# State estimation of a physical system with unknown governing equations

https://doi.org/10.1038/s41586-023-06574-8

Received: 21 February 2023

Accepted: 25 August 2023

Published online: 11 October 2023

Open access

Check for updates

Kevin Course[1] & Prasanth B. Nair[1]✉

State estimation is concerned with reconciling noisy observations of a physical system with the mathematical model believed to predict its behaviour for the purpose of inferring unmeasurable states and denoising measurable ones[1,2]. Traditional state-estimation techniques rely on strong assumptions about the form of uncertainty in mathematical models, typically that it manifests as an additive stochastic perturbation or is parametric in nature[3]. Here we present a reparametrization trick for stochastic variational inference with Markov Gaussian processes that enables an approximate Bayesian approach for state estimation in which the equations governing how the system evolves over time are partially or completely unknown. In contrast to classical state-estimation techniques, our method learns the missing terms in the mathematical model and a state estimate simultaneously from an approximate Bayesian perspective. This development enables the application of state-estimation methods to problems that have so far proved to be beyond reach. Finally, although we focus on state estimation, the advancements to stochastic variational inference made here are applicable to a broader class of problems in machine learning.

State estimation, or data assimilation as it is often referred to in the geosciences, forms the backbone of modern machinery for fusing noisy sensor data with mathematical models of complex systems in many important areas of science, engineering and finance. For example, in robotics systems, state estimation underpins control and route-planning algorithms, as well as being useful in high-level decision-making[4,5]. In finance, state estimation has been applied to estimate stochastic volatility models[6]. In medical imaging, state estimation has been advanced to improve cardiac-imaging technologies[7]. In meteorology, state estimation is used to reconstruct weather in the past and make predictions about weather in the future[8,9].

The field of state estimation can arguably trace its roots back to the work of the early nineteenth-century astronomers. In 1801, the then 24-year-old Gauss devised what could be called the first state-estimation algorithm. Using only Kepler's laws along with ordinary least squares, Gauss accurately computed the orbit of the asteroid Ceres from limited, noisy observations[1,10]. Our more modern conception of the field originates in the foundational works of Kalman and Bucy in the early 1960s. The researchers at the Research Institute for Advanced Studies, building on the work of Wiener and Kolmogorov in the 1940s, published their now well-known filtering algorithms for linear systems in 1960 and 1961[11,12]. As a testament to the success of their work, it was the Kalman filter—extended by Schmidt's group at the NASA's Ames Research Center to nonlinear systems—that solved the Apollo guidance and navigation problem[13]. Guided by a revolution in computing power and in the types of sensor available, such as the Global Positioning System, laser imaging and digital cameras[1], modern developments in state estimation have focused on nonlinear systems[14], high-dimensional systems[15,16] and deriving more accurate approximations[5].

An essential prerequisite for state estimation is a mathematical model describing how the system evolves over time. It is well accepted that some degree of uncertainty is inevitable in the mathematical characterization of any real-world system. Standard state-estimation algorithms assume that uncertainty in the governing equations manifests as an additive stochastic perturbation[1] or is parametric in nature[3,17–20]. In reality, the true nature of uncertainty in mathematical models is much more diverse.

A more complete picture of uncertainty in mathematical models shows that it often also enters as a by-product of modelling errors[21]. Modelling errors are pernicious because they often arise owing to imperfect knowledge of the underlying system in combination with the need to simplify for computational considerations. For example, meteorology often uses empirical assumptions to approximate sub-gridscale processes[22]. Such modelling errors can be expressed mathematically as uncertainty in the structural form of the equations governing how the state vector evolves over time. Existing methods for state estimation cannot account for model-structure uncertainty. Worse still, not accounting for such uncertainty will introduce bias in estimates for the state. Although there has been some work in model-free state estimation to address such realities[4,23,24], these approaches lose the inherent interpretability of handcrafted models.

We introduce a method for handling model-structure uncertainty in a manner that recovers the interpretability of handcrafted models. We do so by learning the motion model in the form of a set of symbolic differential equations simultaneously to a state estimate. This modelling choice enables state estimation in situations in which there are substantial modelling errors or the underlying dynamics are partially or completely unknown. This advancement is made possible by a new

[1] Institute for Aerospace Studies, University of Toronto, Toronto, Ontario, Canada. ✉e-mail: prasanth.nair@utoronto.ca

reparametrization trick for Markov Gaussian processes that we introduce in this work.

Numerical studies are presented for a range of test cases to illustrate the performance of the proposed approach. In practical situations in which modelling errors are present (even small errors), our method outperforms standard state-estimation techniques. Furthermore, our approach for state estimation with unknown governing equations allows analysts to discover missing terms in (or the entirety of) the governing equations using indirect observations. In Methods, we show that our approach outperforms state-of-the-art algorithms for governing-equation discovery, particularly in the low-data and high-noise regimes, and can be used in situations in which the entire state is not measurable. Finally, in cases in which interpretable forward models are not required, we show that our approach can be used to infer neural stochastic differential equations (SDEs) without relying on a forward solver in training.

## Results

### Problem statement

To mathematically define our problem statement, consider a noisy observation process of the form

$$y_t = g(x_t, t) + \eta, \tag{1}$$

in which $y_t \in \mathbb{R}^d$ is the observation vector at time instant $t$, $x_t \in \mathbb{R}^D$ is the latent state vector, $g: \mathbb{R}^D \times \mathbb{R} \to \mathbb{R}^d$ is the observation function and $\eta \in \mathbb{R}^d$ denotes observation noise.

Given the time-series dataset, $\mathcal{D} = \{(t_i, y_{t_i})\}_{i=1}^{N}$, in which $t_i \in [0, T]$ is the timestamp associated with the observation $y_{t_i} \in \mathbb{R}^d$, our goal is to infer

1. A well-calibrated estimate of the state vector, $x_t$, over the time window $[0, T]$, and
2. The equations governing the temporal evolution of the state.

### Related work

This work lies at the intersection between three areas of science: (1) state estimation; (2) discovery of governing equations from measurements; and (3) Bayesian inference over functions. We summarize some relevant work in these areas below.

The classical state-estimation problem in which the governing equations are known has been widely studied in the literature; see, for example, the texts by Barfoot[1] and Särkkä and Svensson[2]. There is also a wide body of literature focusing on simultaneous state and parameter estimation; see, for example, refs. 3,17–20. These approaches are useful for problems in which the structure of the governing equations is known and the uncertainties are strictly parametric. Although model-free state-estimation techniques can be used in situations in which dynamics are unknown[4,23–25], these techniques lose the interpretability of handcrafted models. A comparison of the assumptions made here to those made by existing methods is provided in Extended Data Table 1.

In terms of inferring interpretable governing equations, the special case in which all the states are measurable (that is, the observation function in equation (1) is the identity) has been extensively studied; see, for example, refs. 26–28. In 2011, Wang et al.[26] showed how governing equations could be identified from full state measurements with a compressive-sensing approach using a dictionary of polynomials. Brunton et al.[27] studied the problem of discovering governing equations from full state measurements using a compressive-sensing approach and demonstrated that this is a very powerful tool for solving a wide class of problems in computational science and engineering. In principle, this approach can be used with incomplete measurements using the notion of time-delay embedding[26]. However, it is challenging to get this approach to work well in practice owing to numerical issues; see appendix 4.5 of ref. 27.

The focus of this work is on the practical problem of estimating the state and the governing equations in which only indirect measurements are available.

### Stochastic variational inference

Here we tackle the problem of state estimation with unknown governing equations using the machinery of stochastic variational inference (SVI)[29]. SVI is a powerful approach for approximate Bayesian inference that converts the problem of generating samples from an unnormalized density function into an optimization problem involving maximization of a stochastic approximation of the evidence lower bound (ELBO). Variational inference has been applied to solve a wide variety of problems in machine learning[30–32].

SVI typically requires four ingredients: (1) priors over variables of interest; (2) a parametrized approximate posterior; (3) a tractable expression for the ELBO that admits unbiased stochastic approximations; and (4) a method for approximating gradients of expectations in the ELBO.

In the context of the problem considered here, SVI cannot be applied in a straightforward manner because we seek to estimate a function, $x_t$, given measurements of $y_t$, rather than a finite-dimensional vector. Here we propose a reparametrization trick for approximating expectations under Markov Gaussian processes that enables us to tackle this challenge. A high-level summary of our approach is provided in Fig. 1 and we introduce the core ingredients of our methodology in the sections that follow.

### Priors

We place a hierarchical prior over the state in the form of an Itô SDE with an uncertain drift and diffusion function

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \Sigma_\theta(t)\mathrm{d}\beta_t, \tag{2}$$

in which $f_\theta: \mathbb{R}^D \times \mathbb{R} \to \mathbb{R}^D$ is the drift function, $\Sigma_\theta(t): \mathbb{R} \to \mathbb{R}^{D\times D}$ is the dispersion matrix, $\beta_t \in \mathbb{R}^D$ indicates Brownian motion with diffusion matrix $Q$ and $\theta \in \mathbb{R}^{M+D}$ is a vector of parameters. The prior for the state captures our assumption that the structural form of the governing equations is partially or completely unknown.

To learn interpretable dynamics, it is often convenient to make the assumption that the drift function can be approximated by a sparse, linear combination of known basis functions

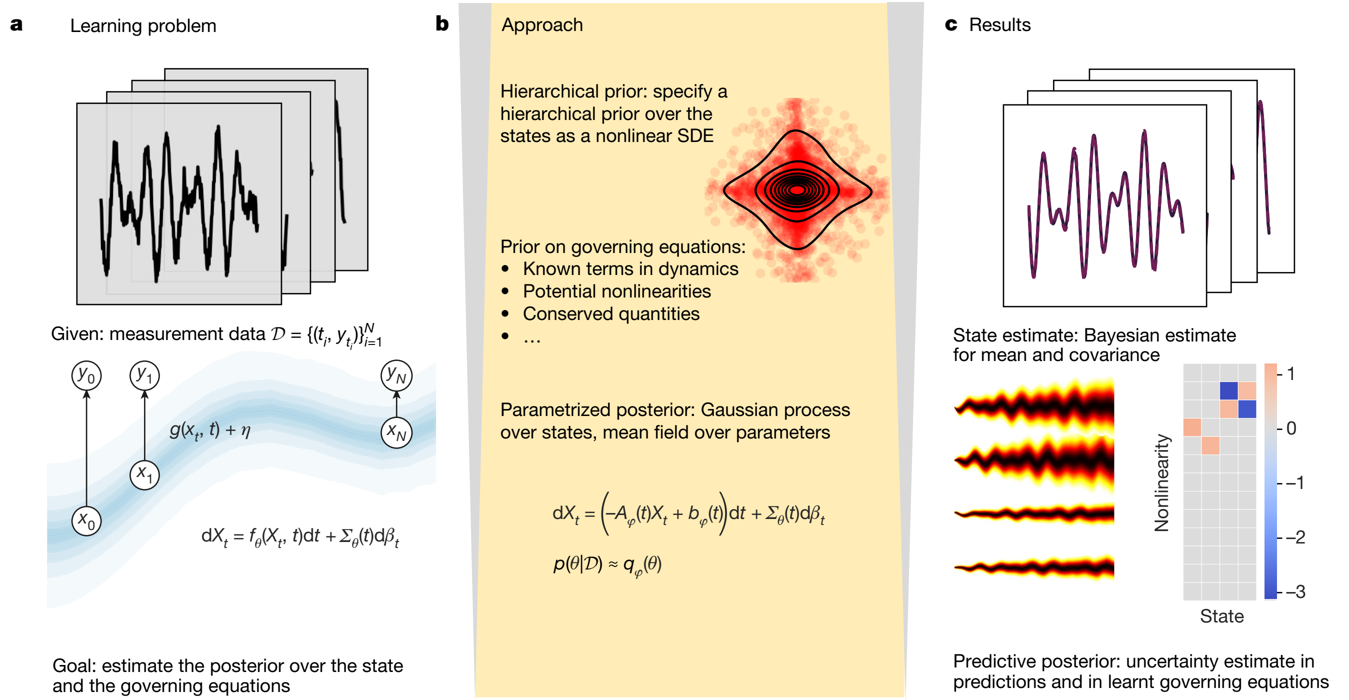
$$f_\theta(x_t, t) = f_\varnothing(x_t, t) + \sum_{i=1}^{M} \psi_i(x_t, t)\theta_i, \tag{3}$$

in which $f_\varnothing: \mathbb{R}^D \times \mathbb{R} \to \mathbb{R}^D$ indicates the drift function of the known dynamics and $\psi_i: \mathbb{R}^D \times \mathbb{R} \to \mathbb{R}^D$ is the $i$th function from a predefined dictionary of basis functions. Often, this dictionary may include polynomials, trigonometric functions etc. We parametrize the dispersion matrix as $\Sigma_\theta = \Sigma_\varnothing + \mathrm{diag}(\theta_{M+1}^2, \theta_{M+2}^2, \ldots, \theta_{M+D}^2)$, in which $\Sigma_\varnothing \in \mathbb{R}^{D\times D}$ refers to the known terms in the dispersion matrix.

As an aside, it is worth noting that it is straightforward to use more general parametrizations of the drift function, $f_\theta$. For instance, if we are interested in inferring Hamiltonians or more general energy cycles, we could parametrize $f_\theta$ by a Hamiltonian[33] or generalized Hamiltonian[34] neural network, respectively. Later, we provide examples to demonstrate the case when $f_\theta$ is parametrized by a fully connected neural network.

In alignment with the assumption of a sparse linear combination of basis functions and to indicate our preference that perturbations to the dispersion matrix be small or sparse if possible, we will make use of a sparsity-inducing horseshoe prior over the parameters, $\theta \sim p(\theta)$ (ref. 35).

We have now specified a hierarchical prior over the latent process we wish to infer, $X|\theta$ and $p(\theta)$. Here $X|\theta$ indicates the process defined by the SDE in equation (2) over sample paths in the interval $[0, T]$ conditioned



a
Learning problem
Given: measurement data $\mathcal{D} = \{(t_i, y_{t_i})\}_{i=1}^N$
$y_0$
$y_1$
$y_N$
$g(x_t, t) + \eta$
$x_1$
$x_N$
$x_0$
$dX_t = f_\theta(X_t, t)dt + \Sigma_\theta(t)d\beta_t$
Goal: estimate the posterior over the state and the governing equations
b
Approach
Hierarchical prior: specify a hierarchical prior over the states as a nonlinear SDE
Prior on governing equations:
• Known terms in dynamics
• Potential nonlinearities
• Conserved quantities
• …
Parametrized posterior: Gaussian process over states, mean field over parameters
$dX_t = \left(-A_\varphi(t)X_t + b_\varphi(t)\right)dt + \Sigma_\theta(t)d\beta_t$
$p(\theta|\mathcal{D}) \approx q_\varphi(\theta)$
c
Results
State estimate: Bayesian estimate for mean and covariance
Nonlinearity
State
1
0
−1
−2
−3
Predictive posterior: uncertainty estimate in predictions and in learnt governing equations

**Fig. 1 | Main ideas. a**, Given noisy time-series data and an observation model, our goal is to estimate the underlying state and any missing terms in the forward model. **b**, We specify a hierarchical prior over the state and the governing equations and then infer an approximate posterior over the state and forward model using SVI. **c**, We arrive with a state estimate, a method for generating forecasts with uncertainty estimates and a Bayesian estimate for missing terms in the governing equations.

on a particular setting of the parameters. State estimation amounts to inferring the posterior process over $X$ and inferring missing terms in the governing equations amounts to inferring the posterior over $\theta$. For the special case in which the governing equations are known, the assumptions made in this work coincide with those made in continuous time-state estimation[1,2].

### Variational distributions

To carry out SVI, we need to construct parametrized variational distributions for the state and the parameters in equations (2) and (3). We approximate the posterior process over the state as the solution to a linear SDE of the form

$$dX_t = (-A_\varphi(t)X_t + b_\varphi(t))dt + \Sigma_\theta d\beta_t, \tag{4}$$

in which $A_\varphi : \mathbb{R} \to \mathbb{R}^{D\times D}$ and $b_\varphi : \mathbb{R} \to \mathbb{R}^D$ are symmetric-matrix and vector-valued functions of time, respectively, $\varphi$ is a vector of variational parameters, $\Sigma_\theta d\beta_t$ is the same diffusion process defined in the prior and the initial condition is assumed to be Gaussian, $X_0 \sim \mathcal{N}(m_0, S_0)$.

Because the solution of a linear SDE is a Markov Gaussian process, we can parametrize the marginal statistics of the solution of equation (4) as $q_\varphi(x_t) = \mathcal{N}(m_\varphi(t), S_\varphi(t))$, in which $m_\varphi : \mathbb{R} \to \mathbb{R}^D$ is the mean and $S_\varphi : \mathbb{R} \to \mathbb{R}^{D\times D}$ is the (marginal) covariance[36]. The marginal statistics satisfy the following system of ordinary differential equations (ODEs):

$$\dot{m}_\varphi(t) = -A_\varphi(t)m_\varphi(t) + b_\varphi(t) \tag{5}$$

$$\dot{S}_\varphi(t) = -A_\varphi(t)S_\varphi(t) - S_\varphi(t)A_\varphi(t)^T + \Sigma_\theta Q \Sigma_\theta^T \tag{6}$$

in which $m_\varphi(0) = m_0$ and $S_\varphi(0) = S_0$.

To proceed further, we also need to define an approximate posterior for the parameters, $\theta$, defined in equation (3). We used the log-normal parametrization from ref. 37, which we compactly denote by $q_\varphi(\theta)$, in which again $\varphi$ denotes the vector of variational parameters; see Methods for more details.

### ELBO

In SVI, the variational parameters are estimated by maximizing the ELBO, which is equivalent to minimizing the Kullback–Leibler (KL) divergence between the approximate posterior and the true, intractable posterior[29]. Here we propose a new reparametrization strategy that enables the variational parameters to be efficiently estimated without using a forward solver.

Letting $\mathbb{E}_{\tilde{P}|\theta}[\cdot]$ indicate expectations under the prior SDE in equation (2), we can derive the following ELBO[38–40]

$$\log p(\mathcal{D}) = \log \mathbb{E}_{p(\theta)}\left[\mathbb{E}_{\tilde{P}|\theta}\left[\prod_{i=1}^N p(y_{t_i}|x_{t_i})\right]\right] \tag{7}$$

$$\begin{aligned} &\geq \sum_{i=1}^N \mathbb{E}_{q_\varphi(x_{t_i})}[\log p(y_{t_i}|x_{t_i})] \\ &-\frac{1}{2}\int_0^T \mathbb{E}_{q_\varphi(x_t)q_\varphi(\theta)}\left[\|r(x_t, t, \theta, \varphi)\|^2_{\Sigma_\theta Q \Sigma_\theta^T}\right]dt \\ &-D_{\mathrm{KL}}(q_\varphi(\theta)\|p(\theta)) = \mathrm{ELBO}(\varphi), \end{aligned} \tag{8}$$

in which $r(x_t, \theta, \varphi) = -A_\varphi(t)x_t + b_\varphi(t) - f_\theta(x_t, t)$ is the drift residual, $||v||^2_{\Sigma_\theta Q \Sigma_\theta^T} = v^T(\Sigma_\theta Q \Sigma_\theta^T)^{-1}v$ and $D_{\mathrm{KL}}(q_\varphi(\theta)||p(\theta))$ indicates the KL divergence between $q_\varphi(\theta)$ and $p(\theta)$.

Maximizing equation (8) with respect to the parameters of the variational distributions, $\varphi$, would provide an approximate state estimate, $q_\varphi(x_t)$, and an estimate for the posterior distribution over the parameters, $q_\varphi(\theta)$. Unfortunately, maximizing this ELBO is computationally challenging because the first two terms depend on expectations with respect to $q_\varphi(x_t)$, the current state estimate. Archambeau et al.[38,39] explored maximizing equation (8) subject to the differential equality constraints in equations (5) and (6) in the context of dynamic data assimilation. This approach requires solving $2(D + D^2)$ ODEs at each optimization iteration. Solving ODEs as part of an optimization procedure is computationally challenging because: (1) gradient-based updates to the variational parameters can cause the ODEs to become extremely stiff mid-optimization, causing the computational cost of an ODE solve to explode, and (2) ODE solvers are inherently iterative sequential methods, making them poorly suited to modern parallel-computing hardware.

Here we introduce a reparametrization for expectations in equation (8) with respect to $q_\varphi(x_t)$ that replaces the need for an ODE forward solver with a stochastic approximation that can be evaluated in an embarrassingly parallel fashion. This eliminates the practical challenges associated with maximizing equation (8). Our main theoretical result takes the form

$$\begin{aligned}\mathrm{ELBO}(\varphi) = &\sum_{i=1}^{N} \mathbb{E}_{\mathcal{N}(m_\varphi(t_i), S_\varphi(t_i))}[\log p(y_{t_i}|x_{t_i})]\\ &-\frac{1}{2}\int_0^T \mathbb{E}_{\mathcal{N}(m_\varphi(t), S_\varphi(t))q_\varphi(\theta)}\left[\|r(x_t, t, \theta, \varphi)\|^2_{\Sigma_\theta Q \Sigma_\theta^T}\right]\mathrm{d}t\\ &-D_{\mathrm{KL}}(q_\varphi(\theta)\|p(\theta)),\end{aligned} \tag{9}$$

in which

$$\begin{aligned}r(x_t, t, \theta, \varphi) = \mathrm{vec}^{-1}((S_\varphi(t) \oplus S_\varphi(t))^{-1}\mathrm{vec}(\Sigma_\theta Q \Sigma_\theta^T - \dot{S}_\varphi(t)))\\ (m_\varphi(t) - x_t) + \dot{m}_\varphi(t) - f_\theta(x_t, t),\end{aligned} \tag{10}$$

denotes the drift residual reparametrized in terms of $m_\varphi$ and $S_\varphi$ (for a derivation, see Methods), $\oplus$ indicates the Kronecker sum, $\mathrm{vec}: \mathbb{R}^{D\times D} \to \mathbb{R}^{D^2}$ maps a matrix into a vector by stacking columns and $\mathrm{vec}^{-1}: \mathbb{R}^{D^2} \to \mathbb{R}^{D\times D}$ unstacks a vector such that $\mathrm{vec}^{-1}(\mathrm{vec}(C)) = C \;\forall\; C \in \mathbb{R}^{D\times D}$.

Recall that, in the original ELBO (equation (8)), the first two terms contained expectations with respect to the marginal statistics of the solution of the SDE (equation (4)), $q_\varphi(x_t)$, and approximating these expectations required the use of a forward solver. In this reparametrized ELBO (equation (9)), expectations are taken with respect to the Gaussian, $\mathcal{N}(m_\varphi(t), S_\varphi(t))$ . It is worth noting that, as we directly parametrize $m_\varphi$ and $S_\varphi$ instead of $A_\varphi$ and $b_\varphi$, it is feasible to estimate all expectations without running a forward ODE solver. Notably, we no longer need to solve any ODEs; we only need to construct stochastic approximations of the terms in the ELBO. Such stochastic approximations are easy to compute using modern parallel-computing hardware. It is this new development that enables us to perform approximate Bayesian inference over both the state and the unknown governing equations.

In Methods, we describe how to parametrize $m_\varphi$ and $S_\varphi$ and how to optimize a stochastic approximation of the ELBO (equation (9)). Having maximized the ELBO, we are left with an approximation to the posterior over the state, $q_\varphi(x_t) = \mathcal{N}(m_\varphi(t), S_\varphi(t))$, and an approximate posterior over the parameters defining the governing equations, $q_\varphi(\theta)$, that can be used to make probabilistic forecasts and perform further analysis. We henceforth refer to our method as SVI for state estimation (SVISE). Our method was implemented in PyTorch[41]. In the following sections, we provide some examples of SVISE applied to problems in state estimation and governing-equation discovery.

### Example 1: state estimation with known motion model

In this section, we compare our method for state estimation with unknown governing equations to the particle filter (PF)[42,43] when the form of the underlying motion model is known exactly. The PF was chosen as the baseline for comparison as it is a fully Bayesian method capable of handling highly nonlinear systems. Both methods were provided with complete knowledge of the underlying dynamics and observation model. We performed this comparison on six benchmark problems and a detailed description of the experiment design is provided in Methods.

The results are summarized in Fig. 2a. We found that our method provided results comparable with the PF in terms of providing an estimate for the mean of the state. We believe that minor differences in performance are because of our choice of basis functions. Clearly there is an opportunity to perform model selection in terms of the basis-function design if desired.

### Example 2: state estimation with modelling errors

In this section, we examine the performance of our method on the same suite of benchmarks when a small modelling error is introduced. We introduce a small state-dependent corruption to the governing equations and compare the performance of SVISE with the PF. Unlike the previous set of examples, we do not provide our method with access to knowledge of the dynamics (that is, we set $f_\varnothing = 0$ and $\Sigma_\varnothing = 0$), so that we are required to learn the dynamics as well as a state estimate. The PF is given access to the uncorrupted dynamics, along with the exact form of the diffusion process.

The specific form of the corruption is chosen so that the form of the governing equations are, on average, the same as in the previous section and so that any particular realization of a corruption could plausibly be mistaken for further process noise rather than model misspecification; see Fig. 2c–e. Further details can be found in Methods.

Because the PF is given the exact form of the diffusion process, we believe that upcoming results represent an overestimate for the PF performance. Despite this advantage, we find that our method now, on average, outperforms the PF in the presence of this mild corruption to the dynamics; see Fig. 2b. As discussed previously, there are many real-world systems for which modelling errors are an unavoidable reality. This work can be used in such circumstances to potentially improve on state-estimation performance.

As was previously discussed, a key differentiating feature of our method compared with model-free state-estimation techniques is that we are learning symbolic differential equations for the motion model. Looking to Fig. 2f,g,h, we see that our approach can be used to make probabilistic forecasts after state estimation has concluded.

### Example 3: governing-equation discovery with neural SDEs

When the state is low dimensional, or the state is spatially extended, it is possible to make use of dictionaries of polynomial basis functions to infer symbolic SDEs simultaneously to a state estimate. In Methods, we provide three numerical studies demonstrating the application of our approach to symbolic-governing-equation discovery. We show that our approach offers superior performance to state-of-the-art algorithms for the sparse identification of nonlinear dynamics in the presence of substantial noise or a lack of data. Also, we demonstrate scalability by showing that our approach can be used to infer symbolic governing equations for a spatially extended system with 1,024 states. Finally, we show that our approach can be used to infer symbolic governing equations for second-order systems using only position measurements.

Unfortunately, dictionaries of polynomial basis functions become prohibitively large even for moderately high-dimensional systems. In such situations, a reasonable alternative to inferring symbolic differential equations is to parametrize the drift function by a neural network. In Methods, we present numerical studies on a binary-black-hole problem that involves inferring a neural SDE using a nonlinear observation function. In the example that follows, we consider a fluid-dynamics problem with a high-dimensional state space to illustrate how our approach can be combined with off-the-shelf dimensionality-reduction algorithms to infer probabilistic reduced-order models (ROMs) in the format of latent neural SDEs. The main idea of ROMs is to infer a mapping that enables the original high-dimensional system to be transformed into a

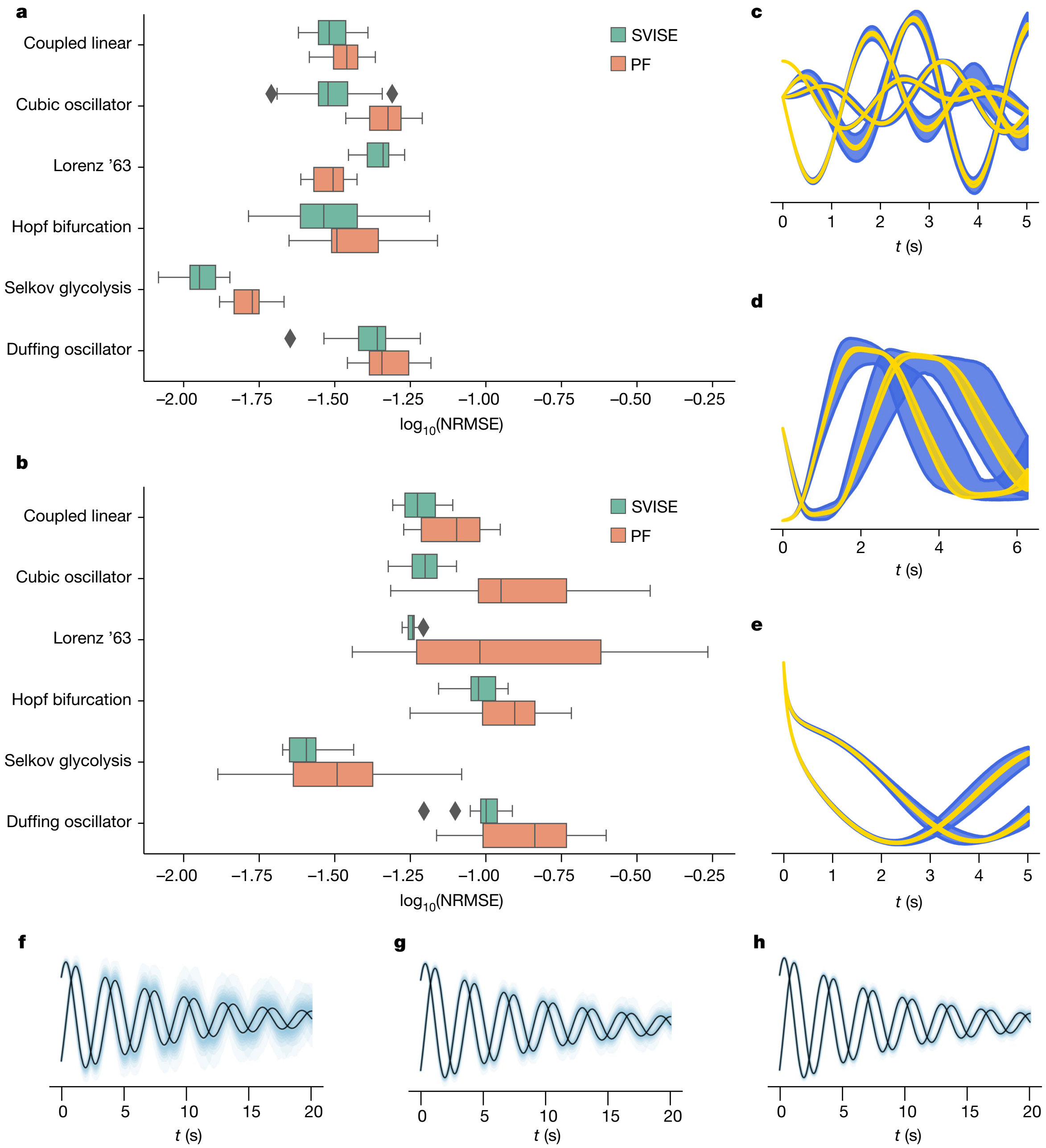


**Fig. 2 | Benchmarking summary with 20 independent trials per system.** **a**, State-estimation results with perfect knowledge of the dynamics. **b**, State-estimation results with minor corruption of the dynamics. **c**–**e**, Before and after adding a small probabilistic corruption for the coupled linear oscillator (**c**), the cubic oscillator (**d**) and the Hopf bifurcation (**e**) (10–90th quantiles shown). **f**–**h**, Probabilistic forecast with 64 (**f**), 512 (**g**) and 1,024 (**h**) data points. We see that our method (SVISE) performs similarly to the PF when the governing equations are known exactly. In the presence of only mild modelling error, our method outperforms the PF on average. Moreover, after training, our method enables us to make probabilistic forecasts. Expectedly, the predictive posterior shrinks as the amount of data is increased.

low-dimensional system that is computationally cheaper to solve[44]. The approach we take is aligned with previous works in the ROMs literature that infer lower-dimensional differential equations on a predefined manifold[45,46].

As a case study, we considered the challenging problem of inferring latent differential equations for fluid flow past a circular cylinder with a Reynolds number of 2,000. We model this problem using a spatial discretization of the two-dimensional incompressible Navier–Stokes equation with 596,602 states. We train a neural SDE on 38 latent states. These latent states were constructed by projecting the data onto the top 38 proper orthogonal decomposition (POD) modes to capture 90% of the total variance. More details on the numerical study design are provided in Methods. It can be seen from Fig. 3 that our approach provides accurate probabilistic predictions of the flow field.

## Discussion

We have presented a method for state estimation that enables the treatment of real-world problems in which there are notable modelling errors or the underlying governing equations are completely or partially unknown. This is an important class of problems that has thus far defied a computationally tractable, statistically rigorous solution. We have also provided results for governing-equation discovery for a variety of challenging problems. The results indicate that our approach

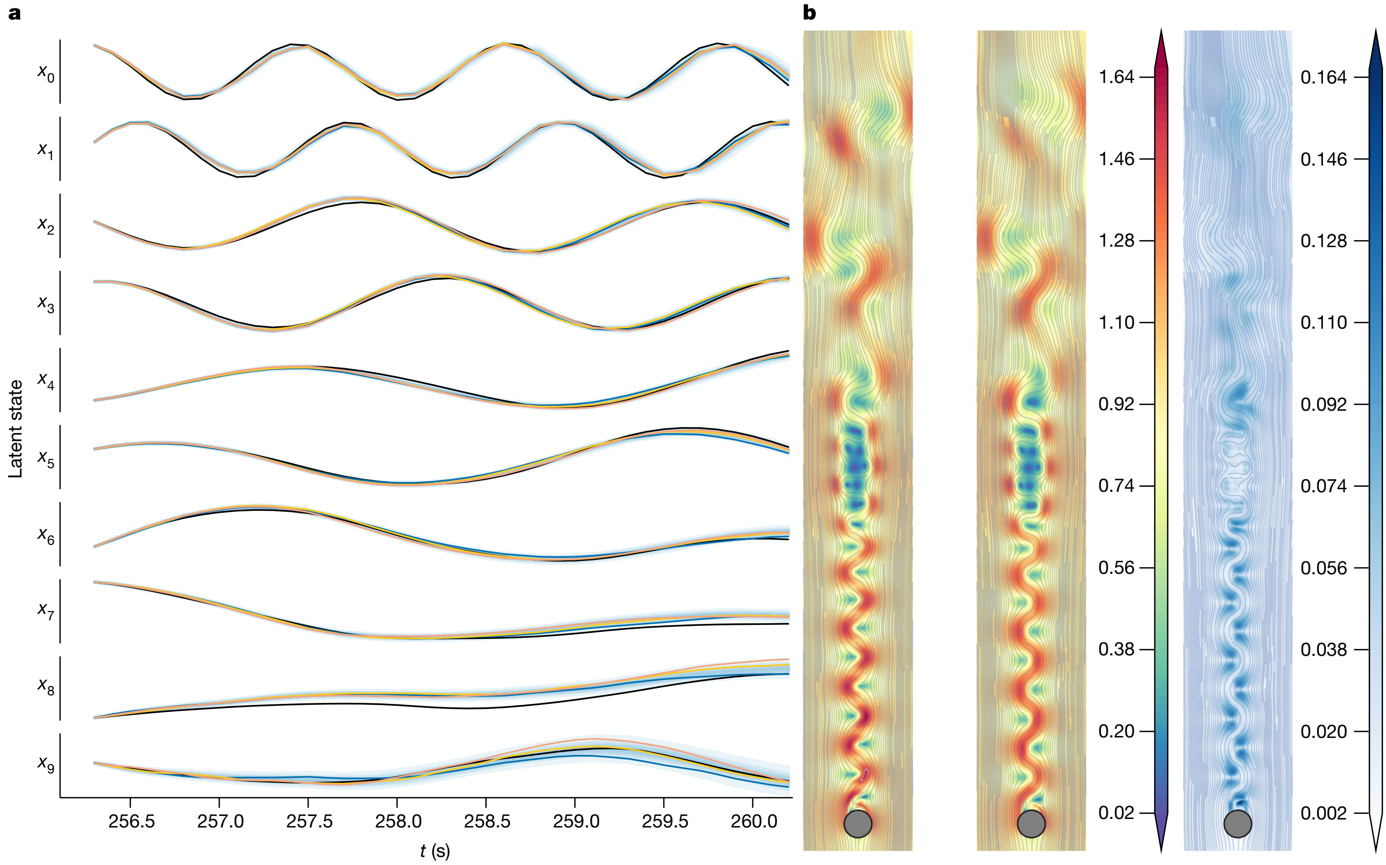


**Fig. 3 | Flow behind a cylinder ROM problem. a**, Predictions for the first ten latent states over the testing time interval $t$ = [256.2, 260.2]. The black lines show the test states and the coloured lines indicate samples from the predictive posterior. **b**, Velocity magnitude and flow lines at $t$ = 260.2. Test (left), mean prediction (centre) and standard deviation (right). In this example, we train a neural SDE on trajectories projected onto the POD basis to construct a ROM. Although a neural SDE is less interpretable than a symbolic model, it is useful in cases in which the state is either (1) high dimensional or (2) it is not clear which basis functions might be appropriate for the problem at hand. We see that the error bars are higher in regions in which the mean prediction seems to differ from the test velocity.

can outperform state-of-the-art algorithms for governing-equation discovery often by more than an order of magnitude; see Methods for details. It is worth reiterating that many governing-equation learning algorithms only provide point parameter estimates[27]. As demonstrated in Fig. 2f–h, our method enables Bayesian governing-equation identification, thereby allowing for probabilistic forecasts to be made.

Moreover, in cases in which specifying a dictionary of basis functions is not feasible (such as when the state space is high dimensional or it is not clear what basis functions are appropriate), we have shown that it is possible to infer the missing dynamics in the form of a neural network. It is worth noting here that our approach enables neural SDEs to be trained without using a computationally expensive forward solver and adjoint sensitivity calculations.

These results are made possible by a reparametrization trick for Markov Gaussian processes that we introduce here. We believe that there is substantial room for future work in applying this technique for performing variational inference over stochastic processes outside the state-estimation domain. Potential application areas include multiphysics ROM problems, state estimation with unknown observation models, stochastic closure modelling[47] and inferring latent forcing models[48]. Also, there is room for using more expressive approximations to the posterior over the parameters to better approximate uncertainty. Finally, there has been a recent resurgence of interest in using SDEs for building complex generative models[40] and we expect that our approach could be used to gain deeper insights into the statistical underpinnings of such models.

## Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41586-023-06574-8.

1. Barfoot, T. D. *State Estimation for Robotics* (Cambridge Univ. Press, 2017).
2. Särkkä, S. & Svensson, L. *Bayesian Filtering and Smoothing* 2nd edn (Cambridge Univ. Press, 2023).
3. Pulido, M., Tandeo, P., Bocquet, M., Carrassi, A. & Lucini, M. Stochastic parameterization identification using ensemble Kalman filtering combined with maximum likelihood methods. *Tellus A* **70**, 1–17 (2018).
4. Guo, Z. C., Korotkine, V., Forbes, J. R. & Barfoot, T. D. Koopman linearization for data-driven batch state estimation of control-affine systems. *IEEE Robot. Autom. Lett.* **7**, 866–873 (2021).
5. Barfoot, T. D., Forbes, J. R. & Yoon, D. J. Exactly sparse Gaussian variational inference with application to derivative-free batch nonlinear state estimation. *Int. J. Robot. Res.* **39**, 1473–1502 (2020).
6. Harvey, A., Ruiz, E. & Shephard, N. Multivariate stochastic variance models. *Rev. Econ. Stud.* **61**, 247–264 (1994).
7. Keil, A. *Dynamic Variational Level Sets for Cardiac 4D Reconstruction*. PhD thesis, Technische Univ. München (2010).
8. Houtekamer, P. L. & Zhang, F. Review of the ensemble Kalman filter for atmospheric data assimilation. *Mon. Weather Rev.* **144**, 4489–4532 (2016).
9. Carrassi, A., Bocquet, M., Bertino, L. & Evensen, G. Data assimilation in the geosciences: an overview of methods, issues, and perspectives. *Wiley Interdiscip. Rev. Clim. Change* **9**, e535 (2018).
10. Teets, D. & Whitehead, K. The discovery of Ceres: how Gauss became famous. *Math. Mag.* **72**, 83–93 (1999).
11. Kalman, R. E. A new approach to linear filtering and prediction problems. *J. Basic Eng.* **82**, 35–45 (1960).

12. Kalman, R. E. & Bucy, R. S. New results in linear filtering and prediction theory. *J. Basic Eng.* **83**, 95–108 (1961).
13. Grewal, M. S. & Andrews, A. P. Applications of Kalman filtering in aerospace 1960 to the present [historical perspectives]. *IEEE Control Syst. Mag.* **30**, 69–78 (2010).
14. Julier, S. J. & Uhlmann, J. K. New extension of the Kalman filter to nonlinear systems. *Proc. SPIE* **3068**, 182–193 (1997).
15. Evensen, G. *Data Assimilation: The Ensemble Kalman Filter* 2nd edn (Springer, 2009).
16. Katzfuss, M., Stroud, J. R. & Wikle, C. K. Understanding the ensemble Kalman filter. *Am. Stat.* **70**, 350–357 (2016).
17. Calderhead, B., Girolami, M. & Lawrence, N. D. Accelerating Bayesian inference over nonlinear differential equations with Gaussian processes. In *Proc. Advances in Neural Information Processing Systems* Vol. 21 (eds Koller, D. et al.) 217–224 (Curran Associates, 2009).
18. Dondelinger, F., Husmeier, D., Rogers, S. & Filippone, M. ODE parameter inference using adaptive gradient matching with Gaussian processes. In *Proc. Sixteenth International Conference on Artificial Intelligence and Statistics* Vol. 31 (eds Carvalho, C. M. & Ravikumar, P.) 216–228 (PMLR, 2013).
19. Gorbach, N. S., Bauer, S. & Buhmann, J. M. Scalable variational inference for dynamical systems. In *Proc. Advances in Neural Information Processing Systems* Vol. 30 (eds Guyon, I. et al.) 4806–4815 (Curran Associates, 2017).
20. Wenk, P. et al. Fast Gaussian process based gradient matching for parameter identification in systems of nonlinear ODEs. In *Proc. Twenty-Second International Conference on Artificial Intelligence and Statistics* Vol. 89 (eds Chaudhuri, K. & Sugiyama, M.) 1351–1360 (PMLR, 2019).
21. Glimm, J. & Sharp, D. in *Stochastic Partial Differential Equations: Six Perspectives* Mathematical Surveys and Monographs 64 (eds Carmona, R. A. & Rozovskii, B.) 3–44 (American Mathematical Society, 1997).
22. Slingo, J. & Palmer, T. Uncertainty in weather and climate prediction. *Phil. Trans. R. Soc. A* **369**, 4751–4767 (2011).
23. Fukumizu, K., Song, L. & Gretton, A. Kernel Bayes' rule: Bayesian inference with positive definite kernels. *J. Mach. Learn. Res.* **14**, 3753–3783 (2013).
24. Nishiyama, Y., Afsharinejad, A., Naruse, S., Boots, B. & Song, L. The nonparametric kernel Bayes smoother. In *Proc. 19th International Conference on Artificial Intelligence and Statistics* Vol. 51 (eds Gretton, A. & Robert, C. C.) 547–555 (PMLR, 2016).
25. Ghahramani, Z. & Hinton, G. E. *Parameter Estimation for Linear Dynamical Systems* Technical Report CRG-TR-92-2 (Univ. Toronto, 1996).
26. Wang, W.-X., Yang, R., Lai, Y.-C., Kovanis, V. & Grebogi, C. Predicting catastrophes in nonlinear dynamical systems by compressive sensing. *Phys. Rev. Lett.* **106**, 154101 (2011).
27. Brunton, S. L., Proctor, J. L. & Kutz, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl Acad. Sci. USA* **113**, 3932–3937 (2016).
28. Mangan, N. M., Brunton, S. L., Proctor, J. L. & Kutz, J. N. Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Trans. Mol. Biol. Multi-Scale Commun.* **2**, 52–63 (2016).
29. Hoffman, M. D., Blei, D. M., Wang, C. & Paisley, J. Stochastic variational inference. *J. Mach. Learn. Res.* **14**, 1303–1347 (2013).
30. Kingma, D. P. & Welling, M. Auto-encoding variational Bayes. In *Proc. 2nd International Conference on Learning Representations* (2014).
31. Rezende, D. & Mohamed, S. Variational inference with normalizing flows. In *Proc. 32nd International Conference on Machine Learning* (eds Bach, F. & Blei, D.) 1530–1538 (PMLR, 2015).
32. Ho, J., Jain, A. & Abbeel, P. Denoising diffusion probabilistic models. In *Proc. Advances in Neural Information Processing Systems* Vol. 33 (eds Larochelle, H. et al.) 6840–6851 (Curran Associates, 2020).
33. Greydanus, S., Dzamba, M. & Yosinski, J. Hamiltonian neural networks. In *Proc. Advances in Neural Information Processing Systems* Vol. 32 (eds Wallach, H. et al.) 15353–15363 (Curran Associates, 2019).
34. Course, K., Evans, T. & Nair, P. B. Weak form generalized Hamiltonian learning. In *Proc. Advances in Neural Information Processing Systems* Vol. 33 (eds Larochelle, H. et al.) 18716–18726 (Curran Associates, 2020).
35. Carvalho, C. M., Polson, N. G. & Scott, J. G. Handling sparsity via the horseshoe. In *Proc. Twelth International Conference on Artificial Intelligence and Statistics* (eds van Dyk, D. & Welling, M.) 73–80 (PMLR, 2009).
36. Särkkä, S. & Solin, A. *Applied Stochastic Differential Equations* Vol. 10 (Cambridge Univ. Press, 2019).
37. Louizos, C., Ullrich, K. & Welling, M. Bayesian compression for deep learning. In *Proc. Advances in Neural Information Processing Systems* Vol. 30 (eds Guyon, I. et al.) (Curran Associates, 2017).
38. Archambeau, C., Cornford, D., Opper, M. & Shawe-Taylor, J. Gaussian process approximations of stochastic differential equations. In *Proc. Gaussian Processes in Practice* 1–16 (PMLR, 2007).
39. Archambeau, C., Opper, M., Shen, Y., Cornford, D. & Shawe-Taylor, J. Variational inference for diffusion processes. In *Proc. Advances in Neural Information Processing Systems* Vol. 20 (Curran Associates, 2007).
40. Li, X., Wong, T.-K. L., Chen, R. T. Q. & Duvenaud, D. Scalable gradients for stochastic differential equations. In *Proc. Twenty Third International Conference on Artificial Intelligence and Statistics* Vol. 108 (eds Chiappa, S. & Calandra, R.) 3870–3882 (PMLR, 2020).
41. Paszke, A. et al. PyTorch: an imperative style, high-performance deep learning library. In *Proc. Advances in Neural Information Processing Systems* Vol. 32 (eds Wallach, H. et al.) 8024–8035 (Curran Associates, 2019).
42. Bocquet, M., Pires, C. A. & Wu, L. Beyond Gaussian statistical modeling in geophysical data assimilation. *Mon. Weather Rev.* **138**, 2997–3023 (2010).
43. Raanes, P. N., Chen, Y., Grudzien, C., Tondeur, M. & Dubois, R. DAPPER: data assimilation with Python: a package for experimental research. *GitHub* https://github.com/nansencenter/DAPPER (2022).
44. Berkooz, G., Holmes, P. & Lumley, J. L. The proper orthogonal decomposition in the analysis of turbulent flows. *Annu. Rev. Fluid Mech.* **25**, 539–575 (1993).
45. Peherstorfer, B. & Willcox, K. Data-driven operator inference for nonintrusive projection-based model reduction. *Comput. Methods Appl. Mech. Eng.* **306**, 196–215 (2016).
46. Lee, K. & Parish, E. J. Parameterized neural ordinary differential equations: applications to computational physics problems. *Proc. R. Soc. A* **477**, 20210162 (2021).
47. San, O., Pawar, S. & Rasheed, A. Variational multiscale reinforcement learning for discovering reduced order closure models of nonlinear spatiotemporal transport systems. *Sci. Rep.* **12**, 17947 (2022).
48. Schmidt, J., Krämer, N. & Hennig, P. A probabilistic state space model for joint inference from differential equations and data. In *Proc. Advances in Neural Information Processing Systems* Vol. 34 (eds Ranzato, M. et al.) 12374–12385 (Curran Associates, 2021).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

# Methods

### Reparametrized drift residual

Recall that, given a linear SDE, $\mathrm{d}X_t = (-A_\varphi(t)X_t + b_\varphi(t))\mathrm{d}t + \Sigma_\theta \mathrm{d}\beta_t$, with initial condition $X_0 \sim \mathcal{N}(m_0, S_0)$, we can solve for the marginal statistics of this SDE, $q_\varphi(x_t) = \mathcal{N}(m_\varphi(t), S_\varphi(t))$, by solving the ODEs in equations (5) and (6). Given $S_\varphi$ and $\dot{S}_\varphi$, we notice that equation (6) is a set of Lyapunov equations in terms of the symmetric matrix $A_\varphi$. We can rearrange this set of equations to express $A_\varphi$ and $b_\varphi$ explicitly in terms of $m_\varphi$ and $S_\varphi$

$$\mathrm{vec}(A_\varphi(t)) = (S_\varphi(t) \oplus S_\varphi(t))^{-1}\mathrm{vec}(\Sigma_\theta Q \Sigma_\theta^T - \dot{S}_\varphi(t)), \tag{11}$$

$$b_\varphi(t) = \dot{m}(t) + A_\varphi(t)m_\varphi(t). \tag{12}$$

$S_\varphi(t) \oplus S_\varphi(t)(S_\varphi(t) \oplus S_\varphi(t) = S_\varphi(t) \otimes I + I \otimes S_\varphi(t)$ is the Kronecker sum, in which $\otimes$ indicates the standard Kronecker product.) is guaranteed to be invertible because $S_\varphi$ is a valid covariance matrix. Using equations (11) and (12), the drift residual can be rewritten as follows:

$$r(x_t, t, \theta, \varphi) = -A_\varphi(t)x_t + b_\varphi(t) - f_\theta(x_t, t), \tag{13}$$

$$= A_\varphi(t)(m_\varphi(t) - x_t) + \dot{m}(t) - f_\theta(x_t, t), \tag{14}$$

$$\begin{aligned} &= \mathrm{vec}^{-1}((S_\varphi(t) \oplus S_\varphi(t))^{-1}\mathrm{vec}(\Sigma_\theta Q \Sigma_\theta^T - \dot{S}_\varphi(t)))\,(m_\varphi(t) - x_t) \\ &\quad + \dot{m}_\varphi(t) - f_\theta(x_t, t), \end{aligned} \tag{15}$$

Substituting this into the expression for the ELBO (equation (8)) yields the final form of the reparametrized ELBO (equation (9)).

### Parametrization of $m_\varphi$ and $S_\varphi$

We parametrize $m_\varphi$ using radial-basis-function models of the form $m_\varphi(t) = (k^{(1)}(t, \tau) \otimes I)\varphi^{(1)}$, in which $\varphi^{(1)} \in \mathbb{R}^{pD}$ is the vector of weights associated with the approximation for the mean, $k^{(1)}(t, \tau) = [1\ k^{(1)}(t, t_1)\ \ldots\ k^{(1)}(t, t_{p-1})] \in \mathbb{R}^{1\times p}$ is the vector of basis functions evaluated at $t$ and $\tau = \{t_1, \ldots, t_{p-1}\}$ is the set of basis function centres. We chose each $k^{(j)}$ to be the Matérn 5/2 kernel[49] centred at $t_i \in [0, T]$,

$$k^{(j)}(t, t_i) = \sigma_j^2(1 + \sqrt{5}\,r(t, t_i)/\ell_j + 5r(t, t_i)^2/(3\ell_j^2))\exp(-\sqrt{5}\,r(t, t_i)/\ell_j),$$

in which $\sigma_j > 0$ and $\ell_j > 0$ are the tunable scale and length scales associated with the basis functions, respectively, and $r(t, t_i) = |\kappa(t) - \kappa(t_i)|$ with $\kappa(t) = T - T(1 - (t/T)^\alpha)^\beta$ denoting the Kumaraswamy warping function with two positive tunable parameters $\alpha$ and $\beta$ (ref. 50). In all experiments, unless otherwise noted, we used 200 evenly spaced basis functions within the time interval $[0, T]$.

When dealing with systems whose dynamics can be characterized in a low-dimensional state space, we use a full-rank, dense, spectral parametrization of the state covariance matrix $S_\varphi(t) = R_\varphi(t)\Lambda_\varphi(t)R_\varphi(t)^T$, in which $R_\varphi(t)$ is an orthonormal matrix and $\Lambda_\varphi(t)$ is a diagonal matrix with all positive elements that we parametrize as $R_\varphi(t) = \exp(((k^{(2)}(t, \tau) \otimes I)\varphi^{(2)})^\vee)$ and $\Lambda_\varphi(t) = \mathrm{softplus}(\mathrm{diag}((k^{(3)}(t, \tau) \otimes I)\varphi^{(3)}))$, in which $\varphi^{(2)} \in \mathbb{R}^{pD(D-1)/2}$ is the vector of weights associated with $R_\varphi$, $\varphi^{(3)} \in \mathbb{R}^{pD}$ is the vector of weights associated with $\Lambda_\varphi$, $k^{(j)}(t, \tau) = [1\ k^{(j)}(t, t_1)\ \ldots\ k^{(j)}(t, t_{p-1})] \in \mathbb{R}^{1\times p}$ is the vector of basis functions evaluated at $t$, $(\cdot)^\vee : \mathbb{R}^{D(D-1)/2} \to \mathbb{R}^{D\times D}$ is an operator that turns a vector of length $D(D-1)/2$ into a skew-symmetric matrix (notation from ref. 1), $\exp : \mathbb{R}^{D\times D} \to \mathbb{R}^{D\times D}$ indicates the matrix exponential, $\mathrm{diag} : \mathbb{R}^D \to \mathbb{R}^{D\times D}$ turns a vector of length $D$ into a diagonal matrix and $\mathrm{softplus} : \mathbb{R} \to \mathbb{R}$ is the Softplus function that acts element wise. This parametrization requires $\mathcal{O}(D^3)$ time to compute.

Under this parametrization, we can solve equation (6) for $A_\varphi$ by first solving the system of linear equations

$$(\Lambda_\varphi(t) \oplus \Lambda_\varphi(t))\mathrm{vec}(B_\varphi(t)) = \mathrm{vec}(R_\varphi(t)^T(\Sigma_\theta Q \Sigma_\theta^T - \dot{S}_\varphi(t))R_\varphi(t)) \tag{16}$$

and then computing $A_\varphi(t) = R_\varphi(t)B_\varphi(t)R_\varphi(t)^T$. This result follows from the application of the Bartels–Stewart algorithm[51].

To summarize, this parametrization allows for complete flexibility in terms of its ability to approximate symmetric positive-definite matrices; however, this flexibility comes at the cost of scaling as $\mathcal{O}(D^3)$ owing to the matrix–matrix products. This computational cost makes it only applicable for use with relatively low-dimensional systems.

When dealing with high-dimensional dynamical systems, as is often the case in the geosciences for example, we propose parametrizing $S_\varphi$ by a purely diagonal covariance matrix, that is, $S_\varphi(t) = \Lambda_\varphi(t)$, in which $\Lambda_\varphi(t)$ is parametrized as described previously for the spectral parametrization. If we also restrict $\Sigma_\theta Q \Sigma_\theta^T$ to be diagonal, we have $A_\varphi(t) = \frac{1}{2}S_\varphi(t)^{-1}(\Sigma_\theta Q \Sigma_\theta^T - \dot{S}_\varphi(t))$. The dimensionality of this parametrization scales linearly in the dimension of the state, $\mathcal{O}(D)$, making state estimation with unknown governing equations possible in extremely high dimensions.

We use the diagonal parametrization for the reduced-order modelling problem in the main text as well as the upcoming studies of our approach applied to symbolic-governing-equation discovery for high-dimensional spatially extended systems, as well as the example of our approach applied to the binary black hole. For all other numerical studies, we use the full spectral parametrization.

### Maximization of the ELBO

We approximate all gradients of expectations using the standard reparametrization trick[30]. If the observation function is linear, we can exactly compute the expected log-likelihood, as is described in an upcoming section. We used the so-called hybrid simulator based on Gaussian quadrature[52] to estimate the integral over time with respect to the drift residual. This integration scheme ensures that approximations to the ELBO gradient remain unbiased while providing lower variance than standard Monte Carlo. In the case that the number of data points is large, stochastic gradient ascent can be used to maximize the objective. We used Adam[53] to optimize the ELBO with respect to the variational parameters.

Putting these computational ingredients together, and letting each $x_t^{(j)}, \theta^{(j)} \sim \mathcal{N}(m_\varphi(t), S_\varphi(t))q_\varphi(\theta)$ be samples from the variational distributions drawn using the reparametrization trick, we can write stochastic approximations to the first term in the ELBO as

$$\sum_{i=1}^{N} \mathbb{E}_{\mathcal{N}(m_\varphi(t_i), S_\varphi(t_i))}[\log p(y_{t_i}|x_{t_i})] \approx \frac{N}{BJ}\sum_{i=1}^{B}\sum_{j=1}^{J} \log p(y_{t_i}|x_{t_i}^{(j)}), \tag{17}$$

in which each $i$ has been uniformly sampled from the set $\{1, 2, \ldots, N\}$. Following ref. 52 and letting $I(t) = -\frac{1}{2J}\sum_{j=1}^{J}||r(x_t^{(j)}, t, \theta^{(j)}, \varphi)||^2_{\Sigma_{\theta^{(j)}}Q\Sigma^T_{\theta^{(j)}}}$, we can write a stochastic approximation to the second term in the ELBO as

$$\begin{aligned} &-\frac{1}{2}\int_0^T \mathbb{E}_{\mathcal{N}(m_\varphi(t), S_\varphi(t))q_\varphi(\theta)}[||r(x_t, t, \theta, \varphi)||^2_{\Sigma_\theta Q \Sigma_\theta^T}]\mathrm{d}t \\ &\approx \frac{1}{K}\sum_{k=1}^{K}\left(TI(t_k) - TP_{M-1}(t_k) + \sum_{i=1}^{M}\omega_i I(t_i)\right) \end{aligned} \tag{18}$$

in which $\omega_i \in \mathbb{R}$ is the $i$th quadrature weight associated with the Gauss–Legendre quadrature rule[54] and $P_{M-1}$ is the $(M-1)$-degree polynomial interpolant of $I(t)$ constructed by matching the value of $I(t)$ at the quadrature nodes. Recall that we have selected $q_\varphi(\theta)$ and $p(\theta)$ such that we can write the KL divergence between the two distributions in closed form.

To demonstrate the robustness of our approach, we kept hyperparameters constant across all experiments unless otherwise specified. We chose $B = \min(128, \text{number of data points})$, $J = 32$, $K = 26$ and $M = 102$. We chose a learning rate of $10^{-3}$ for all parameters related to $q_\varphi(x_t)$ and $10^{-2}$ for all other parameters; for the Lorenz '96 problem, we chose learning rates $10^{-2}$ and $10^{-1}$, respectively. We trained every model for 20,000 iterations; for the Lorenz '96 problem, we trained for 5,000 iterations. We used 5,000 warm-up iterations (1,250 for the Lorenz '96 problem) wherein the KL divergence term is scaled by a constant factor from 0 to 1, increasing every iteration at a linear rate[55]. We decayed the learning rate by multiplying the starting learning rate by 0.9 every 2,500 iterations (625 for the Lorenz '96 problem). Although these were the hyperparameter settings we chose for all experiments here, we would like to emphasize that this is just one setting for the hyperparameters that worked well consistently. Careful tuning for individual experiments can probably improve performance and or decrease training time.

### Initialization

We found that good initialization of the radial-basis-function models used to parametrize $m_\varphi$ and $S_\varphi$ could greatly improve convergence. This is to be expected, given that we are trying to maximize a highly nonconvex objective. For the constant basis function, we initialized the weight to 0 in the mean-function and orthogonal-matrix parametrizations. For the eigenvalue parametrization ($\Lambda_\varphi$), we initialized this weight to approximately −2.5. We also added $10^{-6}$ to the eigenvalue matrix to help ensure that the optimizer stayed away from extremely degenerate regions of the optimization space in the early stages. Initializing the eigenvalues to be small at the start of the optimization procedure helped to avoid convergence to a poor local minima wherein the model finds that the data were generated by a pure random-walk process. For the scale parameter, $\sigma_j$, we always initialized to 1. For the mean function, we initialized the remaining weights in $\varphi^{(1)}$ by minimizing the least-squares error with $\ell_2$ regularization on the training data. We weighted the $\ell_2$ regularization term by $10^{-1}$. We initialized the length scale with a grid search over the length scales from the set $\{10^{-1}, 1/2, 1, 10\}$ using fivefold cross-validation. For the orthogonal matrix, we initialized the weights to a small positive constant, $10^{-6}$. For the sparse approximation to the drift function, we initialized the weights by minimizing the $\ell_2$ regularized least-squares error between the derivatives from the initialized mean function and the drift-function model at the training timestamps. Again, we weighted the $\ell_2$ regularization term by $10^{-1}$.

### Normalization of drift-function features

When training sparse linear models, we found that normalizing the drift-function features dynamically in training could make convergence to a good minima more consistent. In this work, we assumed that we were only interested in time-independent features and that we were using the same features in each dimension. In this case, we can write, $f_\theta(x_t, t) = (\psi(x_t) \otimes I)\theta$, in which $\psi: \mathbb{R}^D \to \mathbb{R}^{1\times M}$ returns a row vector of features and $\theta \in \mathbb{R}^{DM}$ is a vector of parameters. Let $\psi(x)$ be the basis functions evaluated at a batch of inputs. During training, each time we compute $\psi(x)$, we normalize on the basis of the running variance, that is,

$$\psi(x) \leftarrow \frac{\psi(x)}{\sqrt{\mathrm{Var}[\psi(x)] + \epsilon}}, \tag{19}$$

in which $\mathrm{Var}[\psi(x)]$ is the running variance of the output features and $\epsilon > 0$ is a constant we choose to be $10^{-5}$. The running variance is updated according to the rule $\mathrm{Var}[\psi(x)]_{\mathrm{new}} = (1-\mu) \times \mathrm{Var}[\psi(x)]_{\mathrm{old}} + \mu \times \mathrm{Var}[\psi(x)]_{\mathrm{est}}$, in which $\mathrm{Var}[\psi(x)]_{\mathrm{est}}$ is the estimated variance for the current batch of inputs and the momentum, $\mu$, was set to 0.1. The constant basis function $\psi(x) = 1$ was not normalized. This is very similar to the batch-normalization[56] implementation provided in PyTorch[41]. The running variance was initialized using the variance of the inputs at the data timestamps using the initialized mean approximation.

### KL divergence for the half-Cauchy prior

This section summarizes the parametrization defined by ref. 37 that is used to estimate the posterior over the parameters in equation (3) when performing symbolic-governing-equation discovery. This is a useful parametrization as it allows us to easily sample from the approximate posterior and express the KL divergence between the approximate posterior and the prior in closed form.

Recall that the prior for the parameters is written as $p(\theta)$, in which $\theta \in \mathbb{R}^{M+D}$, $M$ is the number of basis functions in the dictionary and $D$ is the dimension of the state. We express the full hierarchical prior as $\theta_i = \widetilde{\theta}_i \sqrt{s_a s_b \alpha_i \beta_i}$, in which

$$\widetilde{\theta}_i \sim \mathcal{N}(0,1),\ s_a \sim \mathcal{G}(0.5, \tau_0^2),\ s_b \sim \mathcal{IG}(0.5, 1), \tag{20}$$

$$\alpha_i \sim \mathcal{G}(0.5, 1) \text{ and } \beta_i \sim \mathcal{IG}(0.5, 1). \tag{21}$$

Here $\mathcal{G}$ and $\mathcal{IG}$ denote the Gamma and inverse Gamma distributions, respectively, and $\tau_0$ is a small positive constant chosen by the user, typically $\mathcal{O}(10^{-5}) - \mathcal{O}(10^{-7})$. The product $z = \sqrt{s_a s_b}$ corresponds to a half-Cauchy distribution on $z$. The idea behind the prior is 'global–local' shrinkage. Here $s_a$ and $s_b$ are 'global' scales that encourage all parameters to be small and $\alpha_i$ and $\beta_i$ are local scales that allow the corresponding parameter to remain unconstrained.

Having specified the prior, we must now specify an approximate posterior. Following ref. 37, we make use of a mean-field assumption, approximating the posterior over the shrinkage parameters using log-normal distributions. The approximate posterior over all parameters can be expressed as the product

$$q_\varphi(\theta) = q_\varphi(s_a, s_b) \prod_{i=1}^{M+D} q_\varphi(\alpha_i, \beta_i) q_\varphi(\widetilde{\theta}_i), \tag{22}$$

in which $\varphi$ is the vector of variational parameters corresponding to the distributions

$$q_\varphi(s_a, s_b) = \mathcal{LN}(s_a | \mu_{s_a}, \sigma_{s_a}^2) \mathcal{LN}(s_b | \mu_{s_b}, \sigma_{s_b}^2); \tag{23}$$

$$q_\varphi(\alpha_i, \beta_i) = \mathcal{LN}(\alpha_i | \mu_{\alpha_i}, \sigma_{\alpha_i}^2) \mathcal{LN}(\beta_i | \mu_{\beta_i}, \sigma_{\beta_i}^2); \tag{24}$$

$$q_\varphi(\widetilde{\theta}_i) = \mathcal{N}(\widetilde{\theta}_i | \mu_{\widetilde{\theta}_i}, \sigma_{\widetilde{\theta}_i}^2). \tag{25}$$

Given this choice of prior and posterior, the KL divergence between the approximate posterior and the prior factorizes as follows:

$$D_{\mathrm{KL}}(q_\varphi(\theta) || p(\theta)) = D_{\mathrm{KL}}(q_\varphi(s_b) || p(s_b)) + D_{\mathrm{KL}}(q_\varphi(\alpha) || p(\alpha)) + \tag{26}$$

$$D_{\mathrm{KL}}(q_\varphi(\beta) || p(\beta)) + D_{\mathrm{KL}}(q_\varphi(\widetilde{\theta}) || p(\widetilde{\theta})). \tag{27}$$

We can write each term in the KL divergence between the approximate posterior and the prior as

$$\begin{aligned} D_{\mathrm{KL}}(q_\varphi(s_b) || p(s_b)) = &\exp\left(\frac{1}{2}\sigma_{s_b}^2 - \mu_{s_b}\right) \\ &- \frac{1}{2}(-\mu_{s_b} + 2\log\sigma_{s_a} + 1 + \log 2), \end{aligned} \tag{28}$$

$$\begin{aligned} D_{\mathrm{KL}}(q_\varphi(\alpha) || p(\alpha)) = \sum_{i=1}^{M+D} \Bigg(&\exp\left(\mu_{\alpha_i} + \frac{1}{2}\sigma_{\alpha_i}^2\right) \\ &- \frac{1}{2}(\mu_{\alpha_i} + 2\log\sigma_{\alpha_i} + 1 + \log 2)\Bigg), \end{aligned} \tag{29}$$

$$\begin{aligned} D_{\mathrm{KL}}(q_\varphi(\beta) || p(\beta)) = \sum_{i=1}^{M+D} \Bigg(&\exp\left(\frac{1}{2}\sigma_{\beta_i}^2 - \mu_{\beta_i}\right) \\ &- \frac{1}{2}(-\mu_{\beta_i} + 2\log\sigma_{\beta_i} + 1 + \log 2)\Bigg), \end{aligned} \tag{30}$$

$$D_{\mathrm{KL}}(q_{\varphi}(\widetilde{\theta})||p(\widetilde{\theta})) = -\frac{1}{2}\sum_{i=1}^{M+D}(1+2\log\sigma_{\widetilde{\theta}_i} - \mu_{\widetilde{\theta}}^2 - \sigma_{\widetilde{\theta}_i}^2). \tag{31}$$

### Linear measurement function

In the case of a linear observation function with Gaussian noise, the expected log likelihood of the measurements can be written in closed form. Specifically, when $y_{t_i} = Gx(t_i) + \eta$, in which $G \in \mathbb{R}^{d\times D}$ and $\epsilon \sim \mathcal{N}(0, R)$, we can write

$$\begin{aligned}\mathbb{E}_{\mathcal{N}(m(t_i),S(t_i))}[\log p(y_{t_i}|x_{t_i})] = -\frac{1}{2}(y_i - Gm(t_i))^T R^{-1}(y_i - Gm(t_i)) \\ -\frac{1}{2}\mathrm{tr}(G^T R^{-1} G S(t_i)) - \frac{1}{2}\log|R| - \frac{d}{2}\log 2\pi.\end{aligned} \tag{32}$$

This is a useful, well-known result that reduces the variance of gradients of the approximation to the ELBO in equations (17) and (18).

### List of governing equations used in numerical studies

Most of the problems used for benchmarking studies have been used previously to benchmark the performance of methods for governing-equation discovery[27,34]. We list the governing equations, initial conditions, time intervals and the distribution over the probabilistic corruption coefficients. As a reminder, we add the following probabilistic corruption to the dynamics of the first state in the second set of numerical studies: $\dot{x}_1 \leftarrow \dot{x}_1 - \alpha x_2 + \beta$.

Damped linear oscillator:

$$\dot{x} = \begin{bmatrix}-0.1 & 2\\ -2 & -0.1\end{bmatrix}x, \quad x(0) = \begin{bmatrix}2.5\\ -5\end{bmatrix}, \quad t \in [0, 20] \\ \alpha, \beta \sim \mathcal{U}(-0.25, 0.25) \tag{33}$$

Damped cubic oscillator:

$$\dot{x} = \begin{bmatrix}-0.1 & 2\\ -2 & -0.1\end{bmatrix}x^3, \quad x(0) = \begin{bmatrix}0\\ -1\end{bmatrix}, \quad t \in [0, 25] \\ \alpha, \beta \sim \mathcal{U}(-0.15, 0.15) \tag{34}$$

Coupled linear system:

$$\begin{aligned}&\ddot{x}_1 = -(4+2)x_1 + 2x_2 \\ &\ddot{x}_2 = 2x_1 - (2+4)x_2\end{aligned}, \quad x(0) = \begin{bmatrix}0\\ 1\end{bmatrix}, \\ \dot{x}(0) = \begin{bmatrix}0\\ 0\end{bmatrix}, \quad t \in [0, 20] \\ \alpha, \beta \sim \mathcal{U}(-0.25, 0.25) \tag{35}$$

Duffing oscillator:

$$\begin{aligned}&\dot{x}_1 = x_2 \\ &\dot{x}_2 = -(x_1^3 - x_1) - 0.35x_2\end{aligned} \quad x(0) = \begin{bmatrix}3\\ 2\end{bmatrix}, \quad t \in [0, 20] \\ \alpha, \beta \sim \mathcal{U}(-0.25, 0.25) \tag{36}$$

Selkov glycolysis[57]:

$$\begin{aligned}&\dot{x}_1 = -x_1 + 0.08x_2 + x_1^2 x_2 \\ &\dot{x}_2 = 0.6 - 0.08x_2 - x_1^2 x_2\end{aligned} \quad x(0) = \begin{bmatrix}0.7\\ 1.25\end{bmatrix}, \quad t \in [0, 30] \\ \alpha, \beta \sim \mathcal{U}(-0.04, 0.04) \tag{37}$$

Lorenz '63:

$$\begin{aligned}&\dot{x}_1 = 10(x_2 - x_1) \\ &\dot{x}_2 = x_1(28 - x_3) - x_2 \\ &\dot{x}_3 = x_1 x_2 - 8/3x_3\end{aligned} \quad x(0) = \begin{bmatrix}-8\\ 7\\ 27\end{bmatrix}, \quad t \in [0, 10] \\ \alpha, \beta \sim \mathcal{U}(-2.5, 2.5) \tag{38}$$

Hopf bifurcation:

$$\begin{aligned}&\dot{x}_1 = 0.5x_1 + x_2 - x_1(x_1^2 + x_2^2) \\ &\dot{x}_2 = -x_1 + 0.5x_2 - x_2(x_1^2 + x_2^2)\end{aligned} \quad x(0) = \begin{bmatrix}2\\ 2\end{bmatrix}, \quad t \in [0, 20] \\ \alpha, \beta \sim \mathcal{U}(-0.125, 0.125) \tag{39}$$

### Detailed setup for example 1

This section provides a detailed breakdown of the comparison activities related to state estimation with a known motion model introduced in the first set of numerical studies. To reiterate, in this set of benchmarks, we compared our method (SVISE) to the PF[42] using an implementation provided by DAPPER[43] when the form of the motion model is known exactly. This serves as the best-case scenario for standard state-estimation methods.

We use the dynamical system definitions described previously as well as the identity observation function. For each dynamical system, we generate 20 independent datasets and task the algorithms with estimating the mean and variance of the state at each timestamp. Each dataset consists of 128 evenly spaced data points generated by simulating the system forward from the initial condition listed previously using Euler–Maruyama integration assuming a diffusion matrix whose diagonal values are given by 1% of the range of the system and additive zero-centred Gaussian observation noise with a standard deviation that is 10% of the range of the system. We define the range of the system as $1/2(\max(\{x(t_i)\}_{i=1}^N) - \min(\{x(t_i)\}_{i=1}^N))$, in which $x(t_i)$ is the true system state at time $t_i$ assuming zero diffusion.

To assess the performance of the methods, we used the normalized root mean squared error (NRMSE) between the mean estimate for the state and the true state position, $\mathrm{NRMSE} = \left(\sum_{i=1}^N ||x(t_i) - x_i||_2^2 / \sum_{i=1}^N ||x(t_i)||_2^2\right)^{1/2}$, in which $x_i$ is the predicted mean of the state at time $i$ and $x(t_i)$ is the true system state.

The state estimators in the DAPPER implementation all require an initial condition. To not unfairly advantage our method, we provided the algorithms we compared ourselves to with the true initial condition for all systems except the Hopf bifurcation (for which we provided the true system state at the time of the first measurement to avoid particles diverging to infinity). Also, we set the initial variance to be 0, as we were providing the PF with the exact initial condition. For the Hopf bifurcation, so as to not unfairly advantage our method, we only measured performance for estimates of the state provided after 10 s (thereby giving the PF sufficient burn-in time).

For all comparisons in this section, we used the default settings provided by DAPPER v1.3.0 (ref. 43), with the exception of choosing 1,000 particles. The results were summarized in the main text and in Fig. 2a.

### Detailed setup for example 2

This section provides a detailed description of the comparison activities related to state estimation with modelling errors that was introduced in the second set of numerical studies. To reiterate, in contrast to the comparisons described in the previous section, in this set of comparisons, we introduce a corruption to the differential equations that the PF was not made aware of. If the uncorrupted dynamics are given by $\dot{x} \in \mathbb{R}^D$, in which $\dot{x}_i$ is the differential equation governing the dynamics of the $i$th component of $x$, we corrupt the dynamics of the first state by setting $\dot{x}_1 \leftarrow \dot{x}_1 - \alpha x_2 + \beta$ while all other states are left uncorrupted. The corruption terms $\alpha$ and $\beta$ are sampled from zero-centred uniform distributions using the bounds listed previously. In the case of the cubic oscillator, for example, this introduces a small amount of extra linear damping to the system. We use the same settings as in the previous benchmarks except for the fact that we generate data by simulating with the corrupted dynamics. For the PF, we set reg = 2.4 and NER = 0.3. These settings were found by starting with the DAPPER defaults and manually tuning.

Looking to Fig. 2b, we see that our method outperforms the PF on average at this level of corruption. To reiterate what was discussed in the main text, we believe that these results make clear the usefulness of this work. Even in the presence of mild modelling error, our method outperforms standard state-estimation tools. As modelling error is an unavoidable reality in many systems of interest, our method for state estimation has the potential to be useful.

### Detailed setup for example 3

To generate data, we used an immersed boundary projection method codebase for solving the two-dimensional incompressible Navier–Stokes equations[58]. We generated 1,953 evenly spaced snapshots each with dimension of 596,602 over the time interval [61, 256.2]. We assume a Gaussian likelihood with a constant variance of $10^{-3}$. In this problem, 38 POD modes were required to capture 90% of the variance. The fully connected neural network model for the drift was chosen to have one hidden layer with 128 hidden units and tanh nonlinearities. We used a batch size of 64, set $K = 100$, $M = 100$ and used 500 basis functions to approximate the posterior of the state. To reiterate what was mentioned previously, looking to Fig. 3, we see that we are successfully able to infer a probabilistic ROM for the flow.

### Example 4: symbolic-governing-equation discovery benchmarking

In this section, we compare our method to standard governing-equation learning algorithms from the literature on the suite of benchmark systems listed previously in Methods. For all experiments in this section, we require $g$ to be the identity measurement function so that methods from the literature can be applied. Also, we provide our method with no knowledge of the underlying dynamics (that is, we set $f_{\varnothing} = 0$ and $\Sigma_{\varnothing} = 0$). We compared our method to the algorithm for sparse identification of nonlinear dynamics (SINDy), SINDy with sequentially thresholded least squares regression (SINDy-STLSQ)[27], SINDy with sparse relaxed regularized regression (SINDy-SR3)[59] and Ensemble-SINDy (ENS-SINDy)[60] using implementations provided by PySINDy[61,62]. Each method is given a dictionary of functions containing all polynomials up to order 5.

For each system, we generate 40 independent datasets and task the algorithms with recovering the underlying governing equations. These 40 independent datasets are split up equally into two cases: (1) the low-noise, low-data regime and (2) the high-noise, large-data regime. In the low-noise, low-data regime, we generate 16 data points for the damped linear and damped cubic oscillators, 32 data points for the Hopf bifurcation, Selkov glycolysis model and Duffing oscillator, and 64 data points for the Lorenz '63 and coupled linear oscillator. Each data point is sampled by assuming zero-centred Gaussian noise with a standard deviation that is 1% of the range of the system. In the high-noise, large-data regime, we generate 2,048 evenly spaced data points for each dataset assuming zero-centred Gaussian noise with a standard deviation that is 25% of the range of the system. We again define the range of the system as $1/2\left(\max\left(\{x(t_i)\}_{i=1}^{N}\right) - \min\left(\{x(t_i)\}_{i=1}^{N}\right)\right)$, in which $x(t_i)$ is the true system state at time $t_i$.

The reconstruction error and the number of incorrect terms are metrics commonly used in compressive sensing to validate the efficacy of sparse learning algorithms; see, for example, refs. 63,64. The reconstruction error is defined as $\mathrm{RER} = ||\theta - \hat{\theta}||_2/||\theta||_2$, in which $\theta$ is the true vector of sparse weights that we are trying to estimate and $\hat{\theta}$ is our estimate for the set of sparse weights. For all estimates of the reconstruction error, we use the mean of the estimated posterior for the weights. The reason for looking at both metrics is that some algorithms may achieve a low reconstruction error without correctly pruning weights.

Summary plots for benchmarking in the high-data regime are provided in Extended Data Fig. 1. Detailed benchmarking tables broken down by each system can be found in Extended Data Tables 2–5. In these tables, the error bars for all terms are given by one standard deviation from the mean. A dash indicates that the equation-recovery algorithm predicted that the governing equations were given by $\dot{x} = 0$.

It can be seen from the results that, as well as improving on reconstruction error, often by more than an order of magnitude, our method was able to identify the correct functional form of the governing equations far more frequently than the methods with which we compared in both the low-data and high-noise regimes. These results are notable because, in many problems for which governing equations are not available, we only have access to noisy/limited data.

Looking more closely at Extended Data Table 2, we see that the proposed method (SVISE) outperformed all other methods in terms of reconstruction error in the low-data regime except on the Duffing oscillator problem, for which our approach was outperformed by ENS-SINDy. For this benchmark, our approach still outperformed ENS-SINDy in terms of the number of mismatched terms. Looking now to Extended Data Table 3, we see that the proposed method outperformed all other methods in terms of the number of mismatched terms for all but the Lorenz '63 benchmark; however, the SINDy-STLSQ method achieved this lower number of incorrect terms score by often ignoring the dynamics in the third state.

In the high-noise regime, we see that SVISE outperformed the methods we compared with in terms of RER on all but the Duffing oscillator benchmark, for which our approach was again outperformed by ENS-SINDy (see Extended Data Table 4). For this benchmark, our method was again the clear winner in terms of the number of mismatched terms (see Extended Data Table 5).

Taken together, these results demonstrate that the method introduced in this work is a good choice for governing-equation discovery in the low-data and high-noise regimes. Also, although the methods we compared ourselves to in this section require that the measurement function is identity (that is, the full state vector is measured), our method is applicable to cases with arbitrary observation functions.

### Example 5: high-dimensional, spatially extended differential-equation discovery with low-rank observation matrices

We now consider the problem of recovering the underlying governing equations for the Lorenz '96 system with 1,024 states using a low-rank observation matrix. The Lorenz '96 model is a set of coupled, chaotic, ODEs designed to be a simplified model of the climate along a line of constant latitude[65]. The governing equations for this system are given by $\dot{x}_k = x_{k-1}(x_{k+1} - x_{k-2}) - x_k + 10$, for which the boundary conditions are assumed to be periodic ($k = 1, 2,\ldots, 1{,}024$).

We generated observation matrices of rank $r$ using the expression $g(x) = (r^{-1}\sum_{i=1}^{r} u_i u_i^T)x$, in which each $u_i \in \mathbb{R}^D$ is a random vector sampled from a standard normal distribution. We studied the performance of our approach in which the rank of the observation matrix is 256, 512 and 1,024. For each experiment, we used 512 snapshots over the time interval of 0 to 10 corrupted by noise that is 2% of the range of the system. We make the assumption that the dynamics are given by $\dot{x}_i = \mathcal{P}_2(x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2})\theta$, in which $\mathcal{P}_2 : \mathbb{R}^5 \to \mathbb{R}^{1\times M}$ returns all quadratic polynomial functions that are a function of $x_i$ and its two closest neighbours to the left and right of the node. Although we found that we were able to exactly recover the underlying functional form of the governing equations with an observation matrix whose rank was half the dimensionality of the state, further work is required to theoretically establish conditions under which the governing equations can be exactly recovered. These results are summarized in Extended Data Fig. 2.

Because many real-world systems for which governing equations are challenging to derive from first principles are both (1) high-dimensional and (2) challenging to measure, we believe this to be a useful result. Our method has a computational cost that scales linearly in the state dimension and can be applied given an arbitrary observation function. This result opens the door for equation discovery in systems that were previously believed to be too large and/or difficult to measure.

### Example 6: governing-equation discovery for second-order systems with unobserved states

We now examine the problem of learning governing equations of second-order systems using only displacement/position measurements. For many physical systems, we often only have access to position measurements (that is, through GPS/Vicon camera measurements) for certain states and/or accelerometer measurements for other states. To apply existing methods from the literature to this class of problem, it becomes necessary to estimate velocities and accelerations using finite-difference approximations[66] in the preprocessing stage or use the weak form of the governing equations[67]. The proposed method can be directly applied to such problems without resorting to finite differences or the weak form.

We consider the task of learning the governing equations of the coupled linear oscillator defined in governing-equations definitions discussed previously. We generate 32 measurements of the system position corrupted by 10% measurement noise. Assuming that the system is autonomous and second order, we know that the dynamics must be governed by second-order differential equations, which are functions of both the position and the velocity.

Using a dictionary containing polynomials up to order five in the state and velocity, we were able to infer both the structural form of the underlying governing equations as well as estimates for the uncertainty in the parameters. The governing equations inferred by our method are provided below (only the mean coefficients are shown):

$$\begin{aligned}\ddot{x}_1 &= -5.23x_1 + 1.74x_2,\\ \ddot{x}_2 &= 1.61x_1 - 5.45x_2,\end{aligned} \tag{40}$$

which is relatively close to the true functional form of the underlying governing equations. Extended Data Fig. 3 shows the measurements and underlying state inferred by our method. It can be seen that the probabilistic state estimate agrees well with the true trajectory.

This experiment demonstrates the use of our method in practical situations in which we only have access to partial measurements of the state for second-order systems. So far, we have only considered problems in which the observation function is linear. In the next section, we consider a problem with a nonlinear observation function in which there are fewer observations than states.

### Example 7: binary black hole system from gravitational-wave measurements

The binary-black-hole modelling problem is concerned with estimating black-hole orbital trajectories using observations of gravitational waves[68]. In comparison with the examples considered so far, this problem involves a nonlinear observation likelihood and the number of observations is less than the number of states. Moreover, in this problem, it is not clear what choice of basis functions might be appropriate for approximating the underlying dynamics. We shall demonstrate that it is indeed possible to use a neural network in place of a linear combination of basis functions, as was mentioned when outlining our approach.

Traditional approaches to solving this problem typically involve reconciling gravitational-wave measurements with the complex partial differential equations that govern their dynamics. This is typically a computationally expensive undertaking. In this section, we attempt to infer the governing equations that were likely to have generated the waveform observations simultaneously to the underlying orbital trajectories. We consider a special case of the binary-black-hole modelling problem for an extreme-mass-ratio system (that is, in which the mass of one object is far greater than the other); see Keith et al.[69] and references therein for more details on this problem.

Centring the origin of the coordinate system at the more massive object, the dynamics can be written as a set of coupled differential equations in terms of the angle of the smaller object with respect to the *x* axis, $\delta(t)$, and the anomaly, $\chi(t)$,

$$\dot{\delta} = \frac{(p-2-2e\cos\chi)(1+e\cos\chi)^2}{p^{3/2}((p-2)^2-4e^2)^{1/2}}, \tag{41}$$

$$\dot{\chi} = \frac{(p-2-2e\cos\chi)(1+e\cos\chi)^2(p-6-2e\cos\chi)^{1/2}}{p^2((p-2)^2-4e^2)^{1/2}}, \tag{42}$$

in which $e = 0.5$ is the eccentricity and $p = 100$ is the semilatus rectum. The angle and anomaly relate to the orbital trajectory according to

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \frac{p}{(1+e\cos\chi(t))}\begin{bmatrix} \cos\delta(t) \\ \sin\delta(t) \end{bmatrix}. \tag{43}$$

As mentioned previously, in practice, we cannot observe the state variables directly. Instead, we only have access to noisy gravitational-waveform measurements, $w(t)$,

$$w(t) = \sqrt{\frac{4\pi}{5}}\left(\frac{\mathrm{d}^2}{\mathrm{d}t^2}x(t)^2 - \frac{\mathrm{d}^2}{\mathrm{d}t^2}y(t)^2\right). \tag{44}$$

Note that we consider only the dominant (2,2)-mode gravitational waveforms.

Given gravitational-waveform observations, $w(t)$, our goal is to reconstruct the underlying orbital trajectories, $x(t)$ and $y(t)$, and to infer with an approximate forward model that can be used to forecast future orbital trajectories. Rather than working directly in the trajectory coordinates, we will infer a SDE in terms of the orbital parameters, $\delta(t)$ and $\chi(t)$, in which $\chi(t)$ is the anomaly and $\delta(t)$ is the angle with respect to the *x* axis of the smaller object. Following the parametrization suggested by Keith et al.[69], we model the drift as

$$f_\theta(t,\delta,\chi) = \frac{(1+e\cos\chi)^2}{Mp^{3/2}}(1+\mathcal{F}_\theta(\cos\chi)), \tag{45}$$

in which $\mathcal{F}_\theta$ is a fully connected neural network with two outputs, *M* is the mass of the more massive object, $e = 0.5$ is the eccentricity and $p = 100$ is the semilatus rectum. We use two hidden layers, each with 128 hidden units and tanh nonlinearities. We collect 1,000 evenly spaced gravitational-waveform observations, $w(t)$, over the interval $[0, 0.6 \times 10^5]$ corrupted by Gaussian noise with a standard deviation of $10^{-3}$. We also provide our algorithm with the initial condition of the underlying state at the first observation time. We place a sparsity-inducing prior on the diagonal of the diffusion term.

We choose a batch size of 256, a learning rate of $10^{-2}$, 20 samples from the variational posterior, 1,000 warm-up iterations, 100 basis functions to approximate the posterior over the state and decayed the learning rate by 0.9 every 500 iterations; see Methods for more details. The results are summarized in Extended Data Fig. 4. We find that we are able to infer a reasonable model for the orbital trajectory while estimating the state. We reiterate that, in contrast to previous approaches to solving this problem, we were not required to solve any differential equations in training. Also, our approach provides probabilistic predictions for the orbital trajectories.

Like for the Lorenz '96 example, further work is required to determine when uncovering the governing equations is possible for general nonlinear observation functions, particularly when there are fewer observations than states. For example, in this problem, we found that the success of our approach relied heavily on the carefully designed parametrization for the drift term suggested by Keith et al.[69]. It is also worth mentioning that, because we are estimating the parameters of the neural network using maximum-likelihood estimation, we expect to systematically underestimate uncertainty. Future work could consider placing priors on the neural-network parameters and performing approximate variational inference over said parameters to more accurately capture uncertainty.

## Data availability

The data in the paper and the Supplementary Information are available at https://github.com/coursekevin/svise.

## Code availability

The software library that was developed to generate the results along with documentation is available at https://github.com/coursekevin/svise.

49. Williams, C. K. & Rasmussen, C. E. *Gaussian Processes for Machine Learning* Vol. 2 (MIT Press, 2006).
50. Snoek, J., Swersky, K., Zemel, R. & Adams, R. Input warping for Bayesian optimization of non-stationary functions. In *Proc. 31st International Conference on Machine Learning* (eds Xing, E. P. & Jebara, T.) 1674–1682 (PMLR, 2014).
51. Bartels, R. H. & Stewart, G. W. Solution of the Matrix Equation AX + XB = C [F4]. *Commun. ACM* **15**, 820–826 (1972).
52. Lee, L.-f. Interpolation, quadrature, and stochastic integration. *Econ. Theory* **17**, 933–961 (2001).
53. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. In *Proc. 3rd International Conference on Learning Representations* (2015).
54. Owen, A. B. *Monte Carlo Theory, Methods and Examples* (Stanford Univ., 2013).
55. Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K. & Winther, O. Ladder variational autoencoders. In *Proc. Advances in Neural Information Processing Systems* Vol. 29 (eds Lee, D. et al.) (Curran Associates, 2016).
56. Ioffe, S. & Szegedy, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proc. 32nd International Conference on Machine Learning* (eds Bach, F. & Blei, D.) 448–456 (PMLR, 2015).
57. Strogatz, S. H. *Nonlinear Dynamics and Chaos: with Applications to Physics, Biology, Chemistry, and Engineering* 2nd edn (Westview Press, 2015).
58. Taira, K. & Colonius, T. The immersed boundary method: a projection approach. *J. Comput. Phys.* **225**, 2118–2137 (2007).
59. Champion, K., Zheng, P., Aravkin, A. Y., Brunton, S. L. & Kutz, J. N. A unified sparse optimization framework to learn parsimonious physics-informed models from data. *IEEE Access* **8**, 169259–169271 (2020).
60. Fasel, U., Kutz, J. N., Brunton, B. W. & Brunton, S. L. Ensemble-SINDy: robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proc. R. Soc. A* **478**, 20210904 (2022).
61. Silva, B. D. et al. PySINDy: a Python package for the sparse identification of nonlinear dynamical systems from data. *J. Open Source Softw.* **5**, 2104 (2020).
62. Kaptanoglu, A. A. et al. PySINDy: a comprehensive Python package for robust sparse system identification. *J. Open Source Softw.* **7**, 3994 (2022).
63. Huang, Y., Beck, J. L., Wu, S. & Li, H. Robust Bayesian compressive sensing for signals in structural health monitoring. *Comput. Aided Civ. Infrastruct. Eng.* **29**, 160–179 (2013).
64. Khashabi, D., Ziyadi, M. & Liang, F. Heteroscedastic relevance vector machine. Preprint available at Preprint at https://arxiv.org/abs/1301.2015 (2013).
65. Lorenz, E. N. in *Predictability of Weather and Climate* (eds Palmer, T. & Hagedorn, R.) 40–58 (Cambridge Univ. Press, 2006).
66. Chartrand, R. Numerical differentiation of noisy, nonsmooth data. *Int. Sch. Res. Not.* **2011**, 164564 (2011).
67. Messenger, D. A. & Bortz, D. M. Weak SINDy: Galerkin-based data-driven model selection. *Multiscale Model. Simul.* **19**, 1474–1497 (2021).
68. Abbott, R. et al. GWTC-2: compact binary coalescences observed by LIGO and Virgo during the first half of the third observing run. *Phys. Rev. X* **11**, 021053 (2021).
69. Keith, B., Khadse, A. & Field, S. E. Learning orbital dynamics of binary black hole systems from gravitational wave measurements. *Phys. Rev. Res.* **3**, 043101 (2021).

**Acknowledgements** K.C. thanks T. Evans for discussions and for providing pointers to previous work on unbiased quadrature; and A. Keil for suggesting background reading on continuous products and random walks. P.B.N. thanks K. Law for discussions on Bayesian inversion. K.C. and P.B.N. thank G. D'Eleuterio and S. Waslander for feedback and suggestions on this research. This work is supported by a grant from the Natural Sciences and Engineering Research Council of Canada.

**Author contributions** K.C. and P.B.N. developed the theory and methods. K.C. implemented the algorithms and carried out the numerical studies. K.C. and P.B.N. designed the numerical experiments, discussed the results and co-wrote the paper. P.B.N. obtained funding and supervised the project.

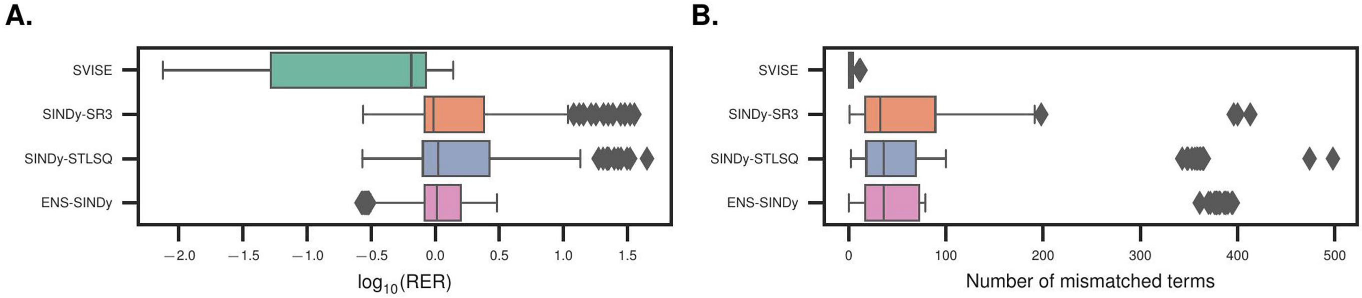
**Competing interests** The authors declare no competing interests.

**Additional information**

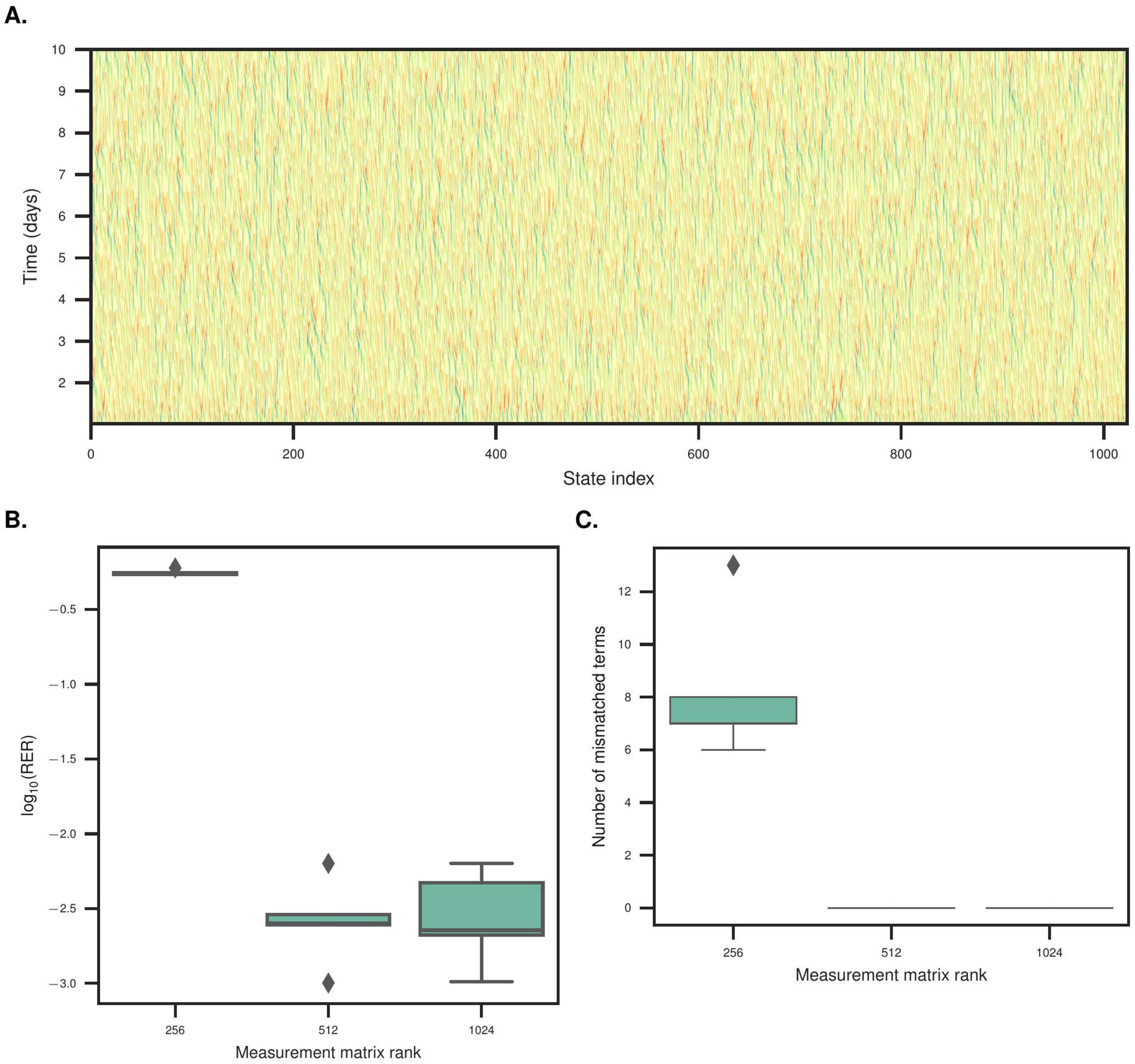
**Correspondence and requests for materials** should be addressed to Prasanth B. Nair.

**Peer review information** *Nature* thanks Brendan Keith and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

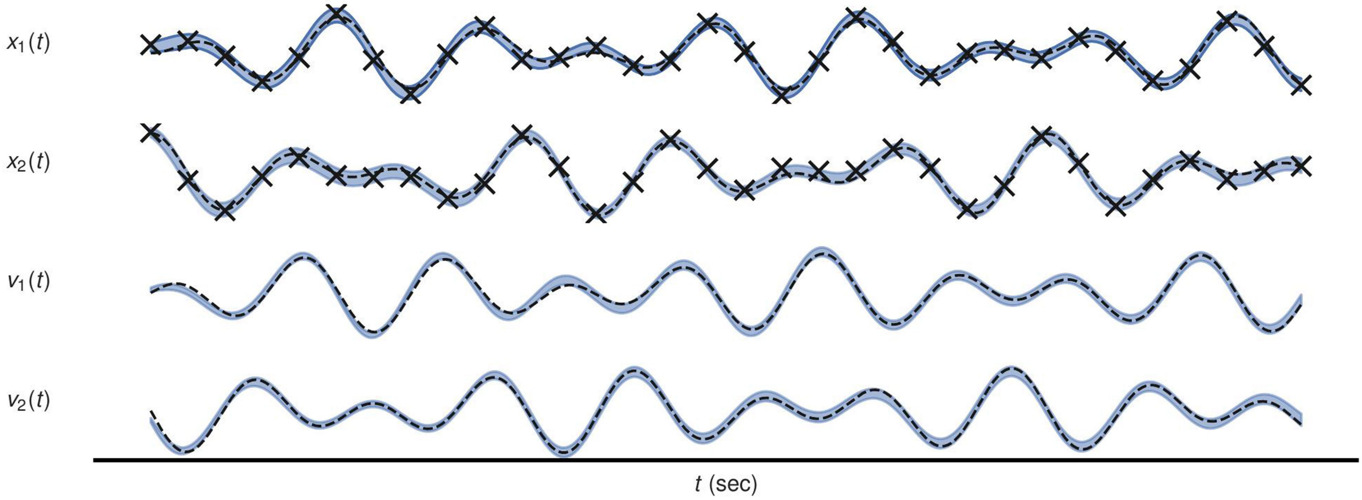
**Reprints and permissions information** is available at http://www.nature.com/reprints.



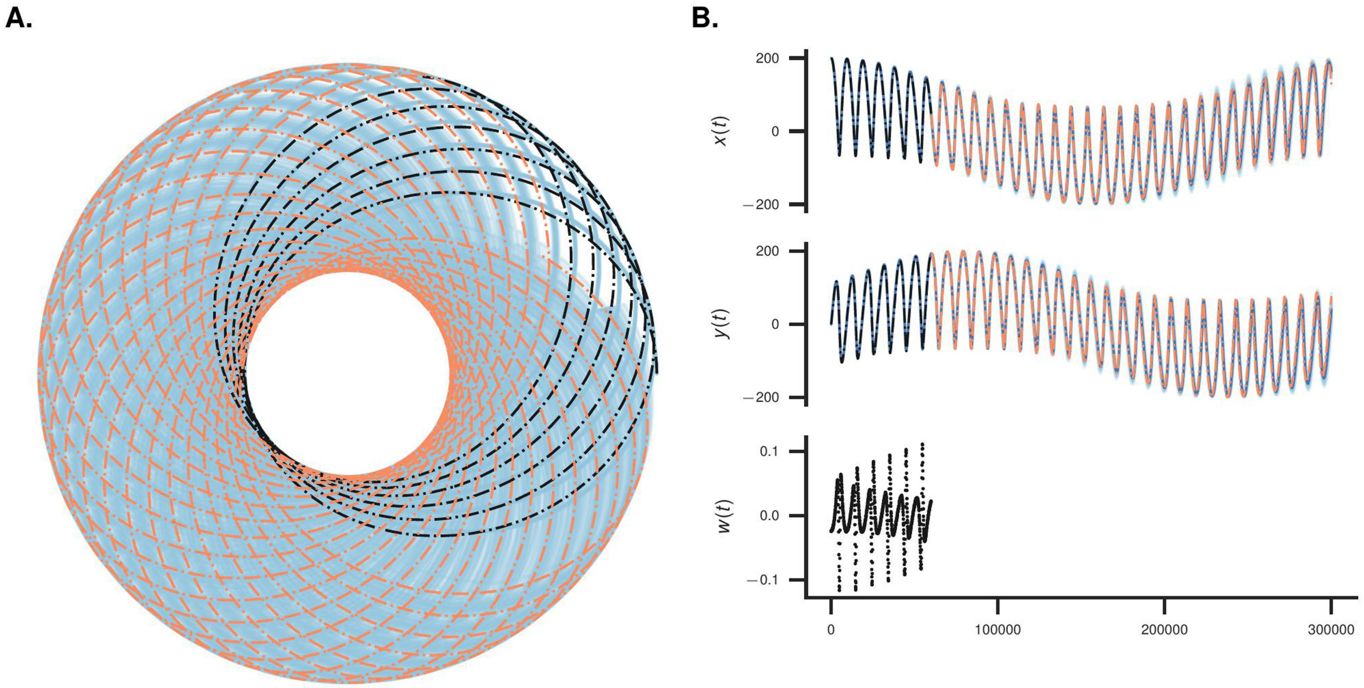
**Extended Data Fig. 1 | Identifying missing terms. a**, Reconstruction error. **b**, Number of mismatched terms. Our method (SVISE) outperforms state-of-the-art algorithms for governing-equation discovery in the high-noise and low-data regimes.



**Extended Data Fig. 2 | Experiment on discovering governing equations for a high-dimensional system with a low-rank observation matrix.** **a**, Visualization of Lorenz '96 full underlying state data with 1,024 states. **b**, Reconstruction error for varying observation matrix ranks. **c**, Number of mismatched terms for varying observation matrix ranks. In this experiment, we assumed that the governing equations were a sparse linear combination of basis functions from the dictionary of quadratic polynomial functions of the node state and its four closest neighbours to the left and right of the node. We were able to successfully recover the exact functional form of the Lorenz '96 governing equations using an observation matrix whose rank was half that of the full underlying state.



**Extended Data Fig. 3 | Governing-equation discovery for second-order system using only position measurements.** The dashed lines indicate the true oscillator state and the shaded blue regions indicate two standard deviations from the mean of the underlying state estimate. The state positions are marked by $x_1$ and $x_2$ and the state velocities are marked by $v_1$ and $v_2$. Here only 32 measurements of the oscillator position corrupted by noise that was 10% of the range of the system state were available.



**Extended Data Fig. 4 | Binary-black-hole modelling problem. a**, Orbital trajectory estimates. **b**, Trajectory estimates versus time. In this experiment, we approximate the dynamics using a neural network. This is a useful model in situations in which it is not clear what dictionary of basis functions is appropriate. The left figure shows the predicted orbital trajectories in the plane of the orbit and the right figure shows the predicted orbital trajectories as a function of time. The black lines indicate the data windows, the orange lines indicate the testing data and the blue intervals indicate probabilistic predictions from our model. Note that we only have access to the waveform observations, $w(t)$, up to time $0.6 \times 10^5$ in training. We are able to accurately estimate the states and provide probabilistic estimates for the orbital trajectories well into the future, despite not having access to the underlying governing equations.

## Extended Data Table 1 | Assumption comparison table

| Method | Parametric uncertainty | Bayes (non-MAP) | Non-Gaussian posterior | Unknown dynamics | Symbolic dynamics |
|---|---|---|---|---|---|
| ERTS [1] | ✓ | ✗ | ✗ | ✗ | ✗ |
| ESGVI [5] | ✓ | ✓ | ✗ | ✗ | ✗ |
| PF [41] | ✓ | ✓ | ✓ | ✗ | ✗ |
| KoopSE [4] | ✓ | ✗ | ✗ | ✓ | ✗ |
| SVISE (ours) | ✓ | ✓ | ✗ | ✓ | ✓ |

Capability comparison between the extended Rauch–Tung–Striebel smoother (ERTS), the exactly sparse Gaussian variational inference method for state estimation (ESGVI), the PF and our approach, SVISE. We note that ours is the only method that allows for symbolic-governing-equation discovery as well as a Bayes (non-MAP) state estimate.

## Extended Data Table 2 | RER in the low-data regime

| Benchmark | SVISE | SINDy-SR3 | SINDy-STLSQ | ENS-SINDy |
|---|---|---|---|---|
| Damped linear osc. | **0.049** ± **0.003** | 1.001 ± 0.001 | 1.000 ± 0.001 | 1.000 ± 0.001 |
| Damped cubic osc. | **0.046** ± **0.007** | 1.366 ± 0.133 | 1.195 ± 0.500 | 1.008 ± 0.013 |
| Lorenz ‘63 | **0.767** ± **0.081** | 26.082 ± 26.105 | 1.275 ± 1.017 | 1.000 ± 0.000 |
| Hopf bifurcation | **0.618** ± **0.050** | 69.839 ± 36.571 | – | – |
| Selkov glycolysis | **0.074** ± **0.006** | 515.730 ± 202.063 | 0.999 ± 0.005 | 0.989 ± 0.034 |
| Duffing osc. | 1.106 ± 0.021 | 0.979 ± 0.059 | – | **0.992** ± **0.035** |
| Coupled linear | **0.012** ± **0.007** | 0.352 ± 0.296 | 0.345 ± 0.017 | 0.350 ± 0.001 |

Reconstruction error averaged over 20 independent trials. Error bars given by 1 standard deviation.

**Extended Data Table 3 | Number of mismatched terms in the low-data regime**

| Benchmark | SVISE | SINDy-SR3 | SINDy-STLSQ | ENS-SINDy |
|---|---|---|---|---|
| Damped linear osc. | **1.000 ± 0.000** | 5.700 ± 6.520 | 5.650 ± 6.966 | 6.800 ± 8.687 |
| Damped cubic osc. | **1.000 ± 0.000** | 12.150 ± 12.920 | 8.500 ± 10.371 | 13.550 ± 14.692 |
| Lorenz ‘63 | 9.800 ± 1.030 | 93.600 ± 19.075 | **4.550 ± 2.655** | 10.450 ± 7.117 |
| Hopf bifurcation | **3.200 ± 0.600** | 32.150 ± 2.688 | – | – |
| Selkov glycolysis | **1.800 ± 0.400** | 34.400 ± 1.685 | 6.100 ± 0.436 | 5.850 ± 0.477 |
| Duffing osc. | **0.000 ± 0.000** | 4.950 ± 4.842 | – | 3.850 ± 0.654 |
| Coupled linear | **0.000 ± 0.000** | 4.950 ± 2.617 | 3.950 ± 0.218 | 4.000 ± 0.000 |

Number of mismatched terms averaged over 20 independent trials. Error bars given by 1 standard deviation.

**Extended Data Table 4 | RER in the high-noise regime**

| Benchmark | SVISE | SINDy-SR3 | SINDy-STLSQ | ENS-SINDy |
|---|---|---|---|---|
| Damped linear osc. | **0.046** ± **0.009** | 0.308 ± 0.045 | 0.295 ± 0.030 | 0.305 ± 0.046 |
| Damped cubic osc. | **0.064** ± **0.012** | 0.839 ± 0.056 | 0.833 ± 0.059 | 0.867 ± 0.070 |
| Lorenz ‘63 | **0.937** ± **0.171** | 1.417 ± 0.303 | 1.371 ± 0.276 | 1.340 ± 0.214 |
| Hopf bifurcation | **0.727** ± **0.047** | 0.927 ± 0.054 | 0.916 ± 0.055 | 0.921 ± 0.046 |
| Selkov glycolysis | **0.533** ± **0.282** | 18.398 ± 7.731 | 12.969 ± 12.383 | 1.556 ± 0.194 |
| Duffing osc. | 1.015 ± 0.082 | 0.879 ± 0.123 | 0.902 ± 0.095 | **0.878** ± **0.078** |
| Coupled linear | **0.022** ± **0.009** | 2.551 ± 0.193 | 2.718 ± 0.246 | 2.684 ± 0.166 |

Reconstruction error averaged over 20 independent trials. Error bars given by 1 standard deviation.

**Extended Data Table 5 | Number of mismatched terms in the high-noise regime**

| Benchmark | SVISE | SINDy-SR3 | SINDy-STLSQ | ENS-SINDy |
|---|---|---|---|---|
| Damped linear osc. | **1.200** ± **0.510** | 8.850 ± 13.499 | 4.800 ± 7.814 | 7.950 ± 10.745 |
| Damped cubic osc. | **2.000** ± **0.000** | 22.050 ± 7.263 | 21.950 ± 8.482 | 23.050 ± 9.367 |
| Lorenz ‘63 | **8.650** ± **1.931** | 91.150 ± 2.574 | 68.400 ± 3.693 | 71.700 ± 3.593 |
| Hopf bifurcation | **4.400** ± **0.800** | 22.800 ± 5.913 | 22.200 ± 7.400 | 24.000 ± 7.849 |
| Selkov glycolysis | **3.150** ± **1.824** | 29.450 ± 4.769 | 28.550 ± 6.111 | 29.300 ± 7.894 |
| Duffing osc. | **0.000** ± **0.000** | 29.750 ± 12.078 | 32.950 ± 10.562 | 28.550 ± 12.576 |
| Coupled linear | **0.000** ± **0.000** | 184.100 ± 54.828 | 324.050 ± 99.701 | 381.100 ± 7.999 |

Number of mismatched terms averaged over 20 independent trials. Error bars given by 1 standard deviation.