Variational Autoencoders for Learning Nonlinear Dynamics of Physical Systems

Ryan Lopez, Paul J. Atzberger

1 Department of Mathematics, University of California Santa Barbara (UCSB).
2 Department of Mechanical Engineering, University of California Santa Barbara (UCSB).
3 Department of Physics, University of California Santa Barbara (UCSB).
+atzberg@gmail.com
http://atzberger.org/

Abstract

We develop data-driven methods for incorporating physical information for priors to learn parsimonious representations of nonlinear systems arising from parameterized PDEs and mechanics. Our approach is based on Variational Autoencoders (VAEs) for learning nonlinear state space models from observations. We develop ways to incorporate geometric and topological priors through general manifold latent space representations. We investigate the performance of our methods for learning low dimensional representations for the nonlinear Burgers equation and constrained mechanical systems.

Introduction

The general problem of learning dynamical models from a time series of observations has a long history spanning many fields [51, 67, 15, 35] including in dynamical systems [8, 67, 68, 47, 50, 52, 32, 19, 23], control [9, 51, 60, 63], statistics [1, 48, 26], and machine learning [15, 35, 46, 58, 3, 73]. Referred to as system identification in control and engineering, many approaches have been developed starting with linear dynamical systems (LDS). These includes the Kalman Filter and extensions [39, 22, 28, 70, 71], Principle Orthogonal Decomposition (POD) [12, 49], and more recently Dynamic Mode Decomposition (DMD) [63, 45, 69] and Koopman Operator approaches [50, 20, 42]. These successful and widely-used approaches rely on assumptions on the model structure, most commonly, that a time-invariant LDS provides a good local approximation or that noise is Gaussian.

There also has been research on more general nonlinear system identification [1, 65, 15, 35, 66, 47, 48, 51]. Nonlinear systems pose many open challenges and fewer unified approaches given the rich behaviors of nonlinear dynamics. For classes of systems and specific application domains, methods have been developed which make different levels of assumptions about the underlying structure of the dynamics. Methods for learning nonlinear dynamics include the NARAX and NOE approaches with function approximators based on neural networks and other models classes [51, 67], sparse symbolic dictionary methods that are linear-in-parameters such as SINDy [9, 64, 67], and dynamic Bayesian networks (DBNs), such as Hidden Markov Chains (HMMs) and Hidden-Physics Models [58, 54, 62, 5, 43, 26].

A central challenge in learning non-linear dynamics is to obtain representations not only capable of reproducing similar outputs as observed directly in the training dataset but infer structures that can provide stable more long-term extrapolation capabilities over multiple future steps and input states. In this work, we develop learning methods aiming to obtain robust non-linear models by providing ways to incorporate more structure and information about the underlying system related to smoothness, periodicity, topology, and other constraints. We focus particularly on developing Probabilistic Autoencoders (PAEs) that incorporate noise-based regularization and priors to learn lower dimensional representations from observations. This provides the basis of nonlinear state space models for prediction. We develop methods for incorporating into such representations geometric and topological information about the system. This facilitates capturing qualitative features of the dynamics to enhance robustness and to aid in interpretability of results. We demonstrate and perform investigations of our methods to obtain models for reductions of parameterized PDEs and for constrained mechanical systems.

Learning Nonlinear Dynamics with Variational Autoencoders (VAEs)

We develop data-driven approaches based on a Variational Autoencoder (VAE) framework [40]. We learn from observation data a set of lower dimensional representations that are used to make predictions for the dynamics. In practice, data can include experimental measurements, large-scale computational simulations, or solutions of complicated dynamical systems for which we seek reduced models. Reductions aid in gaining insights for a class of inputs or physical regimes into the underlying mechanisms generating the observed behaviors. Reduced descriptions are also helpful in many optimization problems in design and in development of controllers [51].

Standard autoencoders can result in encodings that yield unstructured scattered disconnected coding points for system features. VAEs provide probabilistic encoders and decoders where noise provides regularizations that promote more connected encodings, smoother dependence on inputs,
and more disentangled feature components [40]. As we shall discuss, we also introduce other regularizations into our methods to help aid in interpretation of the learned latent representations.

![Diagram](image)

**Figure 1:** Learning Nonlinear Dynamics. Data-driven methods are developed for learning robust models to predict from \(u(x, t)\) the non-linear evolution to \(u(x, t + \tau)\) for PDEs and other dynamical systems. Probabilistic Autoencoders (PAEs) are utilized to learn representations \(z\) of \(u(x, t)\) in low dimensional latent spaces with prescribed geometric and topological properties. The model makes predictions using learnable maps that (i) encode an input \(u(x, t) \in \mathcal{U}\) as \(z(t)\) in latent space (top), (ii) evolve the representation \(z(t) \rightarrow z(t + \tau)\) (top-right), (iii) decode the representation \(z(t + \tau)\) to predict \(\hat{u}(x, t + \tau)\) (bottom-right).

We learn VAE predictors using a Maximum Likelihood Estimation (MLE) approach for the Log Likelihood (LL) \(L_{LL} = \log(p_\theta(X, x))\). For dynamics of \(u(s)\), let \(X = u(t)\) and \(x = u(t + \tau)\). We base \(p_\theta\) on the autoencoder framework in Figure 1 and 2. We use variational inference to approximate the LL by the Evidence Lower Bound (ELBO) [7] to train a model with parameters \(\theta\) using encoders and decoders based on minimizing the loss function

\[
\begin{align*}
\theta^* &= \arg\min_{\theta_e, \theta_d} -L^B(\theta_e, \theta_d, \theta_e; X^{(i)}, x^{(i)}), \\
L^B &= L_{RE} + L_{KL} + L_{RR}, \\
L_{RE} &= E_{q_\theta(z|x^{(i)})} \left[ \log p_{\theta_d}(x^{(i)}|z) \right], \\
L_{KL} &= -\beta D_{KL} \left( q_\theta(z|X^{(i)}) \parallel p_\theta(z) \right), \\
L_{RR} &= \gamma E_{q_\theta(z'|x^{(i)})} \left[ \log p_{\theta_d}(x^{(i)}|z') \right].
\end{align*}
\]

The \(q_\theta\) denotes the encoding probability distribution and \(p_\theta\), the decoding probability distribution. The loss \(\ell = -L^B\) provides a regularized form of MLE.

The terms \(L_{RE}\) and \(L_{KL}\) arise from the ELBO variational bound \(L_{LL} \geq L_{RE} + L_{KL}\) when \(\beta = 1\), [7]. This provides a way to estimate the log likelihood that the encoder-decoder reproduce the observed data sample pairs \((X^{(i)}, x^{(i)})\) using the codes \(z'\) and \(z\). Here, we include a latent-space mapping \(z' = f_\theta(z)\) parameterized by \(\theta\), which we can use to characterize the evolution of the system or further processing of features. The \(X^{(i)}\) is the input and \(x^{(i)}\) is the output prediction. For the case of dynamical systems, we take \(X^{(i)} \sim u(t)\) a sample of the initial state function \(u(t)\) and the output \(x^{(i)} \sim u(t + \tau)\) the predicted state function \(u(t + \tau)\). We discuss the specific distributions used in more detail below.

The \(L_{KL}\) term involves the Kullback-Leibler Divergence [44, 18] acting similar to a Bayesian prior on latent space to regularize the encoder conditional probability distribution so that for each sample this distribution is similar to \(p_\theta\). We take \(p_\theta = \eta(0, \sigma_\theta e)\) a multi-variate Gaussian with independent components. This serves (i) to disentangle the features from each other to promote independence, (ii) provide a reference scale and localization for the encodings \(z\), and (iii) promote parsimonious codes utilizing smaller dimensions than \(d\) when possible.

The \(L_{RR}\) term gives a regularization that promotes retaining information in \(z\) so the encoder-decoder pair can reconstruct functions. As we shall discuss, this also promotes organization of the latent space for consistency over multi-step predictions and aids in model interpretability.

We use for the specific encoder probability distributions conditional Gaussians \(z \sim q_\theta(z|x^{(i)}) = a(X^{(i)}, x^{(i)}) + \eta(0, \sigma^2 z)\) where \(\eta\) is a Gaussian with variance \(\sigma^2 z\), (i.e. \(\mathbb{E}X^{(i)}|z = a, \operatorname{Var}X^{(i)}|z = \sigma^2 z\)). One can think of the learned mean function \(a\) in the VAE as corresponding to a typical encoder \(a(X^{(i)}, x^{(i)}; \theta_e) = a(X^{(i)}; \theta_e) = z^{(i)}\) and the variance function \(\sigma^2 z = \sigma^2 z(\theta_e)\) as providing control of a noise source to further regularize the encoding. Among other
properties, this promotes connectedness of the ensemble of latent space codes. For the VAE decoder distribution, we take \( x \sim p_{\theta_d}(x|z^{(i)}) = b(z^{(i)}) + \eta(0, \sigma_z^2) \). The learned mean function \( b(z^{(i)}; \theta_d) \) corresponds to a typical decoder and the variance function \( \sigma_z^2 = \sigma^2(\theta_d) \) controls the source of regularizing noise.

The terms to be learned in the VAE framework are \((a, \sigma_e, f_a, b, \sigma_d)\) which are parameterized by \( \theta = (\theta_a, \theta_d, \theta_b) \). In practice, it is useful to treat variances \( \sigma_e \) initially as hyper-parameters. We learn predictors for the dynamics by training over samples of evolution pairs \( \{(u_{i_n}, u_{i_{n+1}})\}_{i=1}^m \), where \( i \) denotes the sample index and \( u_{i_n} = u^n(t_n) \) with \( t_n = t_0 + n\tau \) for a time-scale \( \tau \).

To make predictions, the learned models use the following stages: (i) extract from \( u(t) \) the features \( z(t) \), (ii) evolve \( z(t) \rightarrow z(t + \tau) \), (iii) predict using \( z(t + \tau) \) the \( \hat{u}(t + \tau) \), summarized in Figure 1. By composition of the latent evolution map the model makes multi-step predictions of the dynamics.

### Learning with Manifold Latent Spaces

**Roles of Non-Euclidean Geometry and Topology**

For many systems, parsimonious representations can be obtained by working with non-euclidean manifold latent spaces, such as a torus for doubly periodic systems or even non-orientable manifolds, such as a klein bottle as arises in imaging and perception studies [10]. For this purpose, we learn encoders \( E \) over a family of mappings to a prescribed manifold \( M \) of the form

\[
z = \mathcal{E}(x) = \Lambda(\mathcal{E}(x)) = \Lambda(w), \quad w = \mathcal{E}(x).
\]

We take the map \( \mathcal{E}(x) : x \rightarrow w \), where we represent a smooth closed manifold \( M \) of dimension \( m \) in \( \mathbb{R}^{2m} \), as supported by the Whitney Embedding Theorem [72]. The \( \Lambda \) maps (projects) points \( w \in \mathbb{R}^{2m} \) to the manifold representation \( z \in M \subset \mathbb{R}^{2m} \). In practice, we accomplish this two ways: (i) we provide an analytic mapping \( \Lambda \) to \( M \), (ii) we provide a high resolution point-cloud representation of the target manifold along with local gradients and use for \( \Lambda \) a quantized mapping to the nearest point on \( M \). We provide more details in Appendix A.

This allows us to learn VAEs with latent spaces for \( z \) with general specified topologies and controllable geometric structures. The topologies of sphere, torus, klein bottle are intrinsically different than \( \mathbb{R}^n \). This allows for new types of priors such as uniform on compact manifolds or distributions with more symmetry. As we shall discuss, additional latent space structure also helps in learning more robust representations less sensitive to noise since we can unburden the encoder and decoder from having to learn the embedding geometry and avoid the potential for them making erroneous use of extra latent space dimensions. We also have statistical gains since the decoder now only needs to learn a mapping from the manifold \( M \) for reconstructions of \( x \). These more parsimonious representations also aid identifiability and interpretability of models.

### Related Work

Many variants of autoencoders have been developed for making predictions of sequential data, including those based on Recurrent Neural Networks (RNNs) with LSTMs and GRUs [34, 29, 16]. While RNNs provide a rich approximation class for sequential data, they pose for dynamical systems challenges for interpretability and for training to obtain predictions stable over many steps with robustness against noise in the training dataset. Autoencoders have also been combined with symbolic dictionary learning for latent dynamics in [11] providing some advantages for interpretability and robustness, but require specification in advance of a sufficiently expressive dictionary. Neural networks incorporating physical information have also been developed that impose stability conditions during training [53, 46, 24]. The work of [17] investigates combining RNNs with VAEs to obtain more robust models for sequential data and considered tasks related to processing speech and handwriting.

In our work we learn dynamical models making use of VAEs to obtain probabilistic encoders and decoders between euclidean and non-euclidean latent spaces to provide additional regularizations to help promote parsimoniousness, disentanglement of features, robustness, and interpretability. Prior VAE methods used for dynamical systems include [31, 55, 27, 13, 55, 59]. These works use primarily euclidean latent spaces and consider applications including human motion capture and ODE systems. Approaches for incorporating topological information into latent variable representations include the early works by Kohonen on Self-Organizing Maps (SOMs) [41] and Bishop on Generative Topographical Maps (GTMs) based on density networks providing a generative approach [6]. More recently, VAE methods using non-euclidean latent spaces include [37, 38, 25, 14, 21, 2]. These incorporate the role of geometry by augmenting the prior distribution \( p_{\theta_d}(z) \) on latent space to bias toward a manifold. In the recent work [57], an explicit projection procedure is introduced, but in the special case of a few manifolds having an analytic projection map.

In our work we develop further methods for more general latent space representations, including non-orientable manifolds, and applications to parameterized PDEs and constrained mechanical systems. We introduce more general methods for non-euclidean latent spaces in terms of point-cloud representations of the manifold along with local gradient information that can be utilized within general backpropagation frameworks, see Appendix A. This also allows for the case of manifolds that are non-orientable and having complex shapes. Our methods provide flexible ways to design and control both the topology and the geometry of the latent space by merging or subtracting shapes or stretching and contracting regions. We also consider additional types of regularizations for learning dynamical models facilitating multi-step predictions and more interpretable state space models. In our work, we also consider reduced models for non-linear PDEs, such as Burgers Equations, and learning representations for more general constrained mechanical systems. We also investigate the role of non-linearities making comparisons with other data-driven models.
Results

Burgers’ Equation of Fluid Mechanics: Learning Nonlinear PDE Dynamics

We consider the nonlinear viscous Burgers’ equation

\[ u_t = -u u_x + \nu u_{xx}, \]  

where \( \nu \) is the viscosity [4, 36]. We consider periodic boundary conditions on \( \Omega = [0, 1] \). Burgers equation is motivated as a mechanistic model for the fluid mechanics of advective transport and shocks, and serves as a widely used benchmark as a mechanistic model for the fluid mechanics of advective transport and shocks, and serves as a widely used benchmark analysis and computational methods.

The nonlinear Cole-Hopf Transform \( \mathcal{C} \mathcal{H} \) can be used to relate Burgers equation to the linear Diffusion equation \( \phi_t = \nu \phi_{xx} \) [36]. This provides a representation of the solution \( u \)

\[ \phi(x, t) = \mathcal{C} \mathcal{H}[u] = \exp \left( -\frac{1}{2\nu} \int_0^x u(x', t) \, dx' \right) \]

\[ u(x, t) = \mathcal{C} \mathcal{H}^{-1}[\phi] = -2\nu \frac{\partial}{\partial x} \ln \phi(x, t). \]  

This can be represented by the Fourier expansion

\[ \phi(x, t) = \sum_{k = -\infty}^{\infty} \hat{\phi}_k(0) \exp(-4\pi^2 k^2 \nu t) \cdot \exp(i2\pi kx). \]

The \( \hat{\phi}_k(0) = \mathcal{F}_k[\phi(x, 0)] \) and \( \phi(x, t) = \mathcal{F}^{-1}[\{\hat{\phi}_k(0) \exp(-4\pi^2 k^2 \nu t)\}] \) with \( \mathcal{F} \) the Fourier transform. This provides an analytic representation of the solution of the viscous Burgers equation \( u(x, t) = \mathcal{C} \mathcal{H}^{-1}[\phi(x, t)] \) where \( \phi(0) = \mathcal{F}[\mathcal{C} \mathcal{H}[u(x, 0)]] \). In general, for nonlinear PDEs with initial conditions within a class of functions \( \mathcal{U} \), we aim to learn models that provide predictions \( u(t + \tau) = S_\tau u(t) \) approximating the evolution operator \( S_\tau \) over time-scale \( \tau \). For the Burgers equation, the \( \mathcal{C} \mathcal{H} \) provides an analytic way to obtain a reduced order model by truncating the Fourier expansion to \( |k| \leq n_f/2 \). This provides for the Burgers equation a benchmark model against which to compare our learned models. For general PDEs comparable analytic representations are not usually available, motivating development of data-driven approaches.

We develop VAE methods for learning reduced order models for the responses of nonlinear Burgers Equation when the initial conditions are from a collection of functions \( \mathcal{U} \). We learn VAE models that extract from \( u(x, t) \) latent variables \( z(t) \) to predict \( u(x, t + \tau) \). Given the non-uniqueness of representations and to promote interpretability of the model, we introduce the inductive bias that the evolution dynamics in latent space for \( z \) is linear of the form \( \dot{z} = -\lambda_0 z \), giving exponential decay rate \( \lambda_0 \). For discrete times, we take \( z_{n+1} = f_{\theta_\gamma}(z_n) = \exp(-\lambda_0 \tau) \cdot z_n \), where \( \theta_\gamma = (\alpha_0) \). We still consider general nonlinear mappings for the encoders and decoders which are represented by deep neural networks. We train the model on the pairs \( (u(x, t), u(x, t + \tau)) \) by drawing \( m \) samples of \( u^i(x, t_i) \in \mathcal{S}_i \mathcal{U} \) which generates the evolved state under Burgers equation \( u^i(x, t_i + \tau) \) over time-scale \( \tau \). We perform VAE studies with parameters \( \nu = 2 \times 10^{-2}, \tau = 2.5 \times 10^{-1} \) with VAE Deep Neural Networks (DNNs) with layer sizes (in)-400-400-(out), ReLU activations, and \( \gamma = 0.5, \beta = 1 \), and initial standard deviations \( \sigma_d = \sigma_e = 4 \times 10^{-3} \). We show results of our VAE model predictions in Figure 3 and Table 1.

Burgers’ Equation: \( u_t = -u u_x + \nu u_{xx} \)

Table 1. Results

<table>
<thead>
<tr>
<th>( \alpha )</th>
<th>u(x,0)</th>
<th>u(x,t)</th>
<th>prediction u(x,t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.50</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3: Burgers’ Equation: Prediction of Dynamics. We consider responses for \( \mathcal{U}_1 = \{u | u(x, t; \alpha) = \alpha \sin(2\pi x) + (1-\alpha) \cos^3(2\pi x) \} \). Predictions are made for the evolution \( u \) over the time-scale \( \tau \) satisfying equation 2 with initial conditions in \( \mathcal{U}_1 \). We find our nonlinear VAE methods are able to learn with 2 latent dimensions the dynamics with errors < 1%. Methods such as DMD [63, 69] with 3 modes which are only able to use a single linear space to approximate the initial conditions and prediction encounter challenges in approximating the nonlinear evolution. We find our linear VAE method with 2 modes provides some improvements, by allowing for using different linear spaces for representing the input and output functions, but at the cost of additional computations. Results are summarized in Table 1.

We show the importance of the non-linear approximation properties of our VAE methods in capturing system behaviors by making comparisons with Dynamic Mode Decomposition (DMD) [63, 69], Principle Orthogonal Decomposition (POD) [12], and a linear variant of our VAE approach. Recent CNN-AEs have also studied related advantages of non-linear approximations [46]. Some distinctions in our work is the use of VAES to further regularize AEs and using topological latent spaces to facilitate further capturing of structure. The DMD and POD are widely used and successful approaches that aim to find an optimal linear space on which to project the dynamics and learn a linear evolution law for system behaviors. DMD and POD have been successful in obtaining models for many applications, including steady-state fluid mechanics and transport problems [69, 63]. However, given their inherent linear approximations they can encounter well-known challenges related to translational and rotational invariances, as arise in advective phenomena and other settings [8]. Our comparison studies can be found in...
Figure 4: Burgers’ Equation: Latent Space Representations and Extrapolation Predictions. We show the latent space representation \( z \) of the dynamics for the input functions \( u(\cdot, t; \alpha) \in \mathcal{U}_1 \). VAE organizes for \( u \) the learned representations \( z(\alpha, t) \) in parameter \( \alpha \) (blue-green) into circular arcs that are concentric in the time parameter \( t \), (yellow-orange) (left). The reconstruction regularization with \( \gamma \) aligns subsequent time-steps of the dynamics in latent space facilitating multi-step predictions. The learned VAE model exhibits a level of extrapolation to predict dynamics even for some inputs \( u \not\in \mathcal{U}_1 \) beyond the training dataset (right).

Table 1. We also considered how our VAE methods performed when adjusting the parameters \( \beta \) for the strength of the prior \( \tilde{p} \) as in \( \beta \)-VAEs [33] and \( \gamma \) for the strength of the reconstruction regularization. The reconstruction regularization has a significant influence on how the VAE organizes representations in latent space and the accuracy of predictions of the dynamics, especially over multiple steps, see Figure 4 and Table 1. The regularization serves to align representations consistently in latent space facilitating multi-step compositions. We also found our VAE learned representations capable of some level of extrapolation beyond the training dataset. When varying \( \beta \), we found that larger values improved the multiple step accuracy whereas small values improved the single step accuracy, see Table 1.

Constrained Mechanics: Learning with Non-Euclidean Latent Spaces

To learn more parsimonious and robust representations of physical systems, we develop methods for latent spaces having geometries and topologies more general than euclidean space. This is helpful in capturing inherent structure such as periodicities or other symmetries. We consider physical systems with constrained mechanics, such as the arm mechanism for reaching for objects in figure 5. The observations are taken to be the two locations \( x_1, x_2 \in \mathbb{R}^2 \) giving \( x = (x_1, x_2) \in \mathbb{R}^4 \). When the segments are rigidly constrained these configurations lie on a manifold (torus). We can also allow the segments to extend and consider more exotic constraints such as the two points \( x_1, x_2 \) must be on a klein bottle in \( \mathbb{R}^4 \). Related situations arise in other areas of imaging and mechanics, such as in pose estimation and in studies of visual perception [56, 10, 61]. For the arm mechanics, we can use this prior knowledge to construct a torus latent space represented by the product space of two circles \( S^1 \times S^1 \). To obtain a learnable class of manifold encoders, we use the family of maps \( \tilde{\mathcal{E}}_\theta = \Lambda(\tilde{\mathcal{E}}_\theta(x)) \), with \( \tilde{\mathcal{E}}_\theta(x) \) into \( \mathbb{R}^4 \) and \( \Lambda(w) = (w_1, w_2, w_3, w_4) = (z_1, z_2, z_3, z_4) = z \), where \( (z_1, z_2) = (w_1, w_2)/(\|w_1, w_2\|) \), \( (z_3, z_4) = (w_3, w_4)/(\|w_3, w_4\|) \), see VAE Section and Appendix A. For the case of klein bottle constraints, we use our point-cloud representation of the non-orientable manifold with the parameterized embedding in \( \mathbb{R}^4 \)

\[
\begin{align*}
z_1 &= (a + b \cos(u_2)) \cos(u_1) \\
z_2 &= (a + b \cos(u_2)) \sin(u_1) \\
z_3 &= b \sin(u_2) \cos\left(\frac{\pi}{2}\right)
\end{align*}
\]

with \( u_1, u_2 \in [0, 2\pi] \). The \( \Lambda(w) \) is taken to be the map to the nearest point of the manifold \( \mathcal{M} \), which we compute numerically along with the needed gradients for backpropogation as discussed in Appendix A.

Our VAE methods are trained with encoder and decoder DNN’s having layers of sizes (in)-100-500-100-(out) with Leaky-ReLU activations with \( s = 1-6 \) with results reported in Figure 5 and Table 2. We find learning representations is improved by use of the manifold latent spaces, in these trials even showing a slight edge over \( \mathbb{R}^4 \). When the wrong
Figure 5: VAE Representations of Motions using Manifold Latent Spaces. We learn from observations representations for constrained mechanical systems using general non-Euclidean manifolds latent spaces $\mathcal{M}$. The arm mechanism has configurations $x = (x_1, x_2) \in \mathbb{R}^4$. For rigid segments, the motions are constrained to be on a manifold (torus) $\mathcal{M} \subset \mathbb{R}^4$. For extendable segments, we can also consider more exotic constraints, such as requiring $x_1, x_2$ to be on a klein bottle in $\mathbb{R}^4$ (top). Results of our VAE methods for learned representations for motions under these constraints are shown. VAE learns the segment length constraint and two nearly decoupled coordinates for the torus dataset that mimic the roles of angles. VAE learns for the klein bottle dataset two segment motions to generate configurations (middle and bottom).

topology is used, such as in $\mathbb{R}^2$, we find in both cases a significant deterioration in the reconstruction accuracy, see Table 2. This arises since the encoder must be continuous and hedge against the noise regularizations. This results in an incurred penalty for a subset of configurations. The encoder exhibits non-injectivity and a rapid doubling back over the space to accommodate the decoder by lining up nearby configurations in the topology of the input space manifold to handle noise perturbations in $z$ from the probabilistic nature of the encoding. We also studied robustness when training with noise for $\hat{X} = X + \sigma \eta(0, 1)$ and measuring accuracy for reconstruction relative to target $X$. As the noise increases, we see that the manifold latent spaces improve reconstruction accuracy acting as a filter through restricting the representation. The probabilistic decoder will tend to learn to estimate the mean over samples of a common underlying configuration and with the manifold latent space.

Table 2: Manifold Latent Variable Model: VAE Reconstruction Errors The $L^2$-relative errors of reconstruction for our VAE methods. The final is the lowest value during training. The manifold latent spaces show improved learning. When an incompatible topology is used, such as $\mathbb{R}^2$, this can result in deterioration in learned representations. With noise in the input $\hat{X} = X + \sigma \eta(0, 1)$ and reconstructing the target $X$, the manifold latent spaces also show improvements for learning.

Conclusions

We developed VAE’s for learning robustly nonlinear dynamics of physical systems by introducing methods for latent representations utilizing general geometric and topological structures. We demonstrated our methods for learning the non-linear dynamics of PDEs and constrained mechanical systems. We expect our methods can also be used in other physics-related tasks and problems to leverage prior geometric and topological knowledge for improving learning for nonlinear systems.

Acknowledgments

Authors research supported by grants DOE Grant ASCR PHILMS DE-SC0019246 and NSF Grant DMS-1616353. Also to R.N.L. support by a donor to UCSB CCS SURF program. Authors also acknowledge UCSB Center for Scientific Computing NSF MRSEC (DMR1121053) and UCSB MRL NSF CNS-1725797. P.J.A. would also like to acknowledge a hardware grant from Nvidia.
References


Appendix A: Backpropogation of Encoders for Non-Euclidean Latent Spaces given by General Manifolds

We develop methods for using backpropogation to learn encoder maps from $\mathbb{R}^d$ to general manifolds $\mathcal{M}$. We perform learning using the family of manifold encoder maps of the form $\mathcal{E}_0 = \Lambda(\hat{\mathcal{E}}_0(x))$. This allows for use of latent spaces having general topologies and geometries. We represent the manifold as an embedding $\mathcal{M} \subset \mathbb{R}^{2m}$ and computationally use point-cloud representations along with local gradient information, see Figure 6. To allow for $\mathcal{E}_0$ to be learnable, we develop approaches for incorporating our maps into general backpropogation frameworks.

For a manifold $\mathcal{M}$ of dimension $m$, we can represent it by an embedding within $\mathbb{R}^{2m}$, as supported by the Whitney Embedding Theorem [72]. We let $z = \Lambda(w)$ be a mapping $w \in \mathbb{R}^{2m}$ to points on the manifold $z \in \mathcal{M}$. This allows for learning within the family of manifold encoders $w = \mathcal{E}_0(x)$ any function from $\mathbb{R}^d$ to $\mathbb{R}^{2m}$. This facilitates use of deep neural networks and other function classes. In practice, we shall take $z = \Lambda(w)$ to map to the nearest location on the manifold. We can express this as the optimization problem

$$z^* = \arg \min_{z \in \mathcal{M}} \frac{1}{2} \|w - z\|^2.$$  

We can always express a smooth manifold using local coordinate charts $\sigma^k(w)$, for example, by using a local Monge-Gauge quadratic fit to the point cloud [30]. We can express $z^* = \sigma^k(u^*)$ for some chart $k^*$. In terms of the coordinate charts $\{U_k\}$ and local parameterizations $\{\sigma^k(u)\}$ we can express this as

$$u^*, k^* = \arg \min_{k, u \in U_k} \frac{1}{2} \|w - \sigma^k(u)\|^2,$$

where $\Phi_k(u, w) = \frac{1}{2} \|w - \sigma^k(u)\|^2$. The $w$ is the input and $u^*, k^*$ is the solution sought. For smooth parameterizations, the optimal solution satisfies

$$G = \nabla_z \Phi_k(u^*, w) = 0.$$

During learning we need gradients $\nabla_w \Lambda(w) = \nabla_w z$ when $w$ is varied characterizing variations of points on the manifold $z = \Lambda(w)$. We derive these expressions by considering variations $w = w(\gamma)$ for a scalar parameter $\gamma$. We can obtain the needed gradients by determining the variations of $u^* = u^*(\gamma)$. We can express these gradients using the Implicit Function Theorem as

$$0 = \frac{d}{d\gamma} \nabla(u^*(\gamma), w(\gamma)) = \nabla_u G \frac{du^*}{d\gamma} + \nabla_w G \frac{dw}{d\gamma}.$$  

This implies

$$\frac{du^*}{d\gamma} = -\left[\nabla_u G\right]^{-1} \nabla_w G \frac{dw}{d\gamma}.$$  

As long as we can evaluate at $u$ these local gradients $\nabla_u G, \nabla_w G, \frac{dw}{d\gamma}$, we only need to determine computationally the solution $u^*$. For the backpropogation framework, we use these to assemble the needed gradients for our manifold encoder maps $\mathcal{E}_0 = \Lambda(\hat{\mathcal{E}}_0(x))$ as follows.

We first find numerically the closest point in the manifold $z^* \in \mathcal{M}$ and represent it as $z^* = \sigma(u^*) = \sigma^{k^*}(u^*)$ for some chart $k^*$. In this chart, the gradients can be expressed as

$$G = \nabla_u \Phi(u, w) = -(w - \sigma(u))^T \nabla_u \sigma(u).$$  

We take here a column vector convention with $\nabla_u \sigma(u) = [\sigma_{u_1} \ldots \sigma_{u_k}]$. We next compute

$$\nabla_u G = \nabla_{uu} \Phi = \nabla_u \sigma^T \nabla_u \sigma - (w - \sigma(u))^T \nabla_{uu} \sigma(u)$$

and

$$\nabla_w G = \nabla_{ww} \Phi = -I \nabla_u \sigma(u).$$

For implementation it is useful to express this in more detail component-wise as

$$[G]_i = -\sum_k \left( \frac{w_k - \sigma_k(u)}{\partial w_i, \sigma_k(u)} \right),$$

with

$$[\nabla_u G]_{i, j} = [\nabla_{uu} \Phi]_{i, j} = \sum_k \left( \frac{\partial_{u_j} \sigma_k(u)}{\partial_{u_i} \sigma_k(u)} \right) \partial_{u_i}^2 \sigma_k(u) - \sum_k \left( \frac{w_k - \sigma_k(u)}{\partial_{u_i} \sigma_k(u)} \right) \partial_{u_i}^2 \sigma_k(u)$$

and

$$[\nabla_{ww} \Phi]_{i, j} = \sum_k \partial_{w_k} \partial_{w_i} \sigma_k(u) = -\partial_{u_i} \sigma_j(u).$$

The final gradient is given by

$$\frac{d\Lambda(w)}{d\gamma} = \frac{d\hat{z}^*}{d\gamma} = \nabla_u \sigma \frac{du^*}{d\gamma} = -\nabla_u \sigma \left[\nabla_u G\right]^{-1} \nabla_w G \frac{dw}{d\gamma}.$$  

In summary, once we determine the point $z^* = \Lambda(w)$ we need only evaluate the above expressions to obtain the needed gradient for learning via backpropogation

$$\nabla_0 \mathcal{E}_0(x) = \nabla_w \Lambda(w) \nabla_0 \hat{\mathcal{E}}_0(x), \ w = \hat{\mathcal{E}}_0(x).$$

Figure 6: Learnable Mappings to Manifold Surfaces We develop methods based on point cloud representations embedded in $\mathbb{R}^n$ for learning latent manifold representations having general geometries and topologies.
The $\nabla_w \Lambda$ is determined by $d\Lambda(w)/d\gamma$ using $\gamma = w_1, \ldots, w_n$. In practice, the $\tilde{E}_\theta(x)$ is represented by a deep neural network from $\mathbb{R}^d$ to $\mathbb{R}^{2m}$. In this way, we can learn general encoder mappings $\tilde{E}_\theta(x)$ from $x \in \mathbb{R}^d$ to general manifolds $\mathcal{M}$. 