scientific reports

OPEN

Deep kernel learning of dynamical models from high-dimensional noisy data

Nicolò Botteghi✉, Mengwu Guo & Christoph Brune

This work proposes a stochastic variational deep kernel learning method for the data-driven discovery of low-dimensional dynamical models from high-dimensional noisy data. The framework is composed of an encoder that compresses high-dimensional measurements into low-dimensional state variables, and a latent dynamical model for the state variables that predicts the system evolution over time. The training of the proposed model is carried out in an unsupervised manner, i.e., not relying on labeled data. Our learning method is evaluated on the motion of a pendulum—a well studied baseline for nonlinear model identification and control with continuous states and control inputs—measured via high-dimensional noisy RGB images. Results show that the method can effectively denoise measurements, learn compact state representations and latent dynamical models, as well as identify and quantify modeling uncertainties.

Understanding the evolution of dynamical systems over time by discovering their governing laws is essential for science and and engineering[1]. Traditionally, governing equations are derived from physical principles, such as conservation laws and symmetries. However, the governing laws are often difficult to unveil for many systems exhibiting strongly nonlinear behaviors. These complex behaviors are typically captured by high-dimensional noisy measurements, which makes it especially hard to identify the underlying principles. On the other hand, while measurement data are often abundant for many dynamical systems, physical equations, if known, may not exactly govern the actual system evolution due to various uncertainties.

The progress of Machine Learning[2] and Deep Learning[3], combined with the availability of large amounts of data, has paved the road for new paradigms for the analysis and understanding of dynamical systems[1]. These new paradigms are not limited to the discovery of governing laws for system evolution, and have brought revolutionary advancements to the field of dynamical system control. In particular, Reinforcement Learning[4] (RL) has opened the door to model-free control directly from high-dimensional noisy measurements, in contrast to the traditional control techniques that rely on accurate physical models. RL has found its success in the nature-inspired learning paradigm through interaction with the world, in which the control law is solely a function of the measurements and learned by iteratively evaluating its performance a posteriori, i.e., after being applied to the system. Especially, RL stands outs in the control of complex dynamical systems[5]. However, RL algorithms may suffer from high computational cost and data inefficiency as a result of disregarding any prior knowledge about the world.

While data are often high-dimensional, many physical systems exhibit low-dimensional behaviors, effectively described by a limited number of latent state variables that can capture the principal properties of the systems. The process of encoding high-dimensional measurements into a low-dimensional latent space and extracting the predominant state variables is called, in the context of RL and Computer Science, State Representation Learning[6,7]. At the same time, its counterpart in Computational Science and Engineering is often referred to as Model Order Reduction[8].

Reducing the data dimensionality and extracting the latent state variables is often the first step to explicitly represent a reduced model describing the system evolution. Due to their low dimensionality, such reduced models are often computationally lightweight and can be efficiently queried for making predictions of the dynamics[9] and for model-based control, e.g., Model Predictive Control[10] and Model-based RL[4]. The problem of dimensionality reduction and reduced-order modeling is traditionally tackled by the Singular Value Decomposition[11] (SVD) (depending on the context, the SVD is often referred to as Principal Component Analysis[12] or Proper Orthogonal Decomposition[13]). Examples include the Dynamics Mode Decomposition[14,15], sparse identification of latent dynamics[16] (SINDy), operator inference[17–19], and Gaussian process surrogate modeling[20]. More recently, Deep Learning[3], especially a specific type of neural network (NN) termed Autoencoder[3] (AE), has been employed

Mathematics of Imaging and AI, University of Twente, Enschede, Netherlands. ✉email: n.botteghi@utwente.nl

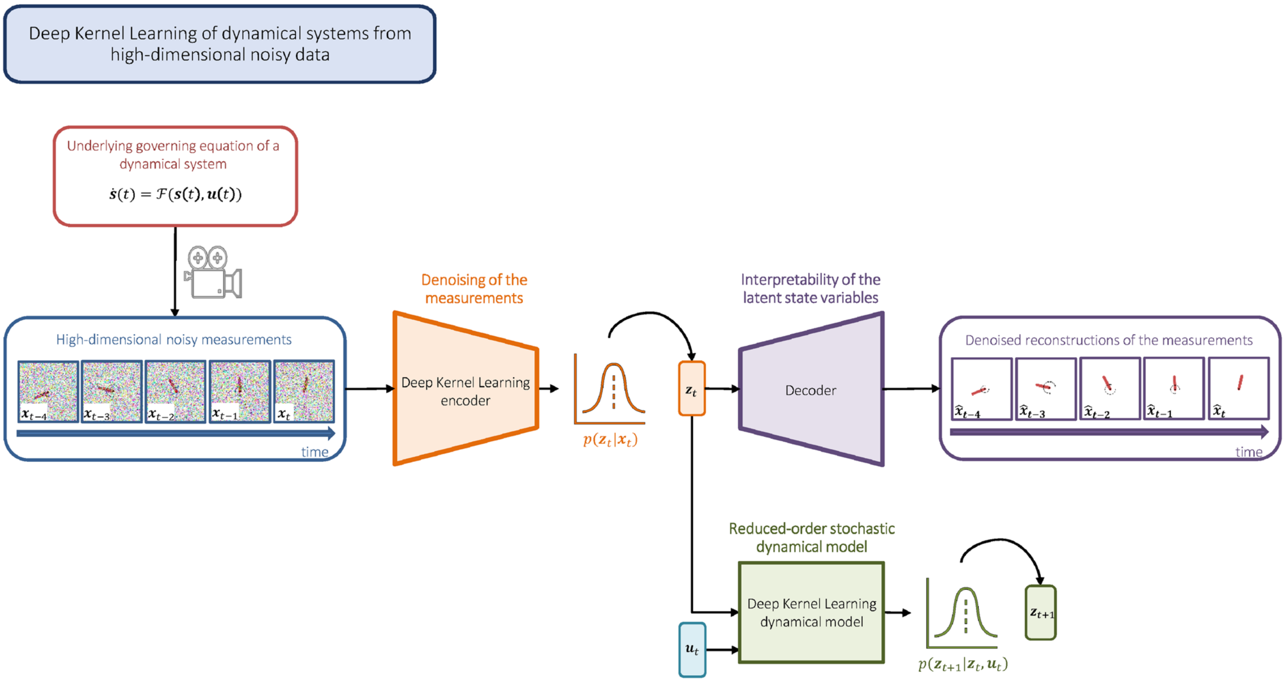
Deep Kernel Learning of dynamical systems from high-dimensional noisy data
Underlying governing equation of a dynamical system
$\dot{s}(t) = \mathcal{F}(s(t), u(t))$
High-dimensional noisy measurements
x_{t-4}
x_{t-3}
x_{t-2}
x_{t-1}
x_t
time
Denoising of the measurements
Deep Kernel Learning encoder
$p(z_t|x_t)$
z_t
Interpretability of the latent state variables
Decoder
Denoised reconstructions of the measurements
$\hat{x}_{t-4}$
$\hat{x}_{t-3}$
$\hat{x}_{t-2}$
$\hat{x}_{t-1}$
$\hat{x}_t$
time
Reduced-order stochastic dynamical model
Deep Kernel Learning dynamical model
u_t
$p(z_{t+1}|z_t, u_t)$
z_{t+1}

Figure 1. Deep Kernel Learning for data-driven dimensionality reduction, latent-state model learning, and uncertainty quantification of dynamical systems from high-dimensional noisy data.

to learn compact state representations successfully. Unlike the SVD, an AE learns a nonlinear mapping from the high-dimensional data space to a low-dimensional latent space through an NN called encoder, as well as an inverse mapping through a decoder. AEs can be viewed as a nonlinear generalization of the SVD, enabling more powerful information compression and better expressivity. AEs have been used for manifold learning[21], in combination with SINDy for latent coordinate discovery[22], and in combination with NN-based surrogate models for latent representation learning towards control[23,24].

Whether we aim to identify parameters of a physical equation or learn the entire system evolution from data, we may face an unavoidable challenge stemming from data noise. Inferring complex dynamics from noisy data is not effortless, as the identification, understanding and quantification of various uncertainties is often required. For example, uncertainties may derive from noise-corrupted sensor measurements, system parameters (e.g., uncertain mass, geometry, or initial conditions), modeling and/or approximation processes, and uncertain system behaviors that may be chaotic (e.g., in the motion of a double pendulum) or affected by unknown disturbances (e.g., uncertain external forces or inaccurate actuation). When data-driven methods consider stochasticity and uncertainties quantification, AEs are often replaced with Variational AEs[25] (VAEs) for learning low-dimensional states as probabilistic distributions. Samples from these distributions can be used for the construction of latent state models via Gaussian models[26–30] and nonlinear NN-based models[31,32]. However, NN-based latent models often disregard the distinction among uncertainty sources, especially between the data noise in the measurements and the modeling uncertainties stemming from the learning process, and only estimate the overall uncertainty on the latent state space through the encoder of a VAE model. We argue, however, that disentangling the uncertainty sources is critical for identifying the governing laws and discovering the latent reduced-order dynamics.

In this work, we propose a data-driven framework for the dimensionality reduction, latent-state model learning, and uncertainty quantification based on high-dimensional noisy measurements generated by unknown dynamical systems (see Fig. 1). In particular, we introduce a Deep Kernel Learning[33] (DKL) encoder, which combines the highly expressive NN with a kernel-based probabilistic model of Gaussian process[34] (GP) to reduce the dimensionality and quantify the uncertainty in the noisy measurements simultaneously, followed by a DKL latent-state forward model that predicts the system dynamics with quantifiable modeling uncertainty, and an NN-based decoder designed to enable reconstruction, prevent representation-collapsing, and improve interpretability. Endowed with quantified uncertainties, such a widely applicable and computationally efficient method for manifold and latent model learning is essential for data-driven physical modeling, control, and digital twinning.

Preliminaries

In scalar-valued supervised learning, we have a set of M d-dimensional input samples $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_M] \in \mathscr{X} \subset \mathbb{R}^d$ and the corresponding set of target data $\mathbf{y} = [y_1, \ldots, y_M]^T \in \mathscr{Y} \subset \mathbb{R}$ related by some unknown function $f^\# : \mathscr{X} \rightarrow \mathscr{Y}$. The goal is to find a function f that best approximates $f^\#$. Many function approximators can be used to learn f, but here we introduce Gaussian process regression (GPR)[34]—a non-parametric method for data-driven surrogate modeling and uncertainty quantification (UQ), deep NNs—a popular class of parametric function approximators of Deep Learning, and the Deep Kernel Learning[33] (DKL) that combines the nonlinear expressive power of deep NNs with the advantages of kernel methods in UQ.

Gaussian process regression. A GP is a collection of random variables, any finite number of which follow a joint Gaussian distribution[34].

$$f(\mathbf{x}) \sim \mathrm{GP}(\mu(\mathbf{x}), \mathrm{k}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\gamma})), \quad \mathrm{y} = \mathrm{f}(\mathbf{x}) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)\,, \tag{1}$$

where the GP is characterized by its mean function $\mu(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ and covariance/kernel function $k(\mathbf{x}, \mathbf{x}'; \boldsymbol{\gamma}) = k_{\boldsymbol{\gamma}}(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))]$ hyperparameters γ, $\mathbf{x}$ and $\mathbf{x}'$ being two input locations, and ε is an independent added Gaussian noise term with variance σ_ε^2. A popular choice of the kernel is the automatic relevance determination (ARD) squared exponential (SE) kernel:

$$k_{\boldsymbol{\gamma}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(- \frac{1}{2} \sum_{j=1}^{d} \frac{(x_j - x_j')^2}{l_j^2} \right), \tag{2}$$

where σ_f is the standard deviation hyperparameter and l_j $(1 \leq j \leq d)$ is the lengthscale along each individual input direction. The optimal values of GP hyperparameters $[\boldsymbol{\gamma}, \sigma_\varepsilon^2] = [\sigma_f^2, l_1, \ldots, l_d, \sigma_\varepsilon^2]$ can be estimated via maximum marginal likelihood given the training targets $\mathbf{y}$[34]:

$$\begin{aligned}
[\boldsymbol{\gamma}^*, (\sigma_\varepsilon^2)^*] &= \underset{\boldsymbol{\gamma}, \sigma_\varepsilon^2}{\arg\max} \ \log p(\mathbf{y}|\mathbf{X}) \\
&= \underset{\boldsymbol{\gamma}, \sigma_\varepsilon^2}{\arg\max} \left\{ -\frac{1}{2} \mathbf{y}^T (k_{\boldsymbol{\gamma}}(\mathbf{X}, \mathbf{X}) + \sigma_\varepsilon^2 \mathbf{I})^{-1} \mathbf{y} - \frac{1}{2} \log |k_{\boldsymbol{\gamma}}(\mathbf{X}, \mathbf{X}) + \sigma_\varepsilon^2 \mathbf{I}| - \frac{M}{2} \log(2\pi) \right\}.
\end{aligned} \tag{3}$$

Optimizing the GP hyperparameters through Eq. (3) requires to repeatedly inverse the covariance matrix $k_{\boldsymbol{\gamma}}(\mathbf{X}, \mathbf{X}) + \sigma_\varepsilon^2 \mathbf{I}$, which can be very expensive or even untrackable in the cases of high-dimensional inputs (e.g., images with thousands of pixels) or big datasets ($M \gg 1$).

Given the training data of input-output pairs $(\mathbf{X}, \mathbf{y})$, the Bayes' rule gives a posterior Gaussian distribution of the noise-free outputs $\mathbf{f}^*$ at unseen test inputs $\mathbf{X}^*$:

$$\begin{aligned}
&\mathbf{f}^* | \mathbf{X}^*, \mathbf{X}, \mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)\,, \\
&\qquad \boldsymbol{\mu}^* = k_{\boldsymbol{\gamma}}(\mathbf{X}, \mathbf{X}^*)^T (k_{\boldsymbol{\gamma}}(\mathbf{X}, \mathbf{X}) + \sigma_\varepsilon^2 \mathbf{I})^{-1} (\mathbf{y} - \mu(\mathbf{X}))\,, \\
&\qquad \boldsymbol{\Sigma}^* = k_{\boldsymbol{\gamma}}(\mathbf{X}^*, \mathbf{X}^*) - k_{\boldsymbol{\gamma}}(\mathbf{X}, \mathbf{X}^*)^T (k_{\boldsymbol{\gamma}}(\mathbf{X}, \mathbf{X}) + \sigma_\varepsilon^2 \mathbf{I})^{-1} k_{\boldsymbol{\gamma}}(\mathbf{X}, \mathbf{X}^*)\,.
\end{aligned} \tag{4}$$

Deep neural networks. NNs are parametric universal function approximators[35] composed of multiple layers sequentially stacked together. Each layer contains a set of learnable parameters known as weights and biases. Collected in a vector θ, these NN parameters are optimized via backpropagation[3] for a function f that best approximates $f^\#$:

$$f(\mathbf{x}) = g(\mathbf{x}; \theta) \tag{5}$$

where $g(\mathbf{x}; \theta)$ denotes an NN with input $\mathbf{x}$ and parameters θ. There are three prominent types of NN layers[3]: fully-connected, convolutional, and recurrent. In practice, the three types of layers are often combined to deal with different characteristics of data and increase the expressivity of the NN model.

Deep kernel learning. To mitigate the limited scalability of GPs to high-dimensional inputs, often referred to as the curse of dimensionality, Deep Kernel Learning[33,36,37] was developed to exploit the nonlinear expressive power of deep NNs to learn compact data representations while maintaining the probabilistic features of kernel-based GP models for UQ. The key idea of DKL is to embed a deep NN, representing a nonlinear mapping from the data to the feature space, into the kernel function for GPR as follows:

$$k_{\mathrm{DKL}}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\gamma}, \boldsymbol{\theta}) = k_{\boldsymbol{\gamma}}(g(\mathbf{x}; \boldsymbol{\theta}), g(\mathbf{x}'; \boldsymbol{\theta}))\,, \tag{6}$$

where $g(\mathbf{x}; \boldsymbol{\theta})$ is an NN with input $\mathbf{x}$ and parameters (weights and biases) $\boldsymbol{\theta}$. Similar to conventional GPs, different kernel functions can be chosen. The GP hyperparameters and the NN parameters are jointly trained by maximizing the marginal likelihood as in Eq. (3).

Thanks to its strong expressive power and versatility, DKL has gained attention in many fields of scientific computing, such as computer vision[33,38,39], natural language processing[40], robotics[36], and meta-learning[41]. However, DKL still suffers from computational inefficiency due to the need for repeatedly inverting the $M \times M$ covariance matrix in Eq. (3) when the dataset is large ($M \gg 1$). In addition, the posterior will be intractable if we change to non-Gaussian likelihoods, and there is no efficient stochastic training[3] (e.g., stochastic gradient descent) that is available for DKL models. All these facts make DKL unable to handle large datasets. To overcome these three limitations, stochastic variational DKL[42] (SVDKL) was introduced. SVDKL utilizes variational inference[34] to approximate the posterior distribution with the best fitting Gaussian to a set of inducing data points sampled from the posterior. Our framework is built upon the SVDKL model.

Rather than other popular deep learning tools, SVDKL is chosen for three main reasons: (i) compared with deterministic NN-based models, GPs—kernel-based models—offer better quantification of uncertainties[33,34], (ii) compared with Bayesian NNs[43], SVDKL is computationally cheaper and feasible to the integration of any deep NN architecture, and (iii) compared with ensemble NNs, SVDKL is memory efficient as only a single model needs to be trained.

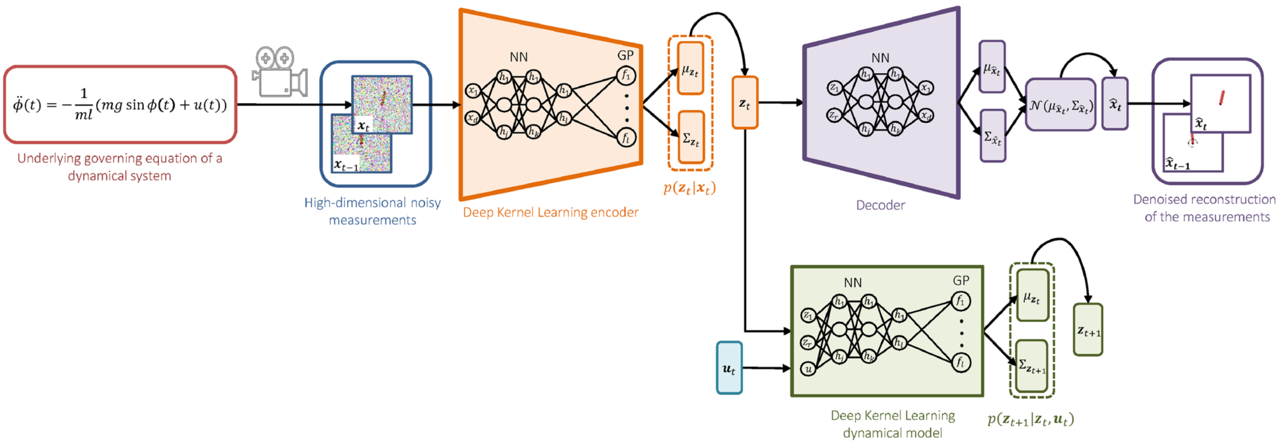
$\ddot{\phi}(t) = -\frac{1}{ml}(mg\sin\phi(t) + u(t))$
Underlying governing equation of a dynamical system
x_t
x_{t-1}
High-dimensional noisy measurements
NN
GP
Deep Kernel Learning encoder
μ_{z_t}
Σ_{z_t}
$p(z_t|x_t)$
z_t
NN
Decoder
$\mu_{\hat{x}_t}$
$\Sigma_{\hat{x}_t}$
$\mathcal{N}(\mu_{\hat{x}_t}, \Sigma_{\hat{x}_t})$
$\hat{x}_t$
$\hat{x}_{t-1}$
Denoised reconstructions of the measurements
u_t
NN
GP
Deep Kernel Learning dynamical model
μ_{z_t}
$\Sigma_{z_{t+1}}$
z_{t+1}
$p(z_{t+1}|z_t, u_t)$

Figure 2. Uncertainty quantification and disentangling with stochastic variational deep kernel learning for dynamical systems generating high-dimensional noisy data.

Methods

In our work, we consider nonlinear dynamical systems generally written in the following form:

$$\frac{d}{dt}\mathbf{s}(t) = \mathscr{F}(\mathbf{s}(t), \mathbf{u}(t)), \quad \mathbf{s}(t_0) = \mathbf{s}_0, \quad t \in [t_0, t_f], \tag{7}$$

where $\mathbf{s}(t) \in \mathscr{S} \subset \mathbb{R}^n$ is the state vector at time t, $\mathbf{u}(t) \in \mathscr{U} \subset \mathbb{R}^m$ is the control input at time t, $\mathscr{F} : \mathscr{S} \times \mathscr{U} \to \mathscr{S}$ is a nonlinear function determining the evolution of the system given the current state $\mathbf{s}(t)$ and control input $\mathbf{u}(t)$, $\mathbf{s}_0$ is the initial condition, and t_0 and t_f are the initial and final time, respectively. In many real-world applications, the state $\mathbf{s}(t)$ is not directly accessible and the function $\mathscr{F}$ is unknown. In spite of this, we can obtain indirect information about the systems through measurements from different sensor devices (measurements can derive, for example, from cameras, laser scanners, or inertial measurement units). Due to the time-discrete nature of the measurements, we indicate with $\mathbf{x}_t$ the measurement vector at a generic time-step t, and $\mathbf{x}_{t+1}$ the measurement at time-step $t+1$.

Given a set of M d-dimensional measurements $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_M] \in \mathscr{X} \subset \mathbb{R}^d$ with $d \gg 1$ and control inputs $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_{M-1}] \in \mathscr{U}$, we consider the problem of learning: (a) a meaningful representation of the unknown states, and (b) a surrogate model for $\mathscr{F}$. However, the high-dimensionality and noise corruption of measurement data makes the two-task learning problem extremely challenging.

Learning latent state representation from measurements.

To begin with, we introduce an SVDKL encoder $E : \mathscr{X} \to \mathscr{Z}$ used to compress the measurements into a low-dimensional latent space $\mathscr{Z}$. Due to the measurement noise, rather than being deterministic, E should map each measurement to a distribution over the latent state space $\mathscr{Z}$. The SVDKL encoder is depicted in Fig. 2. A latent state sample can be obtained as:

$$\begin{aligned} z_{i,t} &= f_i^E(\mathbf{x}_t) + \varepsilon_E, \quad \varepsilon_E \sim \mathscr{N}(0, \sigma_E^2) \\ f_i^E(\mathbf{x}_t) &\sim \mathrm{GP}(\mu(\mathrm{g_E}(\mathbf{x}_t; \boldsymbol{\theta}_\mathrm{E})), \mathrm{k}(\mathrm{g_E}(\mathbf{x}_t; \boldsymbol{\theta}_\mathrm{E}), \mathrm{g_E}(\mathbf{x}'_{t'}; \boldsymbol{\theta}_\mathrm{E}); \boldsymbol{\gamma}_\mathrm{E,i})), \quad 1 \le \mathrm{i} \le |\mathbf{z}|, \end{aligned} \tag{8}$$

where $z_{i,t}$ is the sample from the ith GP with kernel k and mean m, $g_E(\mathbf{x}_t; \boldsymbol{\theta}_E)$ is the feature vector output of the NN part of the SVDKL encoder E, ε_E is an independently added noise, and $|\mathbf{z}|$ indicates the dimension of $\mathbf{z}$.

Because we have no access to the actual state values, we cannot directly use supervised learning techniques to optimize the parameters $[\boldsymbol{\theta}_E, \boldsymbol{\gamma}_E, \boldsymbol{\sigma}_E^2]$ of the SVDKL encoder. Therefore, we utilize a decoder neural network D to reconstruct the measurements given the latent state samples. These reconstructions, denoted by $\hat{\mathbf{x}}_t$, are also used to generate trainable gradients for the SVDKL encoder, which is a common practice for training VAEs. Similar to VAEs, an important aspect of the architecture is the bottleneck created for the low dimensionality of the learned state space $\mathscr{Z}$. While the SVDKL encoder E learns $p(\mathbf{z}_t|\mathbf{x}_t)$, the decoder D learns the inverse mapping $p(\hat{\mathbf{x}}_t|\mathbf{z}_t)$ in which $\hat{\mathbf{x}}_t$ is the reconstruction of $\mathbf{x}_t$. We call this autoencoding architecture SVDKL-AE. To the best of our knowledge, this is the first attempt at training a DKL model without labeled data (unsupervisedly).

Given a randomly sampled minibatch of measurements, we can define the loss function for an SVDKL-AE as follows:

$$\mathscr{L}_E(\boldsymbol{\theta}_E, \boldsymbol{\gamma}_E, \boldsymbol{\sigma}_E^2, \theta_D) = \mathbb{E}_{\mathbf{x}_t \sim \mathbf{X}}[-\log p(\hat{\mathbf{x}}_t|\mathbf{z}_t)], \tag{9}$$

where $\hat{\mathbf{x}}_t|\mathbf{z}_t \sim \mathscr{N}(\boldsymbol{\mu}_{\hat{\mathbf{x}}_t}, \boldsymbol{\Sigma}_{\hat{\mathbf{x}}_t})$ is obtained by decoding the samples of $\mathbf{z}_t|\mathbf{x}_t$ through D. By minimizing the loss function in Eq. (9) with respect to the encoder and decoder parameters, as analogously practiced with VAEs, we can obtain a compact representation of the measurements.

Though our SVDKL-AE resembles a VAE in terms of network architecture and training strategy, we highlight two major advantages of the SVDKL-AE, which have motivated its use in this work:

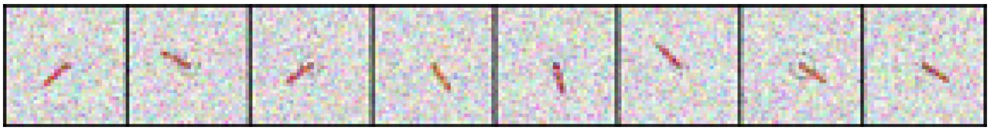

Figure 3. High-dimensional noisy measurements of the pendulum.

- The SVDKL encoder explicitly learns the full distribution $p(\mathbf{z}|\mathbf{x})$ from which we can sample the latent states $\mathbf{z}$ reduced from the full-order states $\mathbf{x}$. A VAE only learns the mean vector and covariance matrix (often chosen to be diagonal) of an assumed joint Gaussian distribution. Clearly, SVDKL-AE should be able to deal with different types of complex distributions more effectively.
- SVDKL-AE can exploit the kernel structure of a Gaussian process to quantify uncertainties, even effectively in low-data regimes[38,40,42]. The kernel choice can be tailored to incorporate prior knowledge into the data-driven modeling.

Learning latent dynamical model. We aim to learn a surrogate dynamical model F predicting the system evolution given the latent state variables sampled from $\mathscr{Z}$ and the control inputs in $\mathscr{U}$. Due to the uncertainties present in the system, we learn a stochastic model $F : \mathscr{Z} \times \mathscr{U} \rightarrow \mathscr{Z}$. Similar to E, the dynamical model F is constructed using an SVDKL architecture. The next latent states $\mathbf{z}_{t+1}$ can be sampled with F:

$$\begin{aligned} z_{i,t+1} &= f_i^F(\mathbf{z}_t, \mathbf{u}_t) + \varepsilon_F, \quad \varepsilon_F \sim \mathscr{N}(0, \sigma_F^2) \\ f_i^F(\mathbf{z}_t, \mathbf{u}_t) &\sim \mathrm{GP}(\mu(\mathrm{g_F}(\mathbf{z}_\mathrm{t}, \mathbf{u}_\mathrm{t}; \boldsymbol{\theta}_\mathrm{F})), \mathrm{k}(\mathrm{g_F}(\mathbf{z}_\mathrm{t}, \mathbf{u}_\mathrm{t}; \boldsymbol{\theta}_\mathrm{F}), \mathrm{g_F}(\mathbf{z}'_{\mathrm{t}'}, \mathbf{u}'_{\mathrm{t}'}; \boldsymbol{\theta}_\mathrm{F}); \boldsymbol{\gamma}_{\mathrm{F,i}})), \quad 1 \leq \mathrm{i} \leq |\mathbf{z}|, \end{aligned} \tag{10}$$

where $z_{i,t+1}$ is sampled from the i^{th} GP, $g_F(\mathbf{z}_t, \mathbf{u}_t; \boldsymbol{\theta}_E)$ is the feature vector output of the NN part of the SVDKL dynamical model F, and ε_F is a noise term.

Again, we do not have access to the true state values obtained by applying the (unknown) control law, but only the sequence of measurements at different time-steps. Here we employ a commonly used strategy in State Representation Learning[31,32,44], which encodes the measurement $\mathbf{x}_{t+1}$ into the distribution $p(\mathbf{z}_{t+1}|\mathbf{x}_{t+1})$ through the SVDKL encoder E, and uses such a distribution as the target for $p(\mathbf{z}_{t+1}|\mathbf{z}_t, \mathbf{u}_t)$ given by the dynamical model F. Therefore, the dynamical model F is trained by minimizing the Kullback-Leibler divergence between the distributions $p(\mathbf{z}_{t+1}|\mathbf{x}_{t+1})$ and $p(\mathbf{z}_{t+1}|\mathbf{z}_t, \mathbf{u}_t)$ (more details in Appendix). The loss for training F is formulated as follows:

$$\mathscr{L}_\mathrm{F}(\boldsymbol{\theta}_F, \boldsymbol{\gamma}_F, \boldsymbol{\sigma}_F^2) = \mathbb{E}_{\mathbf{x}_t, \mathbf{x}_{t+1} \sim \mathbf{X}, \mathbf{u}_\mathrm{t} \sim \mathbf{U}}[\mathrm{KL}[p(\mathbf{z}_{t+1}|\mathbf{x}_{t+1})||p(\mathbf{z}_{t+1}|\mathbf{z}_t, \mathbf{u}_t)]]], \tag{11}$$

where $p(\mathbf{z}_{t+1}|\mathbf{z}_t, \mathbf{u}_t)$ is obtained by feeding a sample from $p(\mathbf{z}_t|\mathbf{x}_t)$ and a control input $\mathbf{u}_t$ to F.

Joint training of models. Instead of training E and F separately, we train them jointly by allowing the gradients of the dynamical model F to flow through the encoder E as well. The overall loss function is

$$\mathscr{L}_{REP}(\boldsymbol{\theta}_E, \boldsymbol{\gamma}_E, \boldsymbol{\sigma}_E^2, \boldsymbol{\theta}_F, \boldsymbol{\gamma}_F, \boldsymbol{\sigma}_F^2, \theta_D) = \mathbb{E}_{\mathbf{x}_t, \mathbf{x}_{t+1} \sim \mathbf{X}, \mathbf{u}_\mathbf{t} \sim \mathbf{U}}[-\log p(\hat{\mathbf{x}}_t|\mathbf{z}_t)) + \beta \mathrm{KL}[p(\mathbf{z}_{t+1}|\mathbf{x}_{t+1})||p(\mathbf{z}_{t+1}|\mathbf{z}_t, \mathbf{u}_t)]], \tag{12}$$

in which $\beta = 1.0$ is used to scale the contribution of the two loss terms.

Variational inference. The two SVDKL models in this work utilize variational inference to approximate the posterior distributions in (8) and (10) with a known family of candidate distributions (e.g., joint Gaussian distributions). The need for variational inference stems from the stochastic gradient descent optimization procedure used for the modeling training[42]. Therefore, we add two extra items to the loss function in Eq. (12), one for each SVDKL model in the following form:

$$\mathscr{L}_\mathrm{var}(\boldsymbol{\theta}, \boldsymbol{\gamma}) = \mathrm{KL}[p(\mathbf{v})||q(\mathbf{v})], \tag{13}$$

in which $p(\mathbf{v})$ is the posterior to be approximated over a collection of sampled locations $\mathbf{v}$ termed *inducing points*, and $q(\mathbf{v})$ represents an approximating candidate distribution. Similar to the original SVDKL work[42], the inducing points are placed on a grid.

Results

Numerical example. For our experiments, we consider the pendulum described by the following equation:

$$\ddot{\phi}(t) = -\frac{1}{ml}(mg \sin\phi(t) + u(t)), \tag{14}$$

where ϕ is the angle of the pendulum, $\ddot{\phi}$ is the angular acceleration, m is the mass, l is the length, and g denotes the gravity acceleration. We assume no access to ϕ or its derivatives, and the measurements are RGB images of size 84 × 84 × 3 obtained through an RGB camera. Examples of high-dimensional and noisy measurements are shown in Fig. 3.

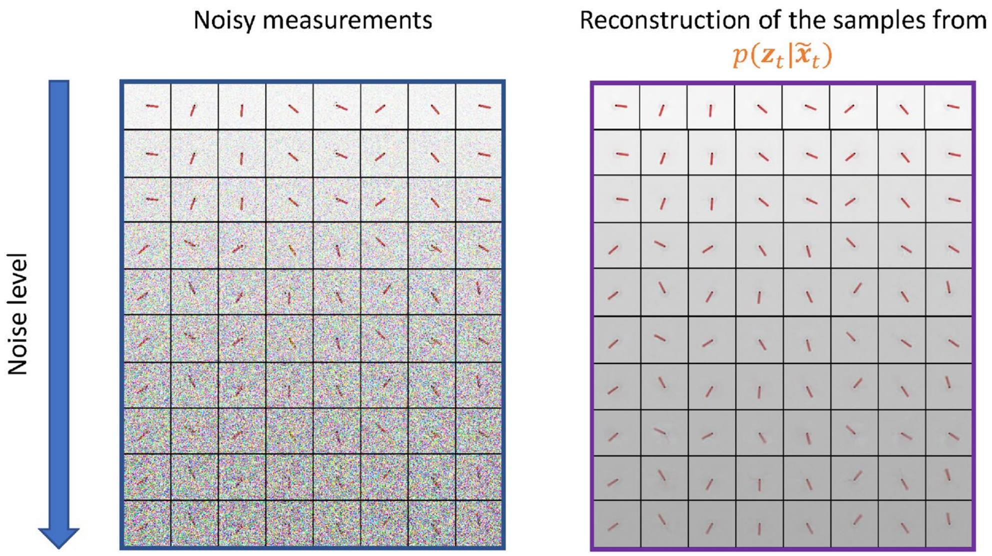

Figure 4. Reconstructions $\hat{\mathbf{x}}_t$ with different noise levels in the measurements $\tilde{\mathbf{x}}_t$. As shown by the sharp reconstructions of $\mathbf{z}_t$, SVDKL-AE can effectively denoise the measurements.

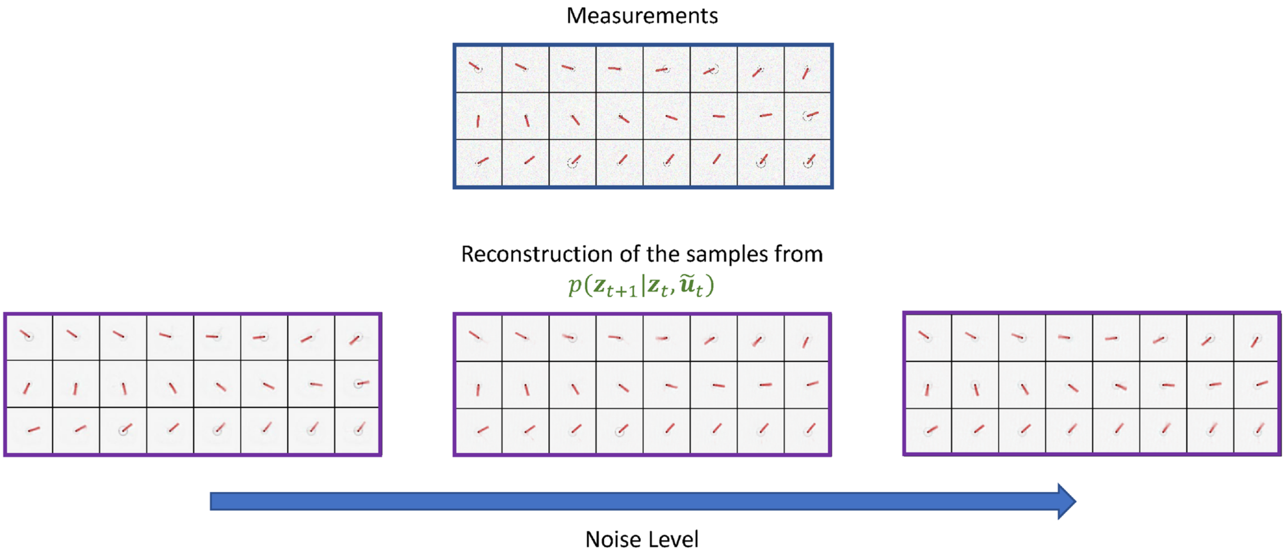
Measurements
Reconstruction of the samples from
$p(z_{t+1}|z_t, \tilde{u}_t)$
Noise Level

Figure 5. Reconstructions $\hat{\mathbf{x}}_{t+1}$ with different noise levels in the control inputs $\mathbf{u}_t$. As shown by the sharp reconstructions of $\mathbf{z}_{t+1}$, the SVDKL forward model can denoise the corrupted control inputs $\mathbf{u}_t$ and predict the dynamics accurately.

The measurements are collected by applying torque values u sampled from a random control law with different initial conditions. The training set is composed of 15000 data tuples $(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1})$, while the test set is composed of 2000 data tuples. Different random seeds are used for collecting training and test sets. The complete list of hyperparameters used in our experiments is shown in Appendix.

Denoising. In Fig. 4, we show the denoising capability of the proposed framework by visualizing the reconstructions of the high-dimensional noisy measurements. The measurements are corrupted by additive Gaussian noise $\mathcal{N}(0, \sigma_x^2)$:

$$\tilde{\mathbf{x}}_t = \mathbf{x}_t + \varepsilon_{\mathbf{x}}, \quad \varepsilon_{\mathbf{x}} \sim \mathcal{N}(0, \sigma_x^2)\,. \tag{15}$$

Moreover, in Fig. 5, we show the reconstructions of the next latent states $\mathbf{z}_{t+1}$ sampled from the dynamic model distribution $p(\mathbf{z}_{t+1}|\mathbf{z}_t, \mathbf{u}_t)$ when the control input $\mathbf{u}_t$ is corrupted by Gaussian noise $\mathcal{N}(0, \sigma_u^2)$:

$$\tilde{\mathbf{u}}_t = \mathbf{u}_t + \varepsilon_{\mathbf{u}}, \quad \varepsilon_{\mathbf{u}} \sim \mathcal{N}(0, \sigma_u^2)\,. \tag{16}$$

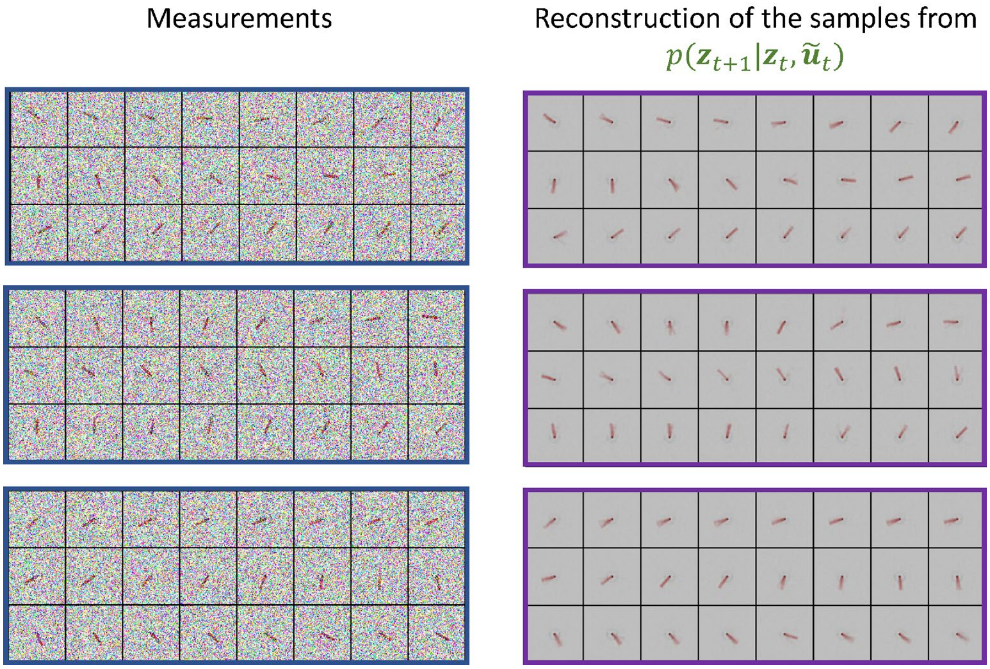
Measurements
Reconstruction of the samples from
$p(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t, \tilde{\boldsymbol{u}}_t)$

Figure 6. Reconstructions $\hat{\mathbf{x}}_{t+1}$ with different noise levels in both the measurements $\tilde{\mathbf{x}}_t$ and the control inputs $\tilde{\mathbf{u}}_t$. With both the measurements and control inputs corrupted by significant noise, the proposed model shows good performance in denoising.

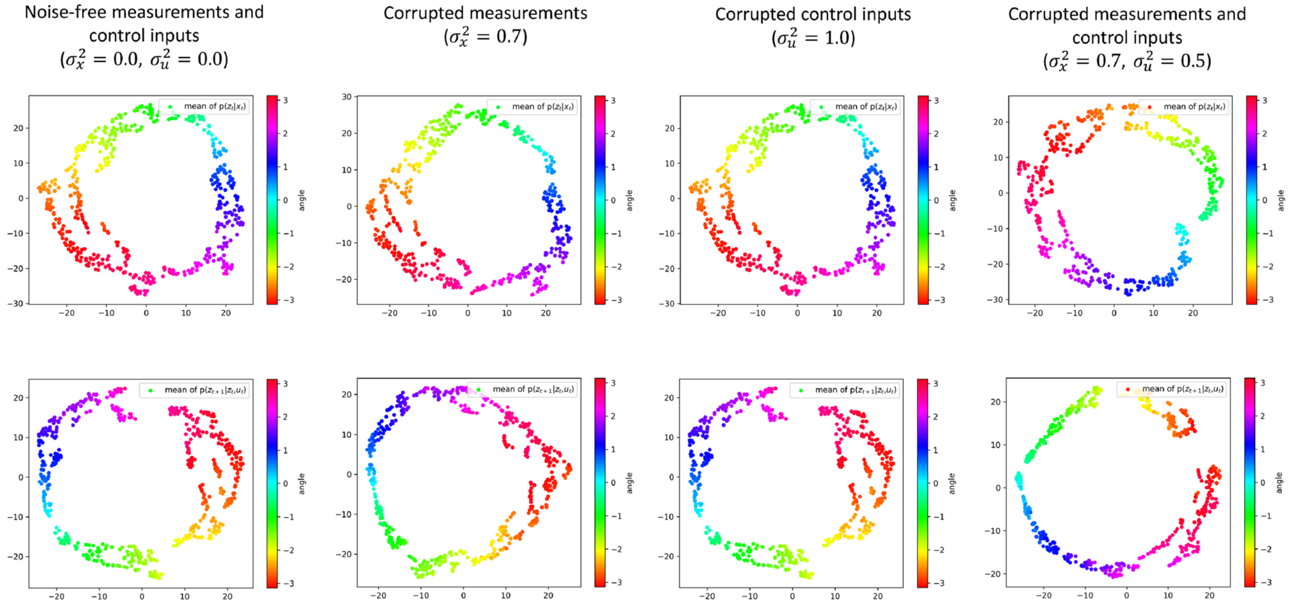

Figure 7. t-SNE visualization for the means of current (top) and next (bottom) latent state distributions with different noise levels in the measurements and control inputs. The color bar represents the true angle of the pendulum. As expected for a good denoising scheme, the change in the means of latent states is inconsiderable while the level of noise in the measurements and control inputs is increased significantly.

Eventually, in Fig. 6, we show the reconstructions when $\mathbf{x}_t$ and $\mathbf{u}_t$ are simultaneously corrupted by Gaussian noises $\mathcal{N}(0, \sigma_x^2)$ and $\mathcal{N}(0, \sigma_u^2)$, respectively.

In all the three cases, our framework can properly denoise the input measurements by encoding the predominant features into the latent space. To support this claim, we show, in Fig. 7, the means of the current and next latent state distributions with different noise corruptions. It is worth noting that the means of such distributions

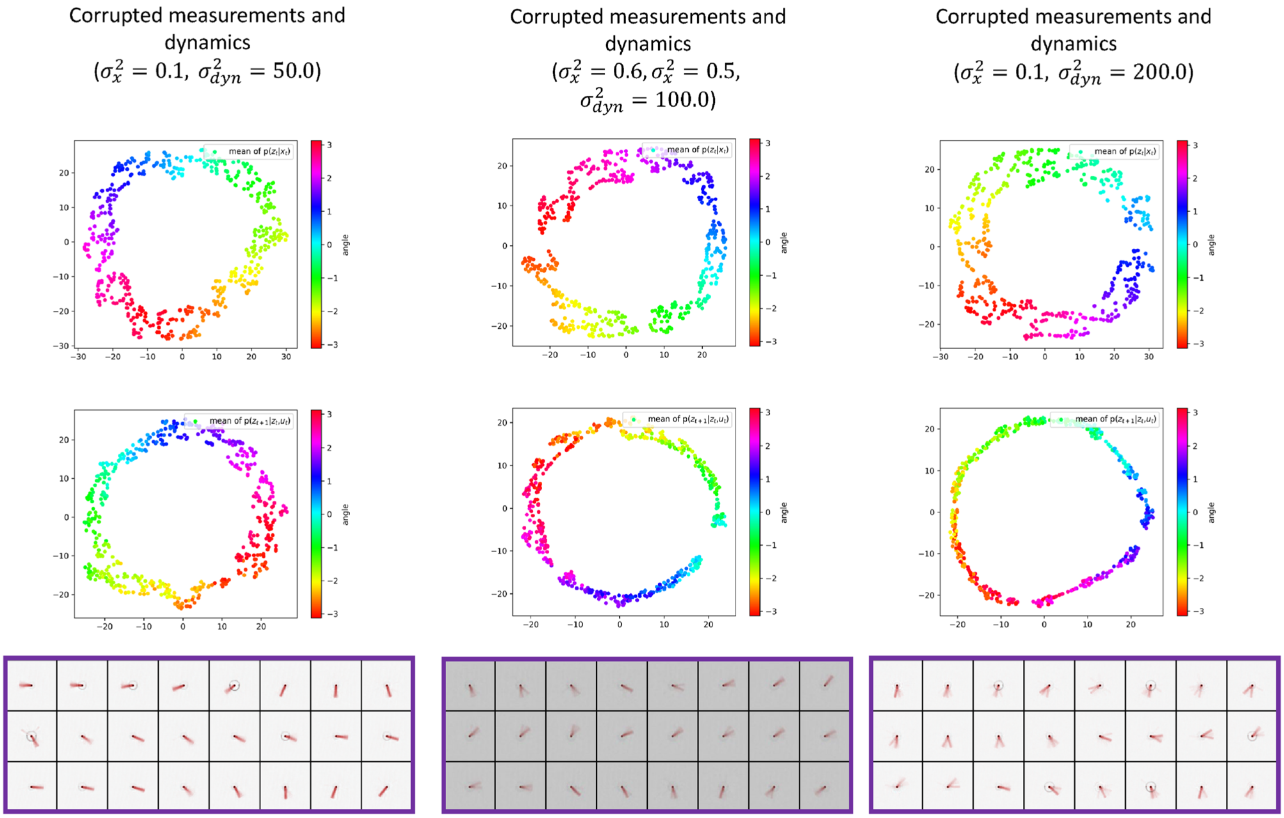

Figure 8. t-SNE visualization for the means of current and next latent state distributions with different levels of measurement noise and dynamics stochasticity, and the corresponding reconstructions $\hat{\mathbf{x}}_{t+1}$.

are a high-quality representation of the actual dynamics of the pendulum. Due to the dimensionality (> 2) of the latent state space, we use t-SNE[45] to visualize the results in 2-dimensional figures with the color bar representing the actual angle of the pendulum. The smooth change of the representation with respect to the true angle indicates its high quality. Moreover, it is worth mentioning that, as the level of noise in the measurements and control inputs is increased dramatically, the changes in the means of learned distributions are insignificant because of the denoising capability of the proposed model.

Prediction of dynamics.

To better demonstrate how well the framework performs in prediction under uncertainties, we modify the pendulum dynamics in Eq. (14) to account for stochasticity due to, for example, external disturbances:

$$\ddot{\phi}(t) = -\frac{1}{ml}(mg\sin\phi(t) + u(t) + \varepsilon_{dyn}), \quad \varepsilon_{dyn} \sim \mathcal{N}(0, \sigma_{dyn}^2). \tag{17}$$

Again, we include an independently added Gaussian noise. While $\varepsilon_{\mathbf{x}}$ and $\varepsilon_{\mathbf{u}}$ are noise terms added to the noise-free measurements $\mathbf{x}_t$ and control inputs $\mathbf{u}_t$ to model, for example, the noise deriving from the sensor devices, ε_{dyn} approximates an unknown disturbance on the actual pendulum dynamics.

In Fig. 8, we show the means of $p(\mathbf{z}_t|\mathbf{x}_t)$ and $p(\mathbf{z}_{t+1}|\mathbf{z}_t, \mathbf{u}_t)$ with different noise levels, and the corresponding (decoded) reconstructions of $\mathbf{z}_{t+1}$ samples from $p(\mathbf{z}_{t+1}|\mathbf{z}_t, \mathbf{u}_t)$. From the mean of $p(\mathbf{z}_t|\mathbf{x}_t)$, we can notice that the SVDKL encoder properly denoises the measurements and extracts the latent state variables when both measurement noise and disturbance on the actual pendulum dynamics exist. The SVDKL dynamical model recovers the mean of $p(\mathbf{z}_{t+1}|\mathbf{z}_t, \mathbf{u}_t)$ when the dynamical evolution of the pendulum is affected by an unknown stochastic disturbance. Even with high level of disturbance, though the system evolution over time becomes stochastic and more difficult to predict, $p(\mathbf{z}_{t+1}|\mathbf{z}_t, \mathbf{u}_t)$ can still capture and predict the evolution. Eventually, we can visualize the overall uncertainty in the dynamics reflected by the reconstruction of $\mathbf{z}_{t+1}$ via the decoder D. Note that the reconstructions in Fig. 8 are obtained by averaging 10 independent samples per data point.

Uncertainty quantification.

In this subsection, we show that the proposed SVDKL-AE enables the quantification of uncertainties in model predictions. It is worth mentioning that visualizing UQ properly is a commonly recognized challenging task in unsupervised learning.

The learned latent state vector $\mathbf{z}$ is 20-dimensional. To visualize the UQ capability of the proposed model, we select the ith-component ($i = 12$ and $i = 13$ in Figs. 9 and 10, respectively) of the state vector that is correlated with the physical states, and depict its predictive uncertainty bounds for different noise levels ($\sigma_x^2 = 0.0, \sigma_u^2 = 0.7$

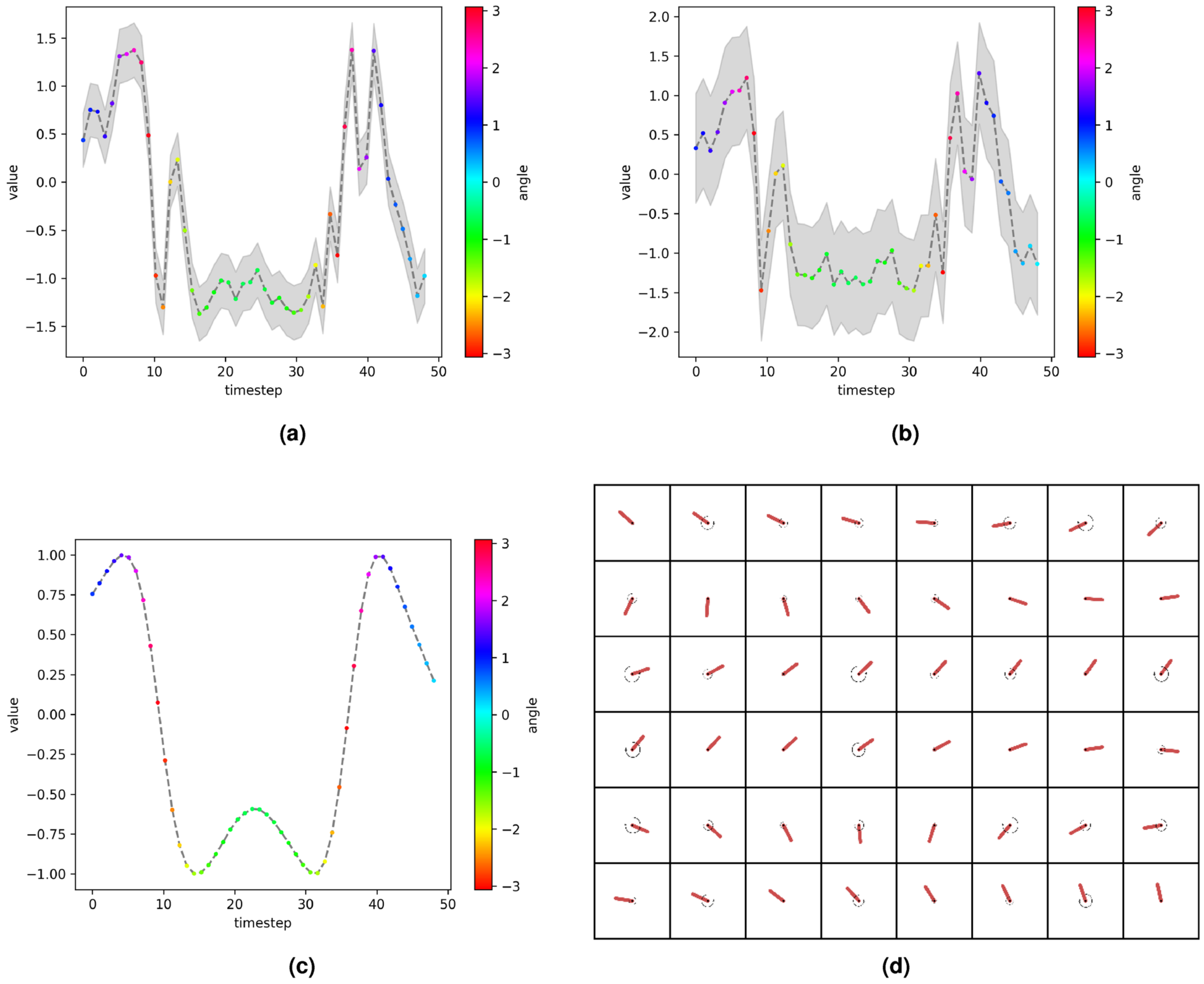

Figure 9. Estimated uncertainties over the 12th-components of the latent state vectors $\mathbf{z}_t$ (**a**) and $\mathbf{z}_{t+1}$ (**b**) predicted by the proposed model for a sampled trajectory. The y-position of the pendulum is given in (**c**), and the corresponding high-dimensional noisy measurements ($\sigma_x^2 = 0.0, \sigma_u^2 = 0.7$) are shown in (**d**). The uncertainty bands are given by ± two standard deviation in the predictive distributions.

and $\sigma_x^2 = 0.5, \sigma_u^2 = 0.5$, respectively). Because we are investigating an unsupervised learning problem, the latent variables may not have a direct physical interpretation. However, a good latent representation should present strong correlation with the physical states, and proper UQ should reflect the existence of noise in the measurements $\mathbf{x}$ and/or control inputs $\mathbf{u}$.

As seen in the figures, the proposed framework achieves a good system representation with latent variables that are highly correlated with physical quantities of interest (see Figs. 9a–c and 10a,b), devised by uncertainty bands reflecting data noise and modeling errors (see Fig. 9a,b in comparison with 10a,b.)

Discussion and future work

Though well researched in supervised learning[46], uncertainty quantification is still an understudied topic in unsupervised dimensionality reduction and latent model learning. However, the combination of these two tasks has the potential to open new doors to the discovery of governing principles of complex dynamical systems from high-dimensional noisy data. Our proposed method provides convincing indications that combining deep NNs with kernel-based models is promising for the analysis of high-dimensional noisy data. Our general framework relies only on the observations of measurements and control inputs, making it applicable to all physical modeling, digital twinning, weather forecast, and patient-specific medical analysis.

Learning compact state representations and latent dynamical models from high-dimensional noisy observations is a critical element of Optimal Control and Model-based RL. In both, the disentanglement of measurement and modeling uncertainties will play a crucial role in optimizing control laws, as well as in devising efficient exploration of the latent state space to aid the collection of new, informative samples for model improvement. The quantified uncertainties can be exploited for Active Learning[47] to steer the data sampling[48].

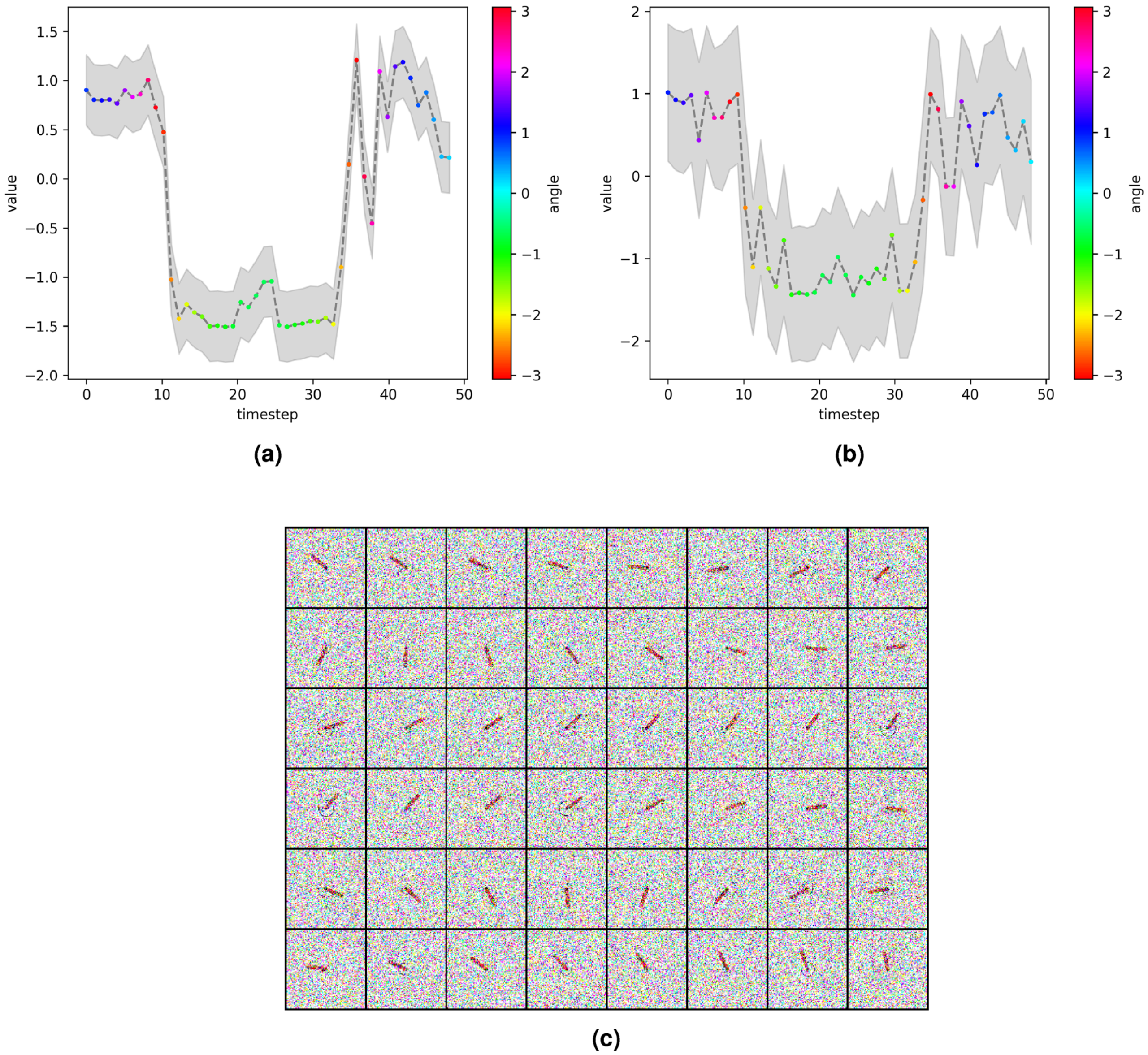

Figure 10. Estimated uncertainties over the 13th-components of the latent state vectors $\mathbf{z}_t$ (**a**) and $\mathbf{z}_{t+1}$ (**b**) predicted by the proposed model for a sampled trajectory. The corresponding high-dimensional noisy measurements ($\sigma_x^2 = 0.5, \sigma_u^2 = 0.5$) are shown in (**c**). The uncertainty bands are given by ± two standard deviation in the predictive distributions.

Conclusions

SVDKL models are integrated into a novel general workflow of unsupervised dimensionality reduction and latent dynamics learning, combining the expressive power of deep NNs with the uncertainty quantification abilities of GPs. The proposed method has shown good capability of generating interpretable latent representations and denoised reconstructions of high-dimensional, noise-corrupted measurements, see Figs. 4, 5, 6 and 7, respectively. It has also been demonstrated that this method can deal with stochastic dynamical systems by identifying the source of stochasticity.

Data availability

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Received: 30 August 2022; Accepted: 29 November 2022
Published online: 13 December 2022

References

1. Brunton, S. L. & Kutz, J. N. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control* (Cambridge University Press, 2022).
2. Mitchell, T. M. & Mitchell, T. M. *Machine Learning* Vol. 1 (McGraw-hill, New York, 1997).
3. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* (MIT Press, 2016).
4. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction* (MIT Press, 2018).
5. Arulkumaran, K., Deisenroth, M. P., Brundage, M. & Bharath, A. A. Deep reinforcement learning: A brief survey. *IEEE Signal Process. Mag.* **34**, 26–38 (2017).
6. Lesort, T., Díaz-Rodríguez, N., Goudou, J.-F. & Filliat, D. State representation learning for control: An overview. *Neural Netw.* **108**, 379–392 (2018).
7. Botteghi, N., Poel, M. & Brune, C. Unsupervised representation learning in deep reinforcement learning: A review. arXiv preprint arXiv:2208.14226 (2022).
8. Quarteroni, A. *et al. Reduced Order Methods for Modeling and Computational Reduction* Vol. 9 (Springer, Berlin, 2014).
9. Hesthaven, J. S., Pagliantini, C. & Rozza, G. Reduced basis methods for time-dependent problems. *Acta Numer* **31**, 265–345 (2022).
10. Camacho, E. F. & Alba, C. B. *Model Predictive Control* (Springer, Berlin, 2013).
11. Wall, M. E., Rechtsteiner, A., & Rocha, L. M. Singular value decomposition and principal component analysis. In *A Practical Approach to Microarray Data Analysis*, pp. 91–109 (Springer, 2003).
12. Wold, S., Esbensen, K. & Geladi, P. Principal component analysis. *Chemomet. Intell. Lab. Syst.* **2**, 37–52 (1987).
13. Berkooz, G., Holmes, P. & Lumley, J. L. The proper orthogonal decomposition in the analysis of turbulent flows. *Ann. Rev. Fluid Mech.* **25**, 539–575 (1993).
14. Schmid, P. J. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* **656**, 5–28 (2010).
15. Proctor, J. L., Brunton, S. L. & Kutz, J. N. Dynamic mode decomposition with control. *SIAM J. Appl. Dyn. Syst.* **15**, 142–161 (2016).
16. 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.* **113**, 3932–3937 (2016).
17. Ghattas, O. & Willcox, K. Learning physics-based models from data: Perspectives from inverse problems and model reduction. *Acta Numer* **30**, 445–554 (2021).
18. Guo, M., McQuarrie, S. A. & Willcox, K. E. Bayesian operator inference for data-driven reduced-order modeling. *Comput. Methods Appl. Mech. Eng.* **402,** 115336 (2022).
19. Peherstorfer, B. & Willcox, K. Data-driven operator inference for nonintrusive projection-based model reduction. *Comput. Methods Appl. Mech. Eng.* **306**, 196–215 (2016).
20. Guo, M. & Hesthaven, J. S. Data-driven reduced order modeling for time-dependent problems. *Comput. Methods Appl. Mech. Eng.* **345**, 75–99 (2019).
21. Lee, K. & Carlberg, K. T. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *J. Comput. Phys.* **404**, 108973 (2020).
22. Champion, K., Lusch, B., Kutz, J. N. & Brunton, S. L. Data-driven discovery of coordinates and governing equations. *Proc. Natl. Acad. Sci.* **116**, 22445–22451 (2019).
23. Wahlström, N., Schön, T. B. & Deisenroth, M. P. From pixels to torques: Policy learning with deep dynamical models. arXiv preprint arXiv:1502.02251 (2015).
24. Assael, J.-A. M., Wahlström, N., Schön, T. B. & Deisenroth, M. P. Data-efficient learning of feedback policies from image pixels using deep dynamical models. arXiv preprint arXiv:1510.02173 (2015).
25. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013).
26. Fraccaro, M., Kamronn, S., Paquet, U. & Winther, O. A disentangled recognition and nonlinear dynamics model for unsupervised learning. *Adv. Neural Inf. Process. Syst.* **30**, 1 (2017).
27. Krishnan, R. G., Shalit, U. & Sontag, D. Deep kalman filters. arXiv preprint arXiv:1511.05121 (2015).
28. Karl, M., Soelch, M., Bayer, J. & Van der Smagt, P. Deep variational bayes filters: Unsupervised learning of state space models from raw data. arXiv preprint arXiv:1605.06432 (2016).
29. Buesing, L. *et al.* Learning and querying fast generative models for reinforcement learning. arXiv preprint arXiv:1802.03006 (2018).
30. Doerr, A. *et al.* Probabilistic recurrent state-space models. In *International Conference on Machine Learning*, pp. 1280–1289 (PMLR, 2018).
31. Hafner, D. *et al.* Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pp. 2555–2565 (PMLR, 2019).
32. Hafner, D., Lillicrap, T., Ba, J. & Norouzi, M. Dream to control: Learning behaviors by latent imagination. arXiv preprint arXiv:1912.01603 (2019).
33. Wilson, A. G., Hu, Z., Salakhutdinov, R. & Xing, E. P. Deep kernel learning. In *Artificial Intelligence and Statistics*, pp. 370–378 (PMLR, 2016).
34. Williams, C. K. & Rasmussen, C. E. *Gaussian Processes for Machine Learning* (MIT Press Cambridge, MA, 2006).
35. Chen, T. & Chen, H. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Trans. Neural Networks* **6**, 911–917 (1995).
36. Calandra, R., Peters, J., Rasmussen, C. E. & Deisenroth, M. P. Manifold gaussian processes for regression. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 3338–3345 (IEEE, 2016).
37. Bradshaw, J., Matthews, A. G. d. G. & Ghahramani, Z. Adversarial examples, uncertainty, and transfer testing robustness in gaussian process hybrid deep networks. arXiv preprint arXiv:1707.02476 (2017).
38. Chen, X., Peng, X., Li, J.-B. & Peng, Y. Overview of deep kernel learning based techniques and applications. *J. Netw. Intell.* **1**, 83–98 (2016).
39. Ober, S. W., Rasmussen, C. E. & van der Wilk, M. The promises and pitfalls of deep kernel learning. In *Uncertainty in Artificial Intelligence*, pp. 1206–1216 (PMLR, 2021).
40. Belanche Muñoz, L. A. & Ruiz Costa-Jussà, M. Bridging deep and kernel methods. In *ESANN2017: 25th European Symposium on Artificial Neural Networks: Bruges, Belgium, April 26-27-28*, 1–10 (2017).
41. Tossou, P., Dura, B., Laviolette, F., Marchand, M. & Lacoste, A. Adaptive deep kernel learning. arXiv preprint arXiv:1905.12131 (2019).
42. Wilson, A. G., Hu, Z., Salakhutdinov, R. R. & Xing, E. P. Stochastic variational deep kernel learning. *Adv. Neural Inf. Process. Syst.* **29**, 1 (2016).
43. Kononenko, I. Bayesian neural networks. *Biol. Cybern.* **61**, 361–370 (1989).
44. Hafner, D., Lillicrap, T., Norouzi, M. & Ba, J. Mastering atari with discrete world models. arXiv preprint arXiv:2010.02193 (2020).
45. Van der Maaten, L. & Hinton, G. Visualizing data using t-sne. *J. Mach. Learn. Res.* **9**, 1 (2008).
46. Abdar, M. *et al.* A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Inf. Fus.* **76**, 243–297 (2021).
47. Settles, B. Active learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **6**, 1–114 (2012).
48. 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).

Acknowledgements

The second author acknowledges the support from Sectorplan Bèta (NL) under the focus area *Mathematics of Computational Science*.

Author contributions

N.B. and M.G. conceived the mathematical models, N.B. implemented the methods and designed the numerical experiments, N.B. and M.G. interpreted the results, N.B. wrote the first draft, and M.G. and C.B. reviewed the manuscript. All authors gave approval for the final manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-022-25362-4.

Correspondence and requests for materials should be addressed to N.B.