Emmanuel Candes. 2021. Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, 34:1660–1672.

- [109] Mihai Giurcanu. 2016. Thresholding least-squares inference in high-dimensional regression models. *Electronic Journal of Statistics*, 10(2):2124–2156.
- [110] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings.
- [111] Tilmann Gneiting and Adrian E Raftery. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378.
- [112] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. 2018. Size-independent sample complexity of neural networks. In *Conference On Learning Theory*, pages 297–299. PMLR.
- [113] Yifei Guan, Steven L Brunton, and Igor Novosselov. 2021. Sparse nonlinear models of chaotic electroconvection. *Royal Society Open Science*, 8(8):202367.
- [114] John Guckenheimer and Philip Holmes. 2013. *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, volume 42. Springer Science & Business Media.
- [115] Vincent Le Guen and Nicolas Thome. 2020. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11474–11484.
- [116] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.
- [117] Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schon. 2020. Evaluating scalable bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 318–319.
- [118] Peter Hall. 2013. *The bootstrap and Edgeworth expansion*. Springer Science & Business Media.

- [119] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009.
- [120] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- [121] Wenchong He, Zhe Jiang, Tingsong Xiao, Zelin Xu, and Yukun Li. 2026. A survey on uncertainty quantification methods for deep learning. *ACM Computing Surveys*, 58(7):1–35.
- [122] José Miguel Hernández-Lobato and Ryan Adams. 2015. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International conference on machine learning*, pages 1861–1869. PMLR.
- [123] Charles Gordon Hewitt. 1921. *The conservation of the wild life of Canada*. New York: C. Scribner.
- [124] Nicholas J Higham. 2002. *Accuracy and stability of numerical algorithms*. SIAM.
- [125] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507.
- [126] Seth M Hirsh, David A Barajas-Solano, and J Nathan Kutz. 2022. Sparsifying priors for bayesian uncertainty quantification in model discovery. *Royal Society Open Science*, 9(2):211823.
- [127] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.
- [128] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

- [129] Matthew Hoffman and Yi-An Ma. 2020. Black-box variational inference as distilled langevin dynamics. In *Proceedings of the 37th International Conference on Machine Learning*, pages 4324–4341.
- [130] Matthew D Hoffman, Andrew Gelman, et al. 2014. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623.
- [131] Moritz Hoffmann, Christoph Fröhner, and Frank Noé. 2019. Reactive sindy: Discovering governing reactions from concentration data. *The Journal of chemical physics*, 150(2):025101.
- [132] Philipp Holl, Vladlen Koltun, and Nils Thuerey. 2020. Learning to control pdes with differentiable physics. *arXiv preprint arXiv:2001.07457*.
- [133] Benjamin Holzsuh, Qiang Liu, Georg Kohl, and Nils Thuerey. 2025. Pde-transformer: Efficient and versatile transformers for physics simulations. *arXiv preprint arXiv:2505.24717*.
- [134] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.
- [135] Jian Huang, Joel L Horowitz, and Shuangge Ma. 2008. Asymptotic properties of bridge estimators in sparse high-dimensional regression models. *The Annals of Statistics*, 36(2):587–613.
- [136] Kexin Huang, Ying Jin, Emmanuel Candes, and Jure Leskovec. 2023. Uncertainty quantification over graph with conformalized graph neural networks. *Advances in Neural Information Processing Systems*, 36:26699–26721.
- [137] Hemant Ishwaran and J Sunil Rao. 2005. Spike and slab variable selection: frequentist and bayesian strategies. *The Annals of Statistics*, 33(2):730–773.
- [138] Vikram Iyer, Rajalakshmi Nandakumar, Anran Wang, Sawyer B Fuller, and Shyamnath Gollakota. 2019. Living iot: A flying wireless platform on live insects. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–15.

- [139] Kadierdan Kaheman, J Nathan Kutz, and Steven L Brunton. 2020. Sindy-pi: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proceedings of the Royal Society A*, 476(2242):20200279.
- [140] Eurika Kaiser, J Nathan Kutz, and Steven L Brunton. 2018. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society A*, 474(2219):20180335.
- [141] Yoshinao Kajikawa and Charles E Schroeder. 2011. How local is the local field potential? *Neuron*, 72(5):847–858.
- [142] Rama Kandukuri, Jan Achterhold, Michael Moeller, and Joerg Stueckler. 2020. Learning to identify physical parameters from video using differentiable physics. In *DAGM German conference on pattern recognition*, pages 44–57. Springer.
- [143] Alan A Kaptanoglu, Jared L Callaham, Aleksandr Aravkin, Christopher J Hansen, and Steven L Brunton. 2021. Promoting global stability in data-driven models of quadratic nonlinear dynamics. *Physical Review Fluids*, 6(9):094401.
- [144] Alan A Kaptanoglu, Kyle D Morgan, Chris J Hansen, and Steven L Brunton. 2021. Physics-constrained, low-dimensional models for magnetohydrodynamics: First-principles and data-driven approaches. *Physical Review E*, 104(1):015206.
- [145] Alan A Kaptanoglu, Brian M de Silva, Urban Fasel, Kadierdan Kaheman, Jared L Callaham, Charles B Delahunt, Kathleen Champion, Jean-Christophe Loiseau, J Nathan Kutz, and Steven L Brunton. 2021. Pysindy: A comprehensive python package for robust sparse system identification. *arXiv preprint arXiv:2111.08481*.
- [146] Jerzy Karczmarczuk. 2001. [Functional differentiation of computer programs](#). *Higher-Order and Symbolic Computation*, 14(1):35–57.
- [147] Nouredine El Karoui and Elizabeth Purdom. 2016. Can we trust the bootstrap in high-dimension? *arXiv preprint arXiv:1608.00696*.

- [148] Jonathan A Kelner, Frederic Koehler, Raghu Meka, and Dhruv Rohatgi. 2022. On the power of preconditioning in sparse linear regression. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 550–561. IEEE.
- [149] Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30.
- [150] Yuehaw Khoo, Jianfeng Lu, and Lexing Ying. 2021. Solving parametric pde problems with artificial neural networks. *European Journal of Applied Mathematics*, 32(3):421–435.
- [151] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- [152] Andrei A Klishin, Joseph Bakarji, J Nathan Kutz, and Krithika Manohar. 2024. Statistical mechanics of dynamical system identification. *arXiv preprint arXiv:2403.01723*.
- [153] James Koch and J Nathan Kutz. 2020. Modeling thermodynamic trends of rotating detonation engines. *Physics of Fluids*, 32(12).
- [154] Yunbum Kook, Yin Tat Lee, Ruoqi Shen, and Santosh Vempala. 2023. Condition-number-independent convergence rate of riemannian hamiltonian monte carlo with numerical integrators. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 4504–4569. PMLR.
- [155] Bernard O Koopman. 1931. Hamiltonian systems and transformation in hilbert space. *Proceedings of the national academy of sciences of the united states of america*, 17(5):315.
- [156] Bernard O Koopman and J v Neumann. 1932. Dynamical systems of continuous spectra. *Proceedings of the National Academy of Sciences*, 18(3):255–263.
- [157] J Nathan Kutz. 2023. Machine learning for parameter estimation. *Proceedings of the National Academy of Sciences*, 120(12):e2300990120.
- [158] J Nathan Kutz, Peter Battaglia, Michael Brenner, Kevin Carlberg, Aric Hagberg, Shirley Ho, Stephan Hoyer, Henning Lange, Hod Lipson, Michael W Mahoney, et al. 2025.

- Accelerating scientific discovery with the common task framework. *arXiv preprint arXiv:2511.04001*.
- [159] J Nathan Kutz and Steven L Brunton. 2022. Parsimony as the ultimate regularizer for physics-informed machine learning. *Nonlinear Dynamics*, 107(3):1801–1817.
- [160] J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. 2016. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM.
- [161] J Nathan Kutz, Maryam Reza, Farbod Faraji, and Aaron Knoll. 2024. Shallow recurrent decoder for reduced order modeling of plasma dynamics. *arXiv preprint arXiv:2405.11955*.
- [162] Zhilu Lai and Satish Nagarajaiah. 2019. Sparse structural system identification method for nonlinear dynamic systems with hysteresis/inelastic behavior. *Mechanical Systems and Signal Processing*, 117:813–842.
- [163] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.
- [164] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. 2023. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421.
- [165] Michel Ledoux and Michel Talagrand. 2013. *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media.
- [166] Jason D Lee, Dennis L Sun, Yuekai Sun, and Jonathan E Taylor. 2016. Exact post-selection inference, with application to the lasso.
- [167] Lihua Lei and Emmanuel J Candès. 2021. Conformal inference of counterfactuals and individual treatment effects. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 83(5):911–938.

- [168] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2018. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31.
- [169] Liyuan Li, Weimin Huang, Irene Yu-Hua Gu, and Qi Tian. 2004. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing*, 13(11):1459–1472.
- [170] Yi Li, Eric Perlman, Minping Wan, Yunke Yang, Charles Meneveau, Randal Burns, Shiyi Chen, Alexander Szalay, and Gregory Eyink. 2008. A public turbulence database cluster and applications to study lagrangian evolution of velocity increments in turbulence. *Journal of Turbulence*, (9):N31.
- [171] Yunzhu Li, Toru Lin, Kexin Yi, Daniel Bear, Daniel Yamins, Jiajun Wu, Joshua Tenenbaum, and Antonio Torralba. 2020. Visual grounding of learned physical models. In *International conference on machine learning*, pages 5927–5936. PMLR.
- [172] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhat-tacharya, Andrew Stuart, and Anima Anandkumar. 2020. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*.
- [173] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhat-tacharya, Andrew Stuart, and Anima Anandkumar. 2020. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*.
- [174] Guang Lin, Christian Moya, and Zecheng Zhang. 2021. Accelerated replica exchange stochastic gradient langevin diffusion enhanced bayesian deepnet for solving noisy parametric pdes. *arXiv preprint arXiv:2111.02484*.
- [175] Kevin K Lin and Fei Lu. 2021. Data-driven model reduction, wiener projections, and the koopman-mori-zwanzig formalism. *Journal of Computational Physics*, 424:109864.
- [176] Yen Ting Lin, Yifeng Tian, Daniel Livescu, and Marian Anghel. 2021. Data-driven learning for the mori–zwanzig formalism: A generalization of the koopman learning framework. *SIAM Journal on Applied Dynamical Systems*, 20(4):2558–2601.

- [177] Baoyuan Liu, Min Wang, Hassan Foroosh, Marshall Tappen, and Marianna Pensky. 2015. Sparse convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 806–814.
- [178] Shaowei Liu, Zhongzheng Ren, Saurabh Gupta, and Shenlong Wang. 2024. Physgen: Rigid-body physics-grounded image-to-video generation. In *European Conference on Computer Vision*, pages 360–378. Springer.
- [179] Jean-Christophe Loiseau. 2020. Data-driven modeling of the chaotic thermal convection in an annular thermosyphon. *Theoretical and Computational Fluid Dynamics*, 34(4):339–365.
- [180] Jean-Christophe Loiseau and Steven L Brunton. 2018. Constrained sparse galerkin regression. *Journal of Fluid Mechanics*, 838:42–67.
- [181] Jean-Christophe Loiseau, Bernd R Noack, and Steven L Brunton. 2018. Sparse reduced-order modelling: sensor-based dynamics to full-state estimation. *Journal of Fluid Mechanics*, 844:459–490.
- [182] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. 2018. Pde-net: Learning pdes from data. In *International conference on machine learning*, pages 3208–3216. PMLR.
- [183] Alessandro Longhi, Danny Lathouwers, and Zoltán Perkó. 2026. Latent space modeling of parametric and time-dependent pdes using neural odes. *Computer Methods in Applied Mechanics and Engineering*, 448:118394.
- [184] Christos Louizos, Max Welling, and Diederik P Kingma. 2017. Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312*.
- [185] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. 2021. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229.
- [186] Bethany Lusch, J Nathan Kutz, and Steven L Brunton. 2018. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):1–10.

- [187] Yi-An Ma, Tianqi Chen, and Emily Fox. 2015. A complete recipe for stochastic gradient mcmc. *Advances in neural information processing systems*, 28.
- [188] Niall M Mangan, Travis Askham, Steven L Brunton, J Nathan Kutz, and Joshua L Proctor. 2019. Model selection for hybrid dynamical systems via sparse regression. *Proceedings of the Royal Society A*, 475(2223):20180534.
- [189] Niall M Mangan, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. 2016. Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 2(1):52–63.
- [190] Krithika Manohar, Bingni W Brunton, J Nathan Kutz, and Steven L Brunton. 2018. Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns. *IEEE Control Systems Magazine*, 38(3):63–86.
- [191] L Mars Gao and J Nathan Kutz. 2024. Bayesian autoencoders for data-driven discovery of coordinates, governing equations and fundamental constants. *Proceedings of the Royal Society A*, 480(2286):20230506.
- [192] Nick McGreivy and Ammar Hakim. 2024. Weak baselines and reporting biases lead to overoptimism in machine learning for fluid-related partial differential equations. *Nature Machine Intelligence*, 6(10):1256–1269.
- [193] Nicolai Meinshausen and Peter Bühlmann. 2010. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473.
- [194] Daniel A Messenger and David M Bortz. 2021. Weak sindy for partial differential equations. *Journal of Computational Physics*, 443:110525.
- [195] Mehdi Mirza. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.
- [196] Toby J Mitchell and John J Beauchamp. 1988. Bayesian variable selection in linear regression. *Journal of the american statistical association*, 83(404):1023–1032.

- [197] Arvind Mohan, Don Daniel, Michael Chertkov, and Daniel Livescu. 2019. Compressed convolutional lstm: An efficient deep learning framework to model high fidelity 3d turbulence. *arXiv preprint arXiv:1903.00033*.
- [198] Thibault Monsel. 2024. *Deep Learning for Partially Observed Dynamical Systems*. Ph.D. thesis, Université Paris-Saclay.
- [199] Peter H Morgan. 2008. Differential evolution and sparse neural networks. *Expert Systems*, 25(4):394–413.
- [200] Damian Mrowca, Chengxu Zhuang, Elias Wang, Nick Haber, Li F Fei-Fei, Josh Tenenbaum, and Daniel L Yamins. 2018. Flexible neural representation for physics prediction. *Advances in neural information processing systems*, 31.
- [201] Ashish S. Nair, Shivam Barwey, Pinaki Pal, Jonathan F. MacArt, Troy Arcomano, and Romit Maulik. 2025. [Understanding latent timescales in neural ordinary differential equation models for advection-dominated dynamical systems](#). *arXiv preprint arXiv:2403.02224*.
- [202] Radford M Neal. 2012. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media.
- [203] Radford M Neal et al. 2011. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2.
- [204] Michael A Newton and Adrian E Raftery. 1994. Approximate bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(1):3–26.
- [205] Tung Nguyen, Arsh Koneru, Shufan Li, and Aditya Grover. 2025. Physix: A foundation model for physics simulations. *arXiv preprint arXiv:2506.17774*.
- [206] Yiyu Ni, Marine A Denolle, Qibin Shi, Bradley P Lipovsky, Shaowu Pan, and J Nathan Kutz. 2024. Wavefield reconstruction of distributed acoustic sensing: Lossy compression,

- wavefield separation, and edge computing. *Journal of Geophysical Research: Machine Learning and Computation*, 1(3):e2024JH000247.
- [207] Lizhen Nie and Veronika Ročková. 2022. Bayesian bootstrap spike-and-slab lasso. *Journal of the American Statistical Association*, pages 1–16.
- [208] Lizhen Nie and Veronika Rockova. 2022. Deep bootstrap for bayesian inference. *arXiv preprint arXiv:2205.15374*.
- [209] Samuel E Otto and Clarence W Rowley. 2019. Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593.
- [210] Samuel E Otto, Nicholas Zolman, J Nathan Kutz, and Steven L Brunton. 2023. A unified framework to enforce, discover, and promote symmetry in machine learning. *arXiv preprint arXiv:2311.00212*.
- [211] Elise Özalp and Luca Magri. 2024. Stability analysis of chaotic systems in latent spaces. *arXiv preprint arXiv:2410.00480*.
- [212] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. 2021. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64.
- [213] Trevor Park and George Casella. 2008. The bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686.
- [214] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. 2022. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*.
- [215] Reese Pathak and Cong Ma. 2024. On the design-dependent suboptimality of the lasso. *arXiv preprint arXiv:2402.00382*.

- [216] Barak A. Pearlmutter and Jeffrey Mark Siskind. 2007. [Lazy multivariate higher-order forward-mode ad](#). In *Proceedings of the 34th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, POPL07, page 155–160. ACM.
- [217] W Dean Pesnell and PC Chamberlin. 2012. The solar dynamics observatory (sdo). In *The solar dynamics observatory*, pages 3–15. Springer.
- [218] Nicholas G Polson and Veronika Ročková. 2018. Posterior concentration for sparse deep learning. *Advances in Neural Information Processing Systems*, 31.
- [219] Stephen B Pope. 2001. Turbulent flows. *Measurement Science and Technology*, 12(11):2020–2021.
- [220] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. 2019. On the spectral bias of neural networks. In *International conference on machine learning*, pages 5301–5310. PMLR.
- [221] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. 2019. 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.
- [222] Maria Refinetti and Sebastian Goldt. 2022. The dynamics of representation learning in shallow, non-linear autoencoders. In *International Conference on Machine Learning*, pages 18499–18519. PMLR.
- [223] Patrick AK Reinbold, Daniel R Gurevich, and Roman O Grigoriev. 2020. Using noisy or incomplete data to discover models of spatiotemporal dynamics. *Physical Review E*, 101(1):010203.
- [224] Patrick AK Reinbold, Logan M Kageorge, Michael F Schatz, and Roman O Grigoriev. 2021. Robust learning from noisy, incomplete, high-dimensional experimental data via physically constrained symbolic regression. *Nature communications*, 12(1):1–8.

- [225] Richard W Reynolds, Nick A Rayner, Thomas M Smith, Diane C Stokes, and Wanqiu Wang. 2002. An improved in situ and satellite sst analysis for climate. *Journal of climate*, 15(13):1609–1625.
- [226] Stefano Riva, Carolina Introini, Antonio Cammi, and J Nathan Kutz. 2024. Robust state estimation from partial out-core measurements with shallow recurrent decoder for nuclear reactors. *arXiv preprint arXiv:2409.12550*.
- [227] Veronika Ročková and Edward I George. 2014. Emvs: The em approach to bayesian variable selection. *Journal of the American Statistical Association*, 109(506):828–846.
- [228] Veronika Ročková and Edward I George. 2018. The spike-and-slab lasso. *Journal of the American Statistical Association*, 113(521):431–444.
- [229] François Rozet, Ruben Ohana, Michael McCabe, Gilles Louppe, François Lanusse, and Shirley Ho. 2025. Lost in latent space: An empirical study of latent diffusion models for physics emulation. *arXiv preprint arXiv:2507.02608*.
- [230] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. 2019. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32.
- [231] Donald B Rubin. 1981. The bayesian bootstrap. *The annals of statistics*, pages 130–134.
- [232] Amy S Rude and J Nathan Kutz. 2025. Shallow recurrent decoders for neural and behavioral dynamics. *bioRxiv*, pages 2025–06.
- [233] Samuel Rudy, Alessandro Alla, Steven L Brunton, and J Nathan Kutz. 2019. Data-driven identification of parametric partial differential equations. *SIAM Journal on Applied Dynamical Systems*, 18(2):643–660.
- [234] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. 2017. Data-driven discovery of partial differential equations. *Science advances*, 3(4):e1602614.

- [235] Salva Rühling Cachay, Bo Zhao, Hailey Joren, and Rose Yu. 2023. Dyffusion: A dynamics-informed diffusion model for spatiotemporal forecasting. *Advances in neural information processing systems*, 36:45259–45287.
- [236] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. 1985. Learning internal representations by error propagation.
- [237] Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. 2018. Graph networks as learnable physics engines for inference and control. In *International conference on machine learning*, pages 4470–4479. PMLR.
- [238] Hayden Schaeffer. 2017. Learning partial differential equations via data discovery and sparse optimization. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2197):20160446.
- [239] Hayden Schaeffer and Scott G McCalla. 2017. Sparse model selection via integral terms. *Physical Review E*, 96(2):023302.
- [240] Martin Schmelzer, Richard P Dwight, and Paola Cinnella. 2020. Discovery of algebraic reynolds-stress models using sparse symbolic regression. *Flow, Turbulence and Combustion*, 104(2):579–603.
- [241] Michael Schmidt and Hod Lipson. 2009. Distilling free-form natural laws from experimental data. *science*, 324(5923):81–85.
- [242] Steven L Scott and Hal R Varian. 2014. Predicting the present with bayesian structural time series. *International Journal of Mathematical Modelling and Numerical Optimisation*, 5(1-2):4–23.
- [243] Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31.
- [244] Jun Shao. 1993. Linear model selection by cross-validation. *Journal of the American statistical Association*, 88(422):486–494.

- [245] Xinwei Shen and Nicolai Meinshausen. 2023. Engression: Extrapolation for nonlinear regression? *arXiv preprint arXiv:2307.00835*.
- [246] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28.
- [247] Joshua H Siegle, Xiaoxuan Jia, Séverine Durand, Sam Gale, Corbett Bennett, Nile Graddis, Gregory Heller, Tamina K Ramirez, Hannah Choi, Jennifer A Luviano, et al. 2021. Survey of spiking in the mouse visual system reveals functional hierarchy. *Nature*, 592(7852):86–92.
- [248] Michael Small, Kevin Judd, and Alistair Mees. 2002. Modeling continuous processes from data. *Physical Review E*, 65(4):046704.
- [249] Hing Cheung So. 2011. Source localization: Algorithms and analysis. *Handbook of position location: theory, practice, and advances*, pages 25–66.
- [250] Alberto Solera-Rico, Carlos Sanmiguel Vila, Miguel Gómez-López, Yuning Wang, Abdulrahman Almashjary, Scott TM Dawson, and Ricardo Vinuesa. 2024. β -variational autoencoders and transformers for reduced-order modelling of fluid flows. *Nature Communications*, 15(1):1361.
- [251] Jake A Soloff, Rina Foygel Barber, and Rebecca Willett. 2023. Bagging provides assumption-free stability. *arXiv preprint arXiv:2301.12600*.
- [252] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. 2021. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34:1415–1428.
- [253] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

- [254] Mariia Sorokina, Stylianos Sygletos, and Sergei Turitsyn. 2016. Sparse identification for nonlinear optical communication systems: Sino method. *Optics express*, 24(26):30433–30443.
- [255] Suraj Srinivas, Akshayvarun Subramanya, and R Venkatesh Babu. 2017. Training sparse neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 138–145.
- [256] Nils Chr Stenseth, Wilhelm Falck, Ottar N Bjørnstad, and Charles J Krebs. 1997. Population regulation in snowshoe hare and canadian lynx: asymmetric food web configurations between hare and lynx. *Proceedings of the National Academy of Sciences*, 94(10):5147–5152.
- [257] Robert Stephany and Christopher Earls. 2024. [PDE-LEARN: Using deep learning to discover partial differential equations from noisy, limited data](#). *Neural Networks*, 174:106242.
- [258] Weijie Su, Małgorzata Bogdan, and Emmanuel Candes. 2017. False discoveries occur early on the lasso path. *The Annals of statistics*, pages 2133–2150.
- [259] Yan Sun, Qifan Song, and Faming Liang. 2021. Consistent sparse deep learning: Theory and computation. *Journal of the American Statistical Association*, pages 1–15.
- [260] Yan Sun, Qifan Song, and Faming Liang. 2022. Learning sparse deep neural networks with a spike-and-slab prior. *Statistics & Probability Letters*, 180:109246.
- [261] Yan Sun, Wenjun Xiong, and Faming Liang. 2021. Sparse deep learning: A new framework immune to local traps and miscalibration. *Advances in Neural Information Processing Systems*, 34:22301–22312.
- [262] Gábor J Székely. 2003. E-statistics: The energy of statistical samples. *Bowling Green State University, Department of Mathematics and Statistics Technical Report*, 3(05):1–18.
- [263] Gábor J Székely and Maria L Rizzo. 2023. *The energy of data and distance correlation*. Chapman and Hall/CRC.

- [264] Floris Takens. 2006. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980: Proceedings of a Symposium Held at the University of Warwick 1979/80*, pages 366–381. Springer.
- [265] Yee Whye Teh, Alexandre H Thiery, and Sebastian J Vollmer. 2016. Consistency and fluctuations for stochastic gradient langevin dynamics. *Journal of Machine Learning Research*, 17.
- [266] Ye Tian, Yuqi Gu, and Yang Feng. 2023. Learning from similar linear representations: Adaptivity, minimaxity, and robustness. *arXiv preprint arXiv:2303.17765*.
- [267] Yifeng Tian, Yen Ting Lin, Marian Anghel, and Daniel Livescu. 2021. Data-driven learning of mori–zwanzig operators for isotropic turbulence. *Physics of Fluids*, 33(12).
- [268] Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.
- [269] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE.
- [270] Matteo Tomasetto, Jan P Williams, Francesco Braghin, Andrea Manzoni, and J Nathan Kutz. 2025. Reduced order modeling with shallow recurrent decoder networks. *arXiv preprint arXiv:2502.10930*.
- [271] Sana Tonekaboni, Tina Behrouzi, Addison Weatherhead, Emily Fox, David Blei, and Anna Goldenberg. 2025. Hdp-flow: Generalizable bayesian nonparametric model for time series state discovery. In *The 41st Conference on Uncertainty in Artificial Intelligence*.
- [272] Brian L Trippe and Richard E Turner. 2018. Conditional density estimation with bayesian normalising flows. *arXiv preprint arXiv:1802.04908*.
- [273] Aad W Van der Vaart. 2000. *Asymptotic statistics*, volume 3. Cambridge university press.
- [274] Aad W Van Der Vaart, Jon A Wellner, Aad W van der Vaart, and Jon A Wellner. 1996. *Weak convergence*. Springer.

- [275] Leonid Nisonovich Vaserstein. 1969. Markov processes over denumerable products of spaces, describing large systems of automata. *Problemy Peredachi Informatsii*, 5(3):64–72.
- [276] Aladin Virmaux and Kevin Scaman. 2018. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in Neural Information Processing Systems*, 31.
- [277] Martin J Wainwright. 2009. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1 -constrained quadratic programming (lasso). *IEEE transactions on information theory*, 55(5):2183–2202.
- [278] Martin J Wainwright. 2019. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press.
- [279] Jon Wakefield. 2013. *Bayesian and frequentist regression methods*, volume 23. Springer.
- [280] Bin Wang, Jie Lu, Zheng Yan, Huaishao Luo, Tianrui Li, Yu Zheng, and Guangquan Zhang. 2019. Deep uncertainty quantification: A machine learning approach for weather forecasting. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2087–2095.
- [281] Hao Wang and Dit-Yan Yeung. 2020. A survey on bayesian deep learning. *ACM Computing Surveys (CSUR)*, 53(5):1–37.
- [282] Jin Wang, Feng Liang, and Yuan Ji. 2016. An ensemble em algorithm for bayesian variable selection. *arXiv preprint arXiv:1603.04360*.
- [283] Wen-Xu Wang, Rui Yang, Ying-Cheng Lai, Vassilios Kovanis, and Celso Grebogi. 2011. Predicting catastrophes in nonlinear dynamical systems by compressive sensing. *Physical review letters*, 106(15):154101.
- [284] Xingyu Wang, Sewoong Oh, and Chang-Han Rhee. 2021. Eliminating sharp minima from sgd with truncated heavy-tailed noise. *arXiv preprint arXiv:2102.04297*.
- [285] Yating Wang, Wei Deng, and Guang Lin. 2021. Bayesian sparse learning with preconditioned stochastic gradient mcmc and its applications. *Journal of Computational Physics*, 432:110134.

- [286] Yuexi Wang and Veronika Rocková. 2020. Uncertainty quantification for sparse deep learning. In *International Conference on Artificial Intelligence and Statistics*, pages 298–308. PMLR.
- [287] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. 2017. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. *Advances in neural information processing systems*, 30.
- [288] Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. 2015. Embed to control: A locally linear latent dynamics model for control from raw images. *Advances in neural information processing systems*, 28.
- [289] Max Welling and Yee W Teh. 2011. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688.
- [290] Tino Werner. 2022. Loss-guided stability selection. *arXiv preprint arXiv:2202.04956*.
- [291] Samuel S Wilks. 1938. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The annals of mathematical statistics*, 9(1):60–62.
- [292] Jan P Williams, Olivia Zahn, and J Nathan Kutz. 2024. Sensing with shallow recurrent decoder networks. *Proceedings of the Royal Society A*, 480(2298):20240054.
- [293] Jan P Williams, Olivia Zahn, and J Nathan Kutz. 2024. Sensing with shallow recurrent decoder networks. *Proceedings of the Royal Society A*, 480(2298):20240054.
- [294] Andrew G Wilson and Pavel Izmailov. 2020. Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708.
- [295] Charlie Wood. 2022. Powerful ‘machine scientists’ distill the laws of physics from raw data. *Quanta Magazine*.
- [296] John Wright and Yi Ma. 2022. *High-dimensional data analysis with low-dimensional models: Principles, computation, and applications*. Cambridge University Press.

- [297] Dongxia Wu, Liyao Gao, Xinyue Xiong, Matteo Chinazzi, Alessandro Vespignani, Yi-An Ma, and Rose Yu. 2021. Quantifying uncertainty in deep spatiotemporal forecasting. *arXiv preprint arXiv:2105.11982*.
- [298] Jiajun Wu, Erika Lu, Pushmeet Kohli, Bill Freeman, and Josh Tenenbaum. 2017. Learning to see physics via visual de-animation. *Advances in neural information processing systems*, 30.
- [299] Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. 2015. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. *Advances in neural information processing systems*, 28.
- [300] Tailin Wu, Takashi Maruyama, and Jure Leskovec. 2022. Learning to accelerate partial differential equations via latent global evolution. *Advances in Neural Information Processing Systems*, 35:2240–2253.
- [301] Philippe Martin Wyder, Judah Goldfeder, Alexey Yermakov, Yue Zhao, Stefano Riva, Jan P Williams, David Zoro, Amy Sara Rude, Matteo Tomasetto, Joe Germany, et al. 2025. Common task framework for a critical evaluation of scientific machine learning algorithms. *arXiv preprint arXiv:2510.23166*.
- [302] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. 2024. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4389–4398.
- [303] Haohang Xu, Shuangrui Ding, Xiaopeng Zhang, Hongkai Xiong, and Qi Tian. 2022. Masked autoencoders are robust data augmentors. *arXiv preprint arXiv:2206.04846*.
- [304] Liu Yang, Xuhui Meng, and George Em Karniadakis. 2021. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics*, 425:109913.

- [305] Liu Yang, Dongkun Zhang, and George Em Karniadakis. 2020. Physics-informed generative adversarial networks for stochastic differential equations. *SIAM Journal on Scientific Computing*, 42(1):A292–A317.
- [306] Chen Yao and Erik M Bollt. 2007. Modeling and nonlinear parameter estimation with kronecker product representation for coupled oscillators and spatiotemporal systems. *Physica D: Nonlinear Phenomena*, 227(1):78–99.
- [307] David Ye, Jan Williams, Mars Gao, Stefano Riva, Matteo Tomasetto, David Zoro, and J Nathan Kutz. 2025. Pyshred: A python package for shallow recurrent decoding for sparse sensing, model reduction and scientific discovery. *arXiv preprint arXiv:2507.20954*.
- [308] Alexey Yermakov, David Zoro, Mars Liyao Gao, and J Nathan Kutz. 2025. T-SHRED: Symbolic regression for regularization and model discovery with transformer shallow recurrent decoders. *arXiv preprint arXiv:2506.15881*.
- [309] PK Yeung, DA Donzis, and KR Sreenivasan. 2012. Dissipation, enstrophy and pressure statistics in turbulence simulations at high reynolds numbers. *Journal of Fluid Mechanics*, 700:5–15.
- [310] Jian Yu and Jan S Hesthaven. 2019. Flowfield reconstruction method using artificial neural network. *Aiaa Journal*, 57(2):482–498.
- [311] Rose Yu and Rui Wang. 2024. Learning dynamical systems from data: An introduction to physics-guided deep learning. *Proceedings of the National Academy of Sciences*, 121(27):e2311808121.
- [312] Laure Zanna and Thomas Bolton. 2020. Data-driven equation discovery of ocean mesoscale closures. *Geophysical Research Letters*, 47(17):e2020GL088376.
- [313] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128.

- [314] Linan Zhang and Hayden Schaeffer. 2019. On the convergence of the sindy algorithm. *Multiscale Modeling & Simulation*, 17(3):948–972.
- [315] Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. 2019. Cyclical stochastic gradient mcmc for bayesian deep learning. *arXiv preprint arXiv:1902.03932*.
- [316] Xingyue Zhang, Yuxuan Bao, Mars Liyao Gao, and J Nathan Kutz. 2026. SENDAI: A hierarchical sparse-measurement, EfficienT data AssImilation framework. *arXiv preprint arXiv:2601.21664*.
- [317] Yuchen Zhang, Percy Liang, and Moses Charikar. 2017. A hitting time analysis of stochastic gradient langevin dynamics. In *Conference on Learning Theory*, pages 1980–2022. PMLR.
- [318] Bidan Zhao, Mingming He, and Junwu Wang. 2022. Data-driven discovery of the governing equation of granular flow in the homogeneous cooling state using sparse regression. *Physics of Fluids*.
- [319] Peng Zhao and Bin Yu. 2006. On model selection consistency of lasso. *The Journal of Machine Learning Research*, 7:2541–2563.
- [320] Peng Zheng, Travis Askham, Steven L Brunton, J Nathan Kutz, and Aleksandr Y Aravkin. 2018. A unified framework for sparse relaxed regularized regression: Sr3. *IEEE Access*, 7:1404–1423.
- [321] Zemin Zheng, Yingying Fan, and Jinchi Lv. 2014. High dimensional thresholded regression and shrinkage effect. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76(3):627–649.
- [322] Shuheng Zhou. 2023. Thresholded lasso for high dimensional variable selection. *arXiv preprint arXiv:2309.15355*.
- [323] Yufa Zhou, Yixiao Wang, Surbhi Goel, and Anru R Zhang. 2025. Why do transformers fail to forecast time series in-context? *arXiv preprint arXiv:2510.09776*.

-
- [324] Yinhao Zhu and Nicholas Zabaras. 2018. Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *Journal of Computational Physics*, 366:415–447.
- [325] Barret Zoph and Quoc V Le. 2016. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*.
- [326] Hui Zou. 2006. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429.