Bayesian autoencoders for data-driven discovery of coordinates, governing equations and fundamental constants

L. Mars Gao¹ and J. Nathan Kutz²

¹Paul G. Allen School of Computer Science and Engineering, and ²Department of Applied Mathematics and Electrical and Computer Engineering, University of Washington, Seattle, WA, USA

Recent progress in autoencoder-based sparse identification of nonlinear dynamics (SINDy) under $\ell_1$ constraints allows joint discoveries of governing equations and latent coordinate systems from spatio-temporal data, including simulated video frames. However, it is challenging for $\ell_1$-based sparse inference to perform correct identification for real data due to the noisy measurements and often limited sample sizes. To address the data-driven discovery of physics in the low-data and high-noise regimes, we propose Bayesian SINDy autoencoders, which incorporate a hierarchical Bayesian Spike-and-slab Gaussian Lasso prior. Bayesian SINDy autoencoder enables the joint discovery of governing equations and coordinate systems with uncertainty estimate. To resolve the challenging computational tractability of the Bayesian hierarchical setting, we adapt an adaptive empirical Bayesian method with Stochastic Gradient Langevin Dynamics (SGLD) which gives a computationally tractable way of Bayesian posterior sampling within our framework. Bayesian SINDy autoencoder achieves better physics discovery with lower data and fewer training epochs, along with valid uncertainty quantification suggested by the experimental studies. The Bayesian SINDy autoencoder can be applied to real video data, with...
accurate physics discovery which correctly identifies the governing equation and provides a close estimate for standard physics constants like gravity $g$, for example, in videos of a pendulum.

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 [1,2]. To identify the governing equation given the latent sparse representations, one can apply data-driven methods via sparse regression [3–5]. The sparse regression-based model discovery enables a computationally efficient way of identifying governing equations with convergence guarantees [6]. In this case of physics discovery from videos, Champion et al. [7] propose SINDy autoencoders which can jointly discover coordinates and equations for synthetic video data. More recently, Chen et al. [8] introduce the Neural State Variables learning from video data, which enables automated discovery of the latent state variables from the underlying dynamical system [8].

The automated discovery of coordinates and equations in real video is significantly more challenging compared to the prior works [8,9]. The challenges come 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 derivative estimates. Second, the number of training data for the real video dataset is very limited in comparison with synthetic videos [7]. For instance, in §4b, 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 [10–12]. 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 [13] (e.g. Bernoulli–Gaussian, Bernoulli–Laplace [14]) with hierarchical Bayesian settings has proven success for both sparse variable selection and uncertainty quantification [15]. In data-driven model discovery, Bayesian SINDy [16] with the spike-and-slab prior has also shown an advantage for correct model identification in the case study of Lynx-hare population [17] under the very low-data limit. However, a significant limitation of Bayesian methods comes from their high computational costs that hinder 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 non-trivial to design a feasible Bayesian solution for the discovery of governing equations and coordinate systems given the computational intractability.

In this paper, we propose Bayesian SINDy autoencoders that extend SINDy autoencoders [7] 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 [18,19]. 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 real video data on a single moving pendulum, we obtain correct governing equation discovery with a close estimate of the gravity constant $g = -9.876$ 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 s recording on our experiment on a pendulum.

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 on how they are combined into our Bayesian SINDy autoencoder framework.

(a) Sparse identification of nonlinear dynamics

We review the sparse identification of nonlinear dynamics (SINDy) [20] algorithm, which uses sparse regression to identify the latent dynamical system from snapshot data. SINDy takes snapshot data $x(t) \in \mathbb{R}^n$ and aims to discover the underlying dynamical system

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

The snapshots are collected by measurements at time $t \in [t_1, t_m]$, and the function $f$ characterizes the dynamics. Assuming the temporal derivatives of the snapshot are available from data, SINDy
forms data matrices in the following way:

\[
X = \begin{pmatrix} x(t_1) \\ x(t_2) \\ \vdots \\ x(t_m) \end{pmatrix} \quad \text{and} \quad \dot{X} = \begin{pmatrix} \dot{x}_1(t_1) \\ \dot{x}_1(t_2) \\ \vdots \\ \dot{x}_1(t_m) \end{pmatrix} ,
\]

with \( X, \dot{X} \in \mathbb{R}^{m \times n} \). The candidate function library is constructed by \( p \) candidate model term \( \theta_j \)'s that \( \Theta(X) = [\theta_1(X) \cdots \theta_p(X)] \in \mathbb{R}^{m \times p} \). A common choice of candidate functions are polynomials in \( x \) targeting common canonical models of dynamical systems [21]. The Fourier library is also very common with \( \sin(\cdot) \) and \( \cos(\cdot) \) terms. In summary, we build a model between \( X \) and \( \dot{X} \) that

\[
\dot{X} = \Theta(X) \Xi ,
\]

where the unknown matrix \( \Xi = (\xi_1, \xi_2, \ldots, \xi_n) \in \mathbb{R}^{p \times n} \) is the set of coefficients. The sparse inference on \( \Xi \) enables sparse identification of the dynamical system \( f \). For high-dimensional systems, the goal is to jointly identify a low-dimensional state \( z = \phi(x) \) with dynamics \( \dot{z} = g(z) \). The standard SINDy approach uses a sequentially thresholded least-squares algorithm to perform sparse inference [20], which is a proxy for \( \ell_0 \) optimization [22] with convergence guarantees [6].

In an alternative approach, Hirsh et al. [16] use 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 \( p(\xi) \) like the Spike-and-slab [13], regularized horseshoe [28,29], Laplace priors [30,31] and so on. After defining the Bayesian likelihood function with sparsifying priors, Bayesian SINDy generates posterior distribution via the No-U-Turn MCMC sampler [32]. Since MCMC for sparse inference can be extremely computationally demanding, [15,33] propose coordinate descent sparse inference methods via Spike-and-slab prior, targeting the mode detection. As an approximation to Bayesian inference, Fasel et al. [34] 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 [35–40], nonlinear optics [41], turbulence closures [42–44], ocean closures [45], chemical reaction [46], plasma dynamics [47–49], structural modelling [50] and model predictive control [51]. There are many extensions of SINDy, including the identification of partial differential equations [4,5], multiscale physics [9], parametrically dependent dynamical models [52], time-dependent PDEs [53], switching dynamical systems [54], rational function nonlinearities [55,56], control inputs [51], constraints on symmetries [35], control for stability [57], control for robustness [48,58–61], stochastic dynamical systems [62,63] and multidimensional approximation on tensors [64].

A related and important extension to the SINDy framework is the SINDy autoencoder [7,65], 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. Owing 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, the SINDy autoencoder is satisfactory due to the transparency of the learning process, and the parsimony in the dynamical system modeling [66,67]. 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.

(b) Bayesian and sparse deep learning

Bayesian deep learning achieves outstanding success in various machine learning tasks like computer vision [68], physics-informed modeling [69] and complex dynamical system control [70]. 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 [71, 72], which is applicable to computer vision [73], spatiotemporal forecasting [74], weather forecasting [75], and so on [76]. Second, the Bayesian framework allows theoretically grounded ensemble neural networks via model averaging [77]. Bayesian neural networks apply weights distribution to neurons [78]. 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 (SGLD) as an MCMC sampler in mini-batch settings [18]. Neal introduces MCMC via Hamiltonian dynamics [79], and Chen et al. extend into deep learning via Stochastic gradient Hamiltonian Monte Carlo (SGHMC) [80, 81]. Other techniques include Nosé–Hoover thermostat [82], replica-exchange SGHMC [83], cyclic SGLD [84], contour SGLD [85], preconditioned SGLD [86] and adaptively weighted SGLD [87]. Variational methods could also help to perform approximate inference for Bayesian deep learning [78, 88].

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 [89, 90]. Glorot et al. use Rectifier Neurons with sparsity constraints to obtain sparse representations [91]. Liu et al. extend to Convolutional neural networks under sparse settings [92]. Then, several follow-up works aimed to train sparse neural networks efficiently [93, 94]. By adapting Bayesian sparse inference methods, the sparse inference process could be accelerated with valid uncertainty quantification via various prior options [95–97]. Variational methods are also applied for efficient Bayesian inference [98]. There have also been theoretical discussions of Bayesian sparse deep learning [99–101]. Overall, sparse deep learning has the potential to make the construction of a Bayesian model computationally tractable.

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 §3a. Then, we propose a Bayesian learning framework that includes the 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.

(a) Likelihood setting for Bayesian SINDy autoencoders

The SINDy autoencoder enables a joint discovery of sparse dynamical models and coordinates. Figure 1a 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 $g_{\theta_2}(\cdot)$.

The 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

\[
\frac{d}{dt} z(t) = \Phi(z(t)) = \Theta(z(t)) \Xi,
\]

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

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

\[
p(D|\theta) \propto \exp \left( \| x - g_\theta(z) \|^2 + \lambda_1 \| \dot{x} - (\nabla z g_\theta(z))(\Theta(z^T) \Xi) \|^2 + \lambda_2 \| \nabla x \dot{x} - \Theta(z^T) \Xi \|^2 \right) .
\]

The log-likelihood of this statistical model is similar to the setting in [7]. In order to promote sparsity on \( \Xi \), we consider two sets of priors in what follows.

(i) Laplace prior

The Laplace prior can be understood as a Bayesian LASSO [30,31]. We define the Laplace prior such that

\[
\Xi_j \sim \mathcal{L}(0, v_0),
\]

where \( \mathcal{L}(\cdot, \cdot) \) denotes the Laplace distribution defined as \( f(z; 0, v_0) = (1/2v_0) \exp(-|z|/v_0) \). We can see the equivalence of the Laplace prior and the LASSO in the negative log-likelihood, where \( (1/v_0)\|\Xi_1\|_1 \) is included as a regularizer.

(ii) Spike-and-slab Gaussian–Laplace (SSGL) prior

We define the SSGL prior as

\[
\Xi_j | \gamma_j \sim (1 - \gamma_j) \mathcal{L}(0, \sigma v_0) + \gamma_j \mathcal{N}(0, \sigma^2 v_1),
\]
where $y_j$ is a binary variable, $\mathcal{E} \in \mathbb{R}^p$, $\sigma, v_0, v_1 \in \mathbb{R}$, $\mathcal{L}(\cdot, \cdot)$ denotes a Laplace distribution and $\mathcal{N}(\cdot, \cdot)$ denotes a Normal distribution. We assign a Bernoulli prior to $y \sim \text{Ber}(\delta)$, $\delta \in [0, 1]$. The prior for $\theta_1, \theta_2$ is specified with a Gaussian, which is equivalent to an $\ell_2$ regularization implementation-wise. For simplicity, we set $\delta, \sigma, v_0, v_1$ as tunable hyperparameters.

### (iii) 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 $\mathcal{E}$ to be a Laplace distribution, the setting will be identical to the basic SINDy autoencoder model, which is equivalent to adding a $\ell_1$ regularization term (c.f. eqn (7) and fig. 1 in [7]). 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 $\mathcal{E}$, 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.

### (iv) Bayesian formulation
Using the Bayes formula, we can construct the posterior distribution from the likelihood function and prior that

$$
\pi(\theta, y|D) \propto p(D|\theta)p(\theta_1)p(\theta_2)p(\mathcal{E}|y)p(y).
$$

We aim to depict the posterior distribution $p(\mathcal{E}|D)$ via the joint distribution $p(\theta|D)$ under the sparsifying SSGL prior. The approximation of $p(\mathcal{E}|D)$ is accessible from posterior samples of $p(\theta|D)$ when dropping $\theta_1, \theta_2$. 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.

### (b) 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 [18]. Denote the learning rate at epoch $t$ by $\epsilon(t)$ which decreases to zero, and the dataset $D = \{d_i\}_{i=1}^N$. The mini-batch setting estimates the gradient $\nabla_{\theta} L(\theta)$ from a subset (batch) $B = \{d_i\}_{j=1}^n$. 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

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

Here, $p(\theta_t)$ denotes the prior specified in equations (3.3), (3.4) 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 [102–104]. 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 (cf. section 4.1 [18]). The Metropolis–Hasting correction can be ignored since the rejection rate for sampling goes to zero asymptotically, resulting from $\epsilon(t) \to 0$ when $t \to \infty$. The discretization error similarly decreases as $\epsilon$ goes to zero.
(i) Cyclical SGLD

The cyclical SGLD method [84] 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 $\epsilon$ is large as well as sample local mode when $\epsilon$ is small. It is also possible to understand cyclical SGLD from a parallel SGLD perspective [83]. We apply this idea in our experiments to perform better inference.

(c) Empirical Bayes variable selection in SINDy autoencoder

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

Using the mini-batch setting, the posterior distribution follows

$$
\pi(\theta, \gamma|B) \propto p(B|\theta)^{\mathcal{N}/n} p(\theta_1)p(\theta_2)p(\mathcal{E}|\gamma)p(\gamma).
$$

(3.7)

The term $p(B|\theta)$ can be evaluated from the loss of SINDy autoencoder for the current mini-batch $B$; the term $p(\theta_1), p(\theta_2)$ can be computed from the $\ell_2$ loss; the term $p(\mathcal{E}|\gamma)$ can be computed from equation (3.4) and the term $p(\gamma)$ can be known from the Bernoulli prior setting.

As a binary variable, $\gamma$ is difficult to optimize for due to non-continuity and non-convexity. An important trick to perform the optimization on $\gamma$ is to alternatively optimize the adaptive posterior mean $\mathbb{E}_{\gamma|\theta^{(k)}}[\pi(\theta, \gamma|D)]$ which treats $\gamma$ as a latent variable. Following previous works with similar settings [15, 83], we could indirectly evaluate $\pi(\theta, \gamma|D)$ by a strict lower bound $Q(\cdot | \cdot)$ that

$$
Q(\theta|\theta^{(k)}) = \mathbb{E}_B[\mathbb{E}_{\gamma|\theta^{(k)}}[\log \pi(\theta, \gamma|B)]] \leq \log \mathbb{E}_{\gamma|\theta^{(k)}}[\mathbb{E}_B[\pi(\theta, \gamma|B)]].
$$

(3.8)

The inequality holds by Fubini’s theorem and Jensen’s inequality (cf. eqn (7) in [83]).

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

$$
Q(\theta|\theta^{(k)}) = \frac{N}{n} \log \pi(B|\theta) - ||\theta_1|| - ||\theta_2|| - \sum_{i \in P} \left[ \frac{|\mathcal{E}_i| \kappa_0}{\sigma} + \frac{\alpha_i^2}{2\sigma^2} \right] + \sum_{i \in P} \log \left( \frac{\delta}{1-\delta} \right) \rho_i + C,
$$

(3.9)

where $\kappa_0 = \mathbb{E}_{\gamma|\theta^{(k)}, D}[1/\upsilon_0(1 - \gamma_i)|1], \kappa_1 = \mathbb{E}_{\gamma|\theta^{(k)}, D}[1/\gamma_i], \rho_i = \mathbb{E}_{\gamma|\theta^{(k)}, D}[\gamma_i]$ and $C \in \mathbb{R}$ is a constant. Notice here $\gamma$ is treated as a latent variable, and we only consider the expectation given the conditional distribution of $\gamma$ given $\theta^{(k)}, D$. The $\rho_i$ could be considered as an inclusion probability estimate of $\mathcal{E}_i$. In this way, $\rho$ softens $\gamma$ from a binary variable into a continuous one.

The term $\kappa$ performs an elastic net-like approach which adaptively optimizes the $\ell_1$ and $\ell_2$ coefficients. Suppose $\mathcal{E}_i$ is identified with a high probability to be a sparse variable (e.g. $\rho_i < 0.05$), the term $\kappa_0$ will be large, strengthening the sparsity constraints. Otherwise, if $\mathcal{E}_i$ is identified as a non-sparse variable (e.g. $\rho_i > 0.95$), the term $\kappa_1$ will be small, and the $\ell_2$ constraint will be dominant instead.

(i) Stochastic approximation from expectation maximization of $\rho$ and $\kappa$

We could derive an asymptotically correct posterior distribution on $\pi(\theta, \kappa, \rho)$ following the steps in [83]:

1. Sample $\theta$ from $Q(\cdot)$ that

$$
\theta^{(k+1)} = \theta^{(k)} + \eta^{(k)} \nabla_{\theta} \mathbb{E}_Q(\rho^{(k)}, \kappa^{(k)}|B^{(k)}) + \mathcal{N}(0, 2\eta^{(k)}),
$$

(3.10)
and 

$$\rho^{(k+1)} = (1 - \omega^{(k+1)})\rho^{(k)} + \omega^{(k+1)}\tilde{\rho}^{(k+1)} \tag{3.11}$$

and 

$$\kappa^{(k+1)} = (1 - \omega^{(k+1)})\kappa^{(k)} + \omega^{(k+1)}\tilde{\kappa}^{(k+1)}. \tag{3.12}$$

Here, $\tilde{\rho}^{(k+1)}, \tilde{\kappa}^{(k+1)}$ is the Expectation Maximization (EM) estimation of $\rho, \kappa$ given iteration $k$ [15].

The EM estimation of the inclusion probability $\rho_i$ is

$$\hat{\rho}_i^{(k+1)} = \mathbb{E}_{\gamma | \theta, \mathcal{B}}[\gamma_i] = P(\gamma_i = 1 | \mathcal{S}_i^{(k)}) = \frac{a_i}{a_i + b_i}, \tag{3.13}$$

where $a_i = \pi(\mathcal{S}_i^{(k)} | \gamma_i = 1)P(\gamma_i = 1 | \delta)$ and $b_i = \pi(\mathcal{S}_i^{(k)} | \gamma_i = 0)P(\gamma_i = 0 | \delta)$. In our case, the term $\pi(\mathcal{S}_i^{(k)} | \gamma_i = 1)$ is the probability of $\mathcal{S}_i^{(k)}$ given the Gaussian prior distribution, and the term $\pi(\mathcal{S}_i^{(k)} | \gamma_i = 0)$ is the probability of $\mathcal{S}_i^{(k)}$ 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 $\kappa$, we have

$$\hat{\kappa}_0^{(k+1)} = \mathbb{E}_{\gamma | \theta, \mathcal{B}}\left[\frac{1}{v_0(1 - \gamma_i)}\right] = \frac{1 - \rho_i}{v_0} \tag{3.14}$$

and

$$\hat{\kappa}_1^{(k+1)} = \mathbb{E}_{\gamma | \theta, \mathcal{B}}\left[\frac{1}{v_1 \gamma_i}\right] = \frac{\rho_i}{v_1}. \tag{3.15}$$

(d) Prediction with Bayesian SINDy autoencoder

The 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 figure 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 §3c, we can generate samples from $p(\theta | \mathcal{D})$, which is the posterior distribution of neural network parameters $\theta$ given observed data $\mathcal{D}$. Suppose the posterior samples are $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_m \sim p(\theta | \mathcal{D})$. The full posterior predictive distribution is defined as

$$p(\hat{x}(t) | x_0, \mathcal{D}) = \int p(\hat{x}(t) | \theta, x_0)p(\theta | \mathcal{D}) \, d\theta$$

$$= \int p(\hat{x}(t) | \theta_2, \hat{z}(t))p(\hat{z}(t) | \mathcal{S}, \hat{z}(0))p(\hat{z}(0) | \theta_1, x_0)p(\theta_1, \theta_2, \mathcal{S} | \mathcal{D}) \, d\theta \tag{3.16}$$
Therefore, we can approximately generate samples from $p(\hat{x}(t)|x_0, D)$ using Monte Carlo estimation. For simplicity, we only consider the maximum-likelihood estimation $\hat{\theta}_1^{\text{MLE}}, \hat{\theta}_2^{\text{MLE}}$. We have the following process:

1. From the input image $x_0$, compute $z(0) = f^{\text{MLE}}_\Theta(x_0)$.
2. From $z(0)$, using posterior samples of $\mathcal{Z}$, generate $\hat{z}^{(i)}(t)$ samples via
   $$
   \hat{z}^{(i)}(t) = z(0) + \int_0^t \Theta(z(t'))\mathcal{Z}^{(i)}(t') dt'.
   $$
3. From samples of $\hat{z}^{(i)}(t)$, generate $\hat{x}(t) = g^{\text{MLE}}_{\hat{\theta}_2}(\hat{z}^{(i)}(t))$.

4. Experiments
In the following subsections, we conduct four case studies on the Bayesian SINDy autoencoder for governing equations and coordinate system discovery for video data. In §4a, we study similar cases to those in [7] 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 §4b, 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 systems 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.

(a) Learning physics from synthetic video data

(i) Chaotic Lorenz system via the Laplace prior

We consider the chaotic Lorenz system in the following:

\[
\begin{align*}
\dot{z}_1 &= -\sigma z_1 + \sigma z_2, \\
\dot{z}_2 &= \rho z_1 - z_2 - z_1 z_3, \\
\dot{z}_3 &= z_1 z_2 - \beta z_3,
\end{align*}
\]

(4.1)

where $z = [z_1, z_2, z_3] \in \mathbb{R}^3$ and $\sigma, \rho, \beta$ 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 $\sigma = 10, \rho = 28, \beta = -8.7$. 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 centred 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 $u_1, u_2, \ldots, u_6 \in \mathbb{R}^{128}$. We transfer from the low-dimensional dynamical system into a high-dimensional dataset via the following rule:

\[
x(t) = u_1 z_1(t) + u_2 z_2(t) + u_3 z_3(t) + u_4 z_1(t)^3 + u_5 z_2(t)^2 + u_6 z_3(t)^3.
\]

(4.2)

We set the autoencoder with latent dimension $d = 3$ corresponding to the latent system in the three-dimensional coordinate system with $z_A, z_B, z_C$. We include polynomials with the highest order 3 composing a library $[1, z_A, z_B, z_C, z_A^2, z_A z_B, z_A z_C, z_B^2, z_B z_C, z_C^2, z_A^3, z_A^2 z_B, z_A^2 z_C, \ldots, z_3^3]$. Via the autoencoder, we wish to identify the correct active terms as well as the value of the coefficients. The coefficient of $\mathcal{Z}$ 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 $1 \times 10^{-3}$ and the batch size to be 1024. For the Laplace prior setting, we set $\lambda_3 = 1 \times 10^{-5}$.

By training with 5000 epochs following the setting from [7], in terms of the error metrics, the best test error of the decoder reconstruction achieves $2 \times 10^{-5}$ of the fraction of the variance from the input. The fraction of unexplained variances are $2 \times 10^{-4}$ for the reconstruction of $\dot{z}$, and $1.3 \times 10^{-3}$ for the reconstruction of $\dot{x}$. Notice here decoder reconstruction is better compared to the SINDy autoencoder without uncertainty quantification, but the reconstruction of $\dot{z}$ is slightly worse compared to point estimation for the SINDy autoencoder.

The coefficient estimate from the Bayesian SINDy autoencoder is shown in figure 3. From the figure, we can observe the uncertainty quantification in the parameter space. The outcome of this model identifies seven 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 [7]. 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 figure 3 from the following transformations:

(a) We inversely transform the permutation group via assigning $z_A$ to $z_1$, $z_B$ to $z_2$ and $z_C$ 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 figure 4c 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

$$\begin{align*}
\dot{z}_1 &= -10.09z_1 + 10.00z_2, \\
\dot{z}_2 &= 27.09z_1 - 0.86z_2 - 5.35z_1z_3 \\
\dot{z}_3 &= 5.35z_1z_2 - 2.71z_3.
\end{align*}$$

(4.3)

Stepping further from the uncertainty quantification of coefficients, we can see the uncertainty quantification in the prediction space in figure 4a,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.
(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.

(ii) Reaction–diffusion via the 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

\[
\begin{align*}
    u_t &= (1 - (u^2 + v^2))u + \beta(u^2 + v^2)v + d_1(u_{xx} + u_{yy}) \\
    v_t &= -\beta(u^2 + v^2)u + (1 - (u^2 + v^2))v + d_2(v_{xx} + v_{yy}),
\end{align*}
\]

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 4.4 from \(t = [0, 500]\) with time step \(\Delta t = 0.05\). Then, we generate the snapshots of the dynamical system spatially in the \(xy\)-domain from its latent dimensions into a video with shape \((10000, 100, 100)\). We obtain a training dataset with size 9000. We also generate a testing dataset with 1000 samples.

We set up the autoencoder with latent dimension \(d = 2\), targeting the \(u, v\), and apply a first-order library of functions including \([1, u_x, v_x, u^2_x, u_x v_x, v_x^2, u^2_x v_x, u_x v_x^2, v^2_x, \sin(u_x), \sin(v_x)]\). We hope to discover the two oscillating spatial modes for this nonlinear oscillatory dynamics. The coefficient of \(\Xi\) 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 \(\delta = 0.08, \nu_0 = 0.1, \nu_1 = 3.0, \omega(0) = 0.02 \times (0.999)^k\).

We follow the same setting in [7]. The SSGL prior only requires 1500 epochs for training while LASSO setting frequently needs more than 3000 epochs for training. The best test error is \(2.1 \times 10^{-3}\) for decoder loss, \(2.1 \times 10^{-5}\) for the reconstruction of \(\dot{z}\), and \(1.7 \times 10^{-4}\) for the reconstruction of \(\dot{x}\). The fraction of unexplained variance of decoder reconstruction is \(9.1 \times 10^{-5}\). 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 [7]. The posterior samples from Bayesian SINDy are shown in figure 5. The discovered governing
Figure 5. (a) Bayesian estimation and uncertainty quantification visualization of SINDy coefficient for reaction–diffusion under Spike-and-slab prior. The colour 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 equation from the Bayesian SINDy autoencoder is

\[
\begin{align*}
\dot{u}_x &= 0.91 \sin(u_x) \\
\dot{z}_2 &= -0.91 \sin(v_x).
\end{align*}
\]

(iii) Nonlinear pendulum via the Spike-and-slab prior

We consider a 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:

\[
\ddot{z} = -\sin z.
\]

We generate the synthetic dataset following the settings in [7]. 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 100 times, we obtain a training dataset with size 50 000. We also generate a testing dataset with 5000 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 $[1, z, \dot{z}, z^2, z^2, z^2, z^2, z^2, z^2, \sin(z), \sin(\dot{z})]$. We hope to infer the real coefficient that $\mathcal{E} = [0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0]$. The coefficient of $\mathcal{E}$ is randomly initialized from Gaussian. The loss coefficients are $\lambda_1 = 5 \times 10^{-3}, \lambda_2 = 5 \times 10^{-5}, \lambda_3 = 8 \times 10^{-4}$. For the encoder and decoder, we use the sigmoid activation function with widths $[128, 64, 32]$. For optimization, we select the Adam optimizer with a learning rate $1e^{-3}$, and batch size to be 1000. For the setting of the SSGL prior, we set $\delta = 0.08, \nu_0 = 0.05, \nu_1 = 3.0$, $\omega(k) = 0.05 \times (0.995)^k$.

We follow the same setting in [7]. The success discovery rate is 80% from 15 training instances, which improves from 50% success discovery rate in the LASSO setting. Additionally, the Bayesian
SINDy autoencoder with the SSGL prior requires only 1500 epochs for training, while the LASSO setting frequently needs 5000 epochs for training. The best test error is $6.5 \times 10^{-8}$ for decoder loss, $1.1 \times 10^{-3}$ for the reconstruction of $\dot{x}$, and $8.1 \times 10^{-4}$ for the reconstruction of $\ddot{z}$. The best fraction of unexplained variance of decoder reconstruction is $4.5 \times 10^{-4}$. For SINDy predictions, the best fraction of unexplained variance of $\dot{x}$ and $\ddot{z}$ reconstruction are $2.3 \times 10^{-4}$ and $5.5 \times 10^{-3}$. The posterior samples from Bayesian SINDy are shown in figure 6a. The discovered governing equation from the Bayesian SINDy autoencoder is
\[ \ddot{z} = -0.99 \sin(z). \] (4.7)

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

(b) Learning dynamical system from GoPro recording on a moving rod

(i) Experimental setting and dataset description

The raw video data has 14 s recording of a moving rod with 390 frames in total. Owing 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 figure 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 [105,106]. This is step (b) to (c) in figure 7.
Figure 7. Preprocessing pipeline to high-dimensional video data to remove auxiliary information and reduce the size of video frames.

(3) Apply Gaussian filters with appropriate variance. The selection of variance varies from 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 figure 7.

(4) The final step downsamples 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 figure 7b, 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.

(ii) 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 $\lambda_1 = 5 \times 10^{-7}, \lambda_2 = 5 \times 10^{-8}$ 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 the Adam optimizer with a learning rate $1 \times 10^{-3}$, and batch size to be 10. Real video data are much harder compared with synthetic video data since it contains rich information, including object colours, 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 [107], 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 [108]. For the setting of the SSGL prior, we set $\delta = 0.09$, $v_0 = 1.5$, $v_1 = 5.0$, $\omega(k) = 0.01 \times (0.999)^k$.

(iii) Results from the Bayesian discovery via SSGL prior

Model selection and the regularization path. Following the standard practice [15], we study the regularization path of the SSGL prior by increasing the spike variance. Figure 8a 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 figure 8b. We select the sparsest model with the lowest testing
Figure 8. (a) SSSL regularization plot. (b) Model selection via test error.

Figure 9. (a) Bayesian estimation and uncertainty quantification visualization of SINDy coefficient for real moving rod data under Spike-and-slab prior. The colour 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.

error from the Occam’s razor principle. The full regularization path study indicates that there exists a governing physics law (with the term \( \sin(z) \)) in this video recording.

Discovered governing equation. The success discovery rate of Bayesian SINDy autoencoder is 100% from 10 training instances, and it only requires 1500 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 fractions 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 figure 9a, and the prediction of trajectory with uncertainty quantification of latent dynamics is shown in the bottom figure of figure 9b. The discovered governing equation from the SSGL prior is

\[
\ddot{z} = -16.06 \sin(z). \tag{4.8}
\]
Video reconstruction, prediction and uncertainty quantification. The latent dimension $\tilde{z}$ with SINDy correctly identifies $\sin(z)$ as the active term and generates an uncertainty estimate. Using the uncertainty estimation in the parameter space, we could perform uncertainty quantification on the prediction space as in figure 9b. The training outcomes of Bayesian SINDy autoencoder are shown in figures 9 and 10. First, from figure 9c, 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 figure 10 that the log loss converges with more training epochs. Throughout these experiments, we uniformly set training epochs to be 1500 and refinement epochs to be 1500. This decision is suggested by the observation in a longer run (7500 training epochs and 1500 refinement epochs) that the log loss remains stable after 1000 epochs. For the reconstruction of $\tilde{z}$, having longer training epochs will harm the testing error due to overfitting. We note here the discovery by the SSGL prior has dominant performance in testing loss comparing to the SINDy autoencoder as shown in appendix A. This suggests the superiority of the SSGL prior in the real video data setting.

As suggested in §3d, the Bayesian SINDy autoencoder enables a trustworthy solution in deep learning for video prediction. In figure 9b, we predict the following frames of video using the decoder and the simulation of latent dynamical systems. We observe that in the 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 figure 9b.

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 $\theta$ from all video frames by humans, as shown in the grey dashed curve in figure 11. The blue curve represents the rescaled latent dimension $z$, and we observe these two curves match mostly perfectly to each other. We apply rescaling in this process due to the coordinate system for the angles $\theta$ being 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 with concentrates at $(-9.889, -9.865)$. The bias is more likely to be removed with an enlarged video dataset.

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.

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’ [109]. 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 [110]. This could also practically reduce the manual effort required for hyperparameter tuning.

— **Full Bayesian specification.** The hyperparameter setting for the sparsity level (\( \delta \)), spike-and-slab distribution variances (\( \nu_0, \nu_1 \)), and environmental noise (\( \sigma \)) can be specified using a full Bayesian inference setup as described in [15,96]. This approach lessens the need for 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 using an ensemble of trained Bayesian SINDy autoencoders initialized from various starting points [74,77]. This process constructs a posterior distribution of the SINDy coefficient \( \mathcal{E} \), and could perform subset selection via stability selection or inclusion probability thresholding [34,111,112]. Applying variational autoencoders [113] could avoid the usage of hard-thresholding procedures. It also helps to perform better subset selection with Bayesian uncertainty estimation to the SINDy coefficient [113].

![Figure 11. Latent dimension visualization after rescaling (blue curve) versus manually labelled moving rod angle by human (grey curve).](image-url)
In future work, it will be important to involve new techniques to discover governing physics with multiple objects and complex chaotic systems. The Bayesian SINDy autoencoder also presumes a static coordinate system, making it challenging to analyse 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 s 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 autoencoders in diverse applications.

Data accessibility. Code is available from the GitHub repository: https://github.com/gaoliyao/BayesianSindyAutoencoder [114].

Supplementary material is available online [115].

Declaration of AI use. We have not used AI-assisted technologies in creating this article.

Conflict of interest declaration. We declare we have no competing interests.

Authors’ contributions. L.M.G.: conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization, writing—original draft, writing—review and editing; J.N.K.: conceptualization, data curation, funding acquisition, investigation, methodology, project administration, resources, supervision, validation, visualization, writing—review and editing.

Both authors gave final approval for publication and agreed to be held accountable for the work performed therein.

Funding. This study supported by the National Science Foundation AI Institute in Dynamic Systems (grant no. 2112085). J.N.K. supported by the Air Force Office of Scientific Research (grant no. FA9550-19-1-0011).

Acknowledgements. We acknowledge support from the National Science Foundation AI Institute in Dynamic Systems (grant no. 2112085). J.N.K. further acknowledges support from the Air Force Office of Scientific Research (FA9550-19-1-0011). We would also like to thank Dr Wei Deng, Dr Bethany Lusch, Prof. Simon Du, Prof. Lexing Ying and Dr Joseph Bakarji for their insightful suggestions and comments.

### Appendix A. 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, the Bayesian autoencoder with the Laplace prior fails to discover governing physics from real video data.

For the setting of the 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. The Bayesian SINDy autoencoder requires 1500 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 fractions of unexplained variance of decoder reconstruction is 0.012. For SINDy predictions, the fractions of unexplained variance of $\ddot{x}$ and $\ddot{z}$ reconstruction are 0.210 and 0.941.

The posterior samples from Bayesian SINDy are shown in figure 12a and the prediction of trajectory with uncertainty quantification of latent dynamics is shown in figure 12c. The discovered governing equation from the Laplace prior is

$$\ddot{z} = -7.10z. \quad (A1)$$

The discovered governing equation with $z$ slightly deviates from the true dynamics similarly as reported in [7], but still creates reasonable uncertainty estimate under model misspecification. The latent dimension in figure 12b follows a similar process in figure 11. The latent dimension $z$ still matches the human labels, but not as close as the learned result by the SSGL prior.
Figure 12. (a) Bayesian estimation and uncertainty quantification visualization of SINDy coefficient for real moving rod data under Laplace prior. The colour 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.

(a) Regularization path and model selection

Figure 13a 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 the $\sin(\theta)$ term uniformly through the entire regularization path. Additionally, combining with the testing loss plot in figure 13b, 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.

Appendix B. Pre-processing

(a) 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.
Figures 13. (a) Laplace prior (LASSO) regularization plot. (b) Model selection via test error.

— Videos contain more auxiliary (unnecessary) information for dynamical system learning including background, colours, 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 these level huge matrix operations w.r.t this size for neural network learning, e.g. $\nabla_x f(x)$.

### (b) 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 $\Delta t$.

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

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

1. Hinton GE, Salakhutdinov RR. 2006 Reducing the dimensionality of data with neural networks. Science 313, 504–507. (doi:10.1126/science.1127647)

2. Bao X, Lucas J, Sachdeva S, Grosse RB. 2020 Regularized linear autoencoders recover the principal components, eventually. Adv. Neural Inf. Process. Syst. 33, 6971–6981.

3. Kutz JN, Brunton SL, Brunton BW, Proctor JL. 2016 Dynamic mode decomposition: data-driven modeling of complex systems. Philadelphia, PA: Society for Industrial and Applied Mathematics.

4. Schaeffer H. 2017 Learning partial differential equations via data discovery and sparse optimization. Proc. R. Soc. A 473, 20160446. (doi:10.1098/rspa.2016.0446)

5. Rudy SH, Brunton SL, Proctor JL, Kutz JN. 2017 Data-driven discovery of partial differential equations. Sci. Adv. 3, e1602614. (doi:10.1126/sciadv.1602614)

6. Zhang L, Schaeffer H. 2019 On the convergence of the SINDy algorithm. Multiscale Model. Simul. 17, 948–972. (doi:10.1137/18M1189828)

7. Champion K, Lusch B, Kutz JN, Brunton SL. 2019 Data-driven discovery of coordinates and governing equations. Proc. Natl Acad. Sci. USA 116, 22 445–22 451. (doi:10.1073/pnas.1906995116)

8. Chen B, Huang K, Raghupathi S, Chandratreya I, Du Q, Lipson H. 2021 Discovering state variables hidden in experimental data. (http://arxiv.org/abs/2112.10755)

9. Champion KP, Brunton SL, Kutz JN. 2019 Discovery of nonlinear multiscale systems: sampling strategies and embeddings. SIAM J. Appl. Dyn. Syst. 18, 312–333. (doi:10.1137/18M1188227)

10. Mitchell TJ, Beauchamp JJ. 1988 Bayesian variable selection in linear regression. J. Am. Stat. Assoc. 83, 1023–1032. (doi:10.1080/01621459.1988.10478694)

11. George EI, McCulloch RE. 1997 Approaches for Bayesian variable selection. Stat. Sinica 18, 339–373.

12. Scott SL, Varian HR. 2014 Predicting the present with Bayesian structural time series. Int. J. Math. Modell. Numer. Optim. 5, 4–23. (doi:10.1504/IJMMNO.2014.059942)

13. Ishwaran H, Rao JS. 2005 Spike and slab variable selection: frequentist and Bayesian strategies. Ann. Stat. 33, 730–773. (doi:10.1214/009053604000001147)

14. Amini A, Kamilov US, Unser M. 2012 The analog formulation of sparsity implies infinite divisibility and rules out Bernoulli-Gaussian priors. 2012 IEEE Information Theory Workshop. Lausanne, Switzerland, 3 September 2012, pp. 682–686. New York, NY: IEEE. (doi:10.1109/ITW.2012.6404765)

15. Roˇcková V, George EI. 2014 EMVS: the EM approach to Bayesian variable selection. J. Am. Stat. Assoc. 109, 828–846. (doi:10.1080/01621459.2013.869223)

16. Hirsh SM, Barajas-Solano DA, Kutz JN. 2022 Sparsifying priors for Bayesian uncertainty quantification in model discovery. R. Soc. Open Sci. 9, 211823. (doi:10.1098/rsos.211823)

17. Hewitt CG. 1921 The conservation of the wild life of Canada. New York, NY: C. Scribner.

18. Welling M, Teh YW. 2007 Modeling and nonlinear parameter estimation with Kronecker product representation for coupled oscillators and spatiotemporal systems. Physica D 227, 78–99. (doi:10.1016/j.physd.2006.12.006)

19. Schmidt M, Lipson H. 2009 Distilling free-form natural laws from experimental data. Science 324, 81–85. (doi:10.1126/science.1165893)
26. Wang WX, Yang R, Lai YC, Kovannis V, Grebogi C. 2011 Predicting catastrophes in nonlinear dynamical systems by compressive sensing. *Phys. Rev. Lett.* 106, 154101. (doi:10.1103/PhysRevLett.106.154101)

27. Williams JP, Zahn O, Kutz JN. 2023 Sensing with shallow recurrent decoder networks. (http://arxiv.org/abs/2301.12011)

28. Carvalho CM, Polson NG, Scott JG. 2009 Handling sparsity via the horseshoe. International Conference on Artificial intelligence and statistics (AISTATS) 2009, Clearwater Beach, FL, 16-18 April 2009, pp. 73–80. Cambridge, MA: Proceedings of Machine Learning Research.

29. Carvalho CM, Polson NG, Scott JG. 2010 The horseshoe estimator for sparse signals. *Biometrika* 97, 465–480. (doi:10.1093/biomet/asq017)

30. Park T, Casella G. 2008 The Bayesian lasso. *J. Am. Stat. Assoc.* 103, 681–686. (doi:10.1198/016214508000000337)

31. Tibshirani R. 1996 Regression shrinkage and selection via the lasso. *J. R. Stat. Soc.: Ser. B (Methodological)* 58, 267–288. (doi:10.1111/j.2517-6161.1996.tb02080.x)

32. Hoffman MD, Gelman A. 2014 The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.* 15, 1593–1623.

33. Roˇcková V, George EI. 2018 The spike-and-slab lasso. *J. Am. Stat. Assoc.* 113, 431–444. (doi:10.1080/01621459.2016.1260469)

34. Fasel U, Kutz JN, Brunton BW, Brunton SL. 2022 Ensemble-SINDy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proc. R. Soc. A* 478, 20210904. (doi:10.1098/rspa.2021.0904)

35. Loiseau JC, Brunton SL. 2018 Constrained sparse Galerkin regression. *J. Fluid Mech.* 838, 42–67. (doi:10.1017/jfm.2017.823)

36. Loiseau JC, Noack BR, Brunton SL. 2018 Sparse reduced-order modelling: sensor-based dynamics to full-state estimation. *J. Fluid Mech.* 844, 459–490. (doi:10.1017/jfm.2018.147)

37. Loiseau JC. 2020 Data-driven modeling of the chaotic thermal convection in an annular thermosyphon. *Theor. Comput. Fluid Dyn.* 34, 339–365. (doi:10.1007/s00162-020-00536-w)

38. Guan Y, Brunton SL, Novoselov I. 2021 Sparse nonlinear models of chaotic electroconvection. *R. Soc. Open Sci.* 8, 202367. (doi:10.1098/rsos.202367)

39. Deng N, Noack BR, Morzyński M, Pastur LR. 2021 Galerkin force model for transient and post-transient dynamics of the fluidic pinball. *J. Fluid Mech.* 918, A4. (doi:10.1017/jfm.2021.299)

40. Callaham JL, Rigas G, Loiseau JC, Brunton SL. 2022 An empirical mean-field model of symmetry-breaking in a turbulent wake. *Sci. Adv.* 8, eabm4786. (doi:10.1126/sciadv.abm4786)

41. Sorokina M, Sygletos S, Turitsyn S. 2016 Sparse identification for nonlinear optical communication systems: SINO method. *Opt. Express* 24, 30433–30443. (doi:10.1364/OE.24.030433)

42. Beetham S, Capecelatro J. 2020 Formulating turbulence closures using sparse regression with embedded form invariance. *Phys. Rev. Fluids* 5, 084611. (doi:10.1103/PhysRevFluids.5.084611)

43. Beetham S, Fox RO, Capecelatro J. 2021 Sparse identification of multiphase turbulence closures for coupled fluid–particle flows. *J. Fluid Mech.* 914, A11. (doi:10.1017/jfm.2021.53)

44. Schmelzer M, Dwight RP, Cinnella P. 2020 Discovery of algebraic Reynolds-stress models using sparse symbolic regression. *Flow, Turbul. Comb.* 104, 579–603. (doi:10.1007/s10494-019-00089-x)

45. Zanna L, Bolton T. 2020 Data-driven equation discovery of ocean mesoscale closures. *Geophys. Res. Lett.* 47, e2020GL088376. (doi:10.1029/2020GL088376)

46. Hoffmann M, Fröhner C, Noé F. 2019 Reactive SINDy: discovering governing reactions from concentration data. *J. Chem. Phys.* 150, 025101. (doi:10.1063/1.5066099)

47. Dam M, Brøns M, Juul Rasmussen J, Naulin V, Hesthaven JS. 2017 Sparse identification of a predator-prey system from simulation data of a convection model. *Phys. Plasmas* 24, 022310. (doi:10.1063/1.4977057)

48. Alves EP, Fiuza F. 2022 Data-driven discovery of reduced plasma physics models from fully kinetic simulations. *Phys. Rev. Res.* 4, 033192. (doi:10.1103/PhysRevResearch.4.033192)

49. Kaptanoglu AA, Morgan KD, Hansen CJ, Brunton SL. 2021 Physics-constrained, low-dimensional models for magnetohydrodynamics: first-principles and data-driven approaches. *Phys. Rev. E* 104, 015206. (doi:10.1103/PhysRevE.104.015206)
50. Lai Z, Nagarajaiah S. 2019 Sparse structural system identification method for nonlinear dynamic systems with hysteresis/inelastic behavior. *Mech. Syst. Signal Process.* 117, 813–842. (doi:10.1016/j.ymssp.2018.08.033)

51. Kaiser E, Kutz JN, Brunton SL. 2018 Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proc. R. Soc. A* 474, 20180335. (doi:10.1098/rspa.2018.0335)

52. Rudy S, Alla A, Brunton SL, Kutz JN. 2019 Data-driven identification of parametric partial differential equations. *SIAM J. Appl. Dyn. Syst.* 18, 643–660. (doi:10.1137/18M1191944)

53. Chen A, Lin G. 2021 Robust data-driven discovery of partial differential equations with time-dependent coefficients. (http://arxiv.org/abs/2102.01432)

54. Mangan NM, Askham T, Brunton SL, Kutz JN, Proctor JL. 2019 Model selection for hybrid dynamical systems via sparse regression. *Proc. R. Soc. A* 475, 20180534. (doi:10.1098/rspa.2018.0534)

55. Mangan NM, Brunton SL, Proctor JL, Kutz JN. 2016 Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Trans. Mol., Biol. Multi-Scale Commun.* 2, 52–63. (doi:10.1109/TMBMC.2016.2633265)

56. Kaheman K, Kutz JN, Brunton SL. 2020 SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proc. R. Soc. A* 476, 20200279. (doi:10.1098/rspa.2020.0279)

57. Kaptanoglu AA, Callaham JL, Aravkin A, Hansen CJ, Brunton SL. 2021 Promoting global stability in data-driven models of quadratic nonlinear dynamics. *Phys. Rev. Fluids* 6, 094401. (doi:10.1103/PhysRevFluids.6.094401)

58. Schaeffer H, McCalla SG. 2017 Sparse model selection via integral terms. *Phys. Rev. E* 96, 023302. (doi:10.1103/PhysRevE.96.023302)

59. Reinbold PA, Gorevich DR, Grigoriev RO. 2020 Using noisy or incomplete data to discover models of spatiotemporal dynamics. *Phys. Rev. E* 101, 010203. (doi:10.1103/PhysRevE.101.010203)

60. Reinbold PA, Kageorge LM, Schatz MF, Grigoriev RO. 2021 Robust learning from noisy, incomplete, high-dimensional experimental data via physically constrained symbolic regression. *Nat. Commun.* 12, 1–8. (doi:10.1038/s41467-021-23479-0)

61. Messenger DA, Bortz DM. 2021 Weak SINDy for partial differential equations. *J. Comput. Phys.* 443, 110525. (doi:10.1016/j.jcp.2021.110525)

62. Boninseuga L, Nüske F, Clementi C. 2018 Sparse learning of stochastic dynamical equations. *J. Chem. Phys.* 148, 241723. (doi:10.1063/1.5018409)

63. Callaham JL, Loiseau JC, Rigas G, Brunton SL. 2021 Nonlinear stochastic modelling with Langevin regression. *Proc. R. Soc. A* 477, 20210092. (doi:10.1098/rspa.2021.0092)

64. Gelß P, Klus S, Eisert J, Schütte C. 2019 Multidimensional approximation of nonlinear dynamical systems. *J. Comput. Nonlinear Dyn.* 14, 061006. (doi:10.1115/1.4043148)

65. Bakarji J, Champion K, Kutz JN, Brunton SL. 2022 Discovering governing equations from partial measurements with deep delay autoencoders. (http://arxiv.org/abs/2201.05136)

66. Kutz JN, Brunton SL. 2022 Parsimony as the ultimate regularizer for physics-informed machine learning. *Nonlinear Dyn.* 107, 1801–1817. (doi:10.1007/s11071-021-07118-3)

67. Kutz JN. 2023 Machine learning for parameter estimation. *Proc. Natl Acad. Sci. USA* 120, e2300990120. (doi:10.1073/pnas.2300990120)

68. Gustafsson FK, Danelljan M, Schon TB. 2020 Evaluating scalable Bayesian deep learning methods for image denoising. In *Proc. of the IEEE/CVF Conf. on computer vision and pattern recognition workshops*, pp. 318–319.

69. Yang L, Meng X, Karniadakis GE. 2021 B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. *J. Comput. Phys.* 425, 109913. (doi:10.1016/j.jcp.2020.109913)

70. Watter M, Springenberg J, Boedecker J, Riedmiller M. 2015 Embed to control: a locally linear latent dynamics model for control from raw images. *Adv. Neural Inform. Process. Syst.* 28, 2746–2754.

71. Wang H, Yeung DY. 2020 A survey on Bayesian deep learning. *ACM Comput. Surv. (CSUR)* 53, 1–37. (doi:10.1145/3409383)

72. Abdar M et al. 2021 A review of uncertainty quantification in deep learning: techniques, applications and challenges. *Inf. Fusion* 76, 243–297. (doi:10.1016/jинфus.2021.05.008)

73. Kendall A, Gal Y. 2017 What uncertainties do we need in bayesian deep learning for computer vision? *Adv. Neural Inform. Process. Syst.* 30, 5580–5590.
74. Wu D, Gao L, Xiong X, Chinazzi M, Vespignani A, Ma YA, Yu R. 2021 Quantifying uncertainty in deep spatiotemporal forecasting. (http://arxiv.org/abs/2105.11982)
75. Wang B, Lu J, Yan Z, Luo H, Li T, Zheng Y, Zhang G. 2019 Deep uncertainty quantification: a machine learning approach for weather forecasting. In Proc. of the 25th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining, pp. 2087–2095.
76. Zhu Y, Zabaras N. 2018 Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. J. Comput. Phys. 366, 415–447. (doi:10.1016/j.jcp.2018.04.018)
77. Wilson AG, Izmailov P. 2020 Bayesian deep learning and a probabilistic perspective of generalization. Adv. Neural Inform. Processing Syst. 33, 4697–4708.
78. Blundell C, Cornebise J, Kavukcuoglu K, Wierstra D. 2015 Weight uncertainty in neural network. In Proc. of the 32nd Int. Conf. on Machine Learning, pp. 1613–1622. PMLR.
79. Neal RM. 2011 MCMC using Hamiltonian dynamics. Handbook of Markov Chain Monte Carlo 2, 97–124. (doi:10.1201/b10905-6)
80. Chen T, Fox E, Guestrin C. 2014 Stochastic gradient hamiltonian monte carlo. In Int. Conf. on machine learning, pp. 1683–1691. PMLR.
81. Ma YA, Chen T, Fox E. 2015 A complete recipe for stochastic gradient MCMC. Adv. Neural Inform. Process. Syst. 28, 2917–2925.
82. Ding N, Fang Y, Babbush R, Chen C, Skeel RD, Neven H. 2014 Bayesian sampling using stochastic gradient thermostats. Adv. Neural Inform. Process. Syst. 27, 3203–3211.
83. Deng W, Feng Q, Gao L, Liang F, Lin G. 2020 Non-convex learning via replica exchange stochastic gradient mcmc. In Int. Conf. on Machine Learning, pp. 2474–2483. PMLR.
84. Zhang R, Li C, Zhang J, Chen C, Wilson AG. 2019 Cyclical stochastic gradient MCMC for Bayesian deep learning. (http://arxiv.org/abs/1902.03932)
85. Deng W, Lin G, Liang F. 2020 A contour stochastic gradient Langevin dynamics algorithm for simulations of multi-modal distributions. Adv. Neural Inform. Process. Syst. 33, 15725–15736.
86. Wang Y, Deng W, Lin G. 2021 Bayesian sparse learning with preconditioned stochastic gradient MCMC and its applications. J. Comput. Phys. 432, 110134. (doi:10.1016/j.jcp.2021.110134)
87. Deng W, Lin G, Liang F. 2022 An adaptively weighted stochastic gradient MCMC algorithm for Monte Carlo simulation and global optimization. Stat. Comput. 32, 1–24. (doi:10.1007/s11222-022-10120-3)
88. Hernández-Lobato JM, Adams R. 2015 Probabilistic backpropagation for scalable learning of Bayesian neural networks. In Int. Conf. on machine learning, pp. 1861–1869. PMLR.
89. Elizondo D, Fiesler E, Korczak J. 1995 Non-ontogenic sparse neural networks. In Proc. of ICNN’95-Int. Conf. on Neural Networks, vol. 1, pp. 290–295. IEEE.
90. Morgan PH. 2008 Differential evolution and sparse neural networks. Expert Syst. 25, 394–413. (doi:10.1111/j.1468-0394.2008.00466.x)
91. Glorot X, Bordes A, Bengio Y. 2011 Deep sparse rectifier neural networks. In Proc. of the fourteenth Int. Conf. on artificial intelligence and statistics, pp. 315–323. JMLR Workshop and Conference Proceedings.
92. Liu B, Wang M, Foroosh H, Tappen M, Pensky M. 2015 Sparse convolutional neural networks. In Proc. of the IEEE Conf. on computer vision and pattern recognition, pp. 806–814.
93. Louizos C, Welling M, Kingma DP. 2017 Learning sparse deep neural networks through L_0 regularization. (http://arxiv.org/abs/1712.01312)
94. Srinivas S, Subramanyam A, Venkatesh Babu R. 2017 Training sparse neural networks. In Proc. of the IEEE Conf. on computer vision and pattern recognition workshops, pp. 138–145.
95. Sun Y, Song Q, Liang F. 2022 Learning sparse deep neural networks with a spike-and-slab prior. Stat. Prob. Lett. 180, 109246. (doi:10.1016/j.spl.2021.109246)
96. Deng W, Zhang X, Liang F, Lin G. 2019 An adaptive empirical Bayesian method for sparse deep learning. Adv. Neural Inform. Process. Syst. 32, 5563–5573.
97. Wang Y, Rocková V. 2020 Uncertainty quantification for sparse deep learning. In Int. Conf. on Artificial Intelligence and Statistics, pp. 298–308. PMLR.
98. Bai J, Song Q, Cheng G. 2020 Efficient variational inference for sparse deep learning with theoretical guarantee. Adv. Neural Inform. Process. Syst. 33, 466–476.
99. Sun Y, Song Q, Liang F. 2021 Consistent sparse deep learning: theory and computation. J. Amer. Stat. Assoc. 117, 1–15. (doi:10.1080/01621459.2021.1895175)
100. Polson NG, Ročková V. 2018 Posterior concentration for sparse deep learning. Adv. Neural Inform. Process. Syst. 31, 938–949.
101. Sun Y, Xiong W, Liang F. 2021 Sparse deep learning: a new framework immune to local traps and miscalibration. *Adv. Neural Inform. Process. Syst.* **34**, 22,301–22,312.

102. Zhang Y, Liang P, Charikar M. 2017 A hitting time analysis of stochastic gradient langevin dynamics. In *Conf. on Learning Theory*, pp. 1980–2022. PMLR.

103. Teh YW, Thiery AH, Vollmer SJ. 2016 Consistency and fluctuations for stochastic gradient Langevin dynamics. *J. Mach. Learn. Res.* **17**, 1–33.

104. Hoffman M, Ma YA. 2020 Black-box variational inference as distilled Langevin dynamics. In *Proc. of the 37th Int. Conf. on Machine Learning*, pp. 4324–4341.

105. Li L, Huang W, Gu IYH, Tian Q. 2004 Statistical modeling of complex backgrounds for foreground object detection. *IEEE Trans. Image Process.* **13**, 1459–1472. (doi:10.1109/TIP.2004.836169)

106. Wright J, Ma Y. 2022 *High-dimensional data analysis with low-dimensional models: principles, computation, and applications*. Cambridge, UK: Cambridge University Press.

107. He K, Chen X, Xie S, Li Y, Dollár P, Girshick R. 2022 Masked autoencoders are scalable vision learners. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pp. 16,000–16,009.

108. Xu H, Ding S, Zhang X, Xiong H, Tian Q. 2022 Masked autoencoders are robust data augmentors. (http://arxiv.org/abs/2206.04846)

109. Wood C. 2022 Powerful ‘machine scientists’ distill the laws of physics from raw data. *Quanta Magazine*.

110. Sener O, Koltun V. 2018 Multi-task learning as multi-objective optimization. *Adv. Neural Inform. Process. Syst.* **31**, 525–536.

111. Gao L, Fasel U, Brunton SL, Kutz JN. 2023 Convergence of uncertainty estimates in ensemble and Bayesian sparse model discovery. (http://arxiv.org/abs/2301.12649)

112. Meinshausen N, Bühlmann P. 2010 Stability selection. *J. R. Stat. Soc.: Ser. B (Statistical Methodology)* **72**, 417–473. (doi:10.1111/j.1467-9868.2010.00740.x)

113. Kingma DP, Welling M. 2013 Auto-encoding variational Bayes. (http://arxiv.org/abs/1312.6114)

114. Mars Gao L, Nathan Kutz J. 2024 Bayesian autoencoders for data-driven discovery of coordinates, governing equations and fundamental constants. GitHub repository. (https://github.com/gaoliyao/BayesianSindyAutoencoder)

115. Mars Gao L, Nathan Kutz J. 2024 Bayesian autoencoders for data-driven discovery of coordinates, governing equations and fundamental constants. Figshare. (doi:10.6084/m9.figshare.c.7090159)