Extracting Interpretable Physical Parameters from Partial Differential Equations using Unsupervised Learning

Peter Y. Lu[1] lup@mit.edu

Samuel Kim[2] samkim@mit.edu

Marin Soljačić[1] soljacic@mit.edu

[1]Department of Physics
[2]Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139

1 Introduction

Experimental data often contains many uncontrolled parameters that make analysis and interpretation difficult. Modern machine learning methods are particularly well-suited for data analysis, but, to be effective in science, the results need to be interpretable. Using physical systems described by partial differential equations (PDEs), we demonstrate the use of unsupervised learning techniques for extracting interpretable physical parameters from unlabeled time-series data.

Recently, there has been significant interest in the data-driven analysis and discovery of PDEs [1, 2, 3, 4, 5]. However, previous work on PDE discovery and parameter extraction often assume the entire dataset is governed by the same dynamics and explicitly provide the key parameters to be determined. In more complex scenarios, we have limited control over the systems that we are studying and yet still want to model them and extract relevant physical features. To do this, we must identify important model parameters that are uncontrolled and may vary in the raw data, producing very different behaviors. Recent work on learning parametric PDEs has taken steps toward addressing this issue [6]. We propose a model architecture (Fig. 1) based on variational autoencoders (VAEs) [7]. While similar architectures have been proposed for physical systems such as interacting particles [8] and moving objects [9], our model is specifically designed to study phenomena described by PDEs, which have a continuous set of degrees of freedom. In our numerical experiments, we examine the 2D convection–diffusion equation with a varying diffusion constant and flow velocity, the 2D incompressible Navier–Stokes equation with a varying viscosity, the 1D nonlinear Schrödinger equation with a varying nonlinearity coefficient, and the 1D Kuramoto–Sivashinsky equation with a varying viscosity damping parameter.

We demonstrate that our VAE-based architecture can accurately identify the number of relevant varying physical parameters and extract them from PDEs by constructing a flexible predictive model. We further show that our parameter extraction method is robust to noisy data and can still be effective for chaotic systems where accurate prediction is difficult.

2 Model Architecture

Our model (Fig. 1) has an encoder–decoder architecture based on variational autoencoders (VAEs) [7], which allows us to extract interpretable latent parameters that parameterize the dynamics of a system. The model requires time-series data that are grouped in pairs $(\{x_t\}, \{y_t\})$; each pair of time-series must follow the same dynamics. For real datasets, this can be constructed by taking pairs measurements in quick succession, splitting a long time-series into a pair of shorter time-series, or repeating the same time-series twice to form a pair $(\{x_t\}, \{x_t\})$. We will refer to the first time-series as the *input series* $\{x_t\}_{t=0}^{T_x}$ and the second time-series as the *target series* $\{y_t\}_{t=0}^{T_y}$.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

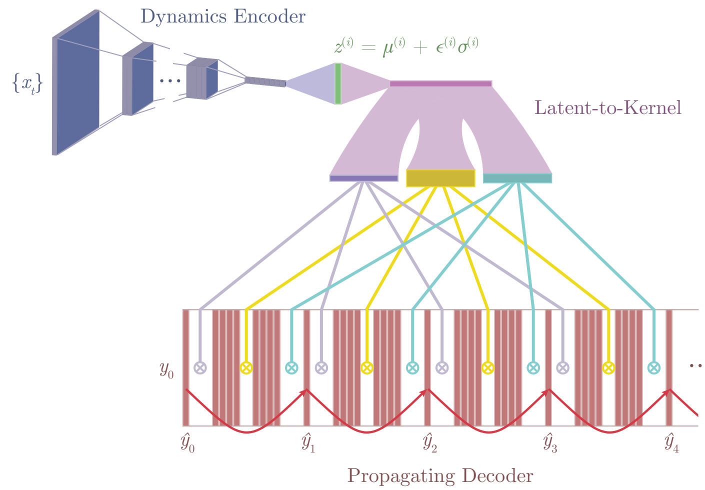


Figure 1: VAE-based model architecture consisting of a dynamics encoder (DE) and a propagating decoder (PD) with kernels given by a latent-to-kernel (L2K) network. The DE extracts latent parameters $z^{(i)}$ from the input series $\{x_t\}$. The L2K network then maps the $z^{(i)}$ to convolution kernels, which are used by the PD to predict PDE propagation $\{\hat{y}_t\}$ from an initial condition y_0.

The input series is first fed into a *dynamics encoder* (DE) network which uses a series of convolutional layers to extract the latent parameters $z^{(i)}$. During training, each $z^{(i)}$ is sampled using the VAE reparameterization trick. These parameters $z^{(i)}$ along with an initial condition—the first state y_0 in the target series—are then used by the *propagating decoder* (PD) network to predict the full target series. The PD recurrently applies a series of dynamic convolutional layers [10] with kernels parameterized by the latent parameters $z^{(i)}$. By providing the PD with an initial condition, we allow the the DE to focus on encoding parameters that describe the dynamics of the data rather encoding a particular state of the system.

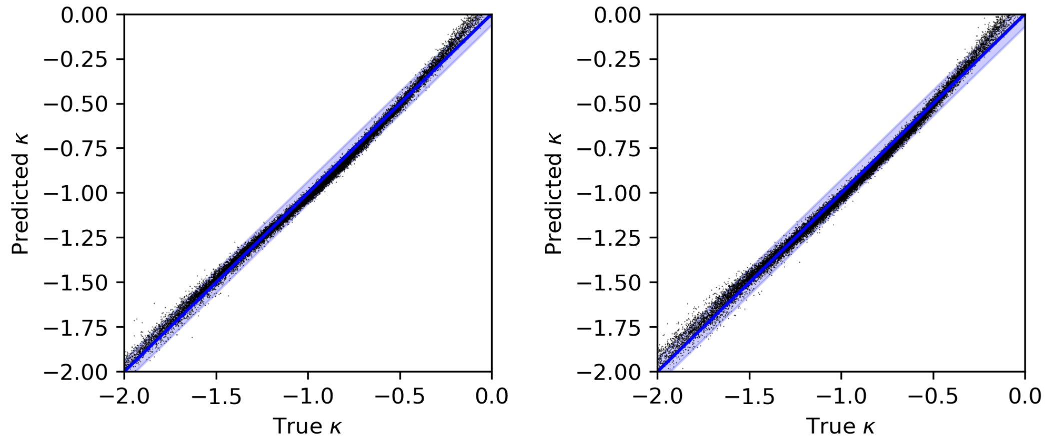
The model is trained end-to-end using a mean-squared error loss between the PD predicted series $\{\hat{y}_t\}$ and the target series $\{y_t\}$ in addition to VAE regularization loss. By using the VAE sampling method and regularizer, we compel the model to learn independent and interpretable latent parameters.

3 Parameter Extraction Experiments

For parameter extraction, we train a model with 3 latent parameters on each of our PDE datasets and apply principal component analysis (PCA). We then evaluate on our test set and use linear regression to measure the correlation of the extracted PCA components with the true parameters used to generate each dataset. Our results show that PCA components with high explained variance ratios (EVRs) tend to correlate well with physically interpretable parameters. We also find these correlations tend to be linear with high R^2 coefficients (often > 0.9), validating our use of linear regression. As an example of parameter prediction from the extracted PCA components, see Fig. 2.

For the nonlinear Schrödinger dataset, the multi-parameter model correctly determines that only the first PCA component—which is well correlated with the true nonlinearity parameter κ—is relevant (Table 1). We obtain similar results for the Navier–Stokes and Kuramoto–Sivashinsky datasets. For the convection—diffusion dataset, the three extracted PCA components correlate well with the three varying parameters: the x and y velocity components and the diffusion constant (Table 2).

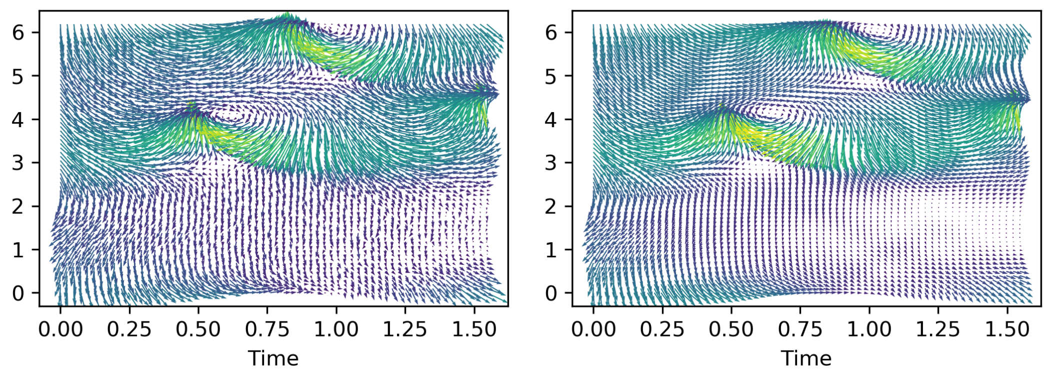
On all of the PDE datasets, our model consistently shows a robustness to noise. We test this by adding Gaussian noise with standard deviation $\sigma = 0.1$ to the generated data, which have initial conditions normalized to unit variance. Despite the roughly 10% added noise, our architecture is still able to extract relevant physical parameters with no significant loss in accuracy. The prediction performance of the model is also maintained, where the predicted PDE propagation corresponds to a denoised version of the original noisy dataset example (Fig. 3).



(a) No Noise

(b) $\sigma = 0.1$ Noise

Figure 2: Predicted nonlinearity parameter κ from a linear fit of the first extracted PCA component (Table 1) vs. true κ from the nonlinear Schrödinger dataset using the multi-parameter extraction (3 latent parameters) model. The light blue shaded region shows the 95% confidence interval of the linear regression. Results are shown for (a) the original noiseless dataset as well as (b) with added $\sigma = 0.1$ Gaussian noise.



(a) Dataset Example

(b) Predicted Propagation

Figure 3: Prediction example using the model trained on the 1D nonlinear Schrödinger dataset with $\sigma = 0.1$ added Gaussian noise. (a) shows the original noisy time-series from the dataset, and (b) shows the PDE propagation predicted by the model from the initial condition at time 0.

	No Noise		$\sigma = 0.1$ Noise	
Comp. #	EVR	R^2 (κ)	EVR	R^2 (κ)
1	**0.999**	**0.995**	**0.987**	**0.993**
2	0.001	0.002	0.013	0.002
3	0.000	0.000	0.000	0.001

Table 1: Explained variance ratios (EVRs) for each PCA component, and R^2 correlation coefficients for linear fits of the extracted PCA components with the true nonlinearity coefficient κ in the nonlinear Schrödinger dataset. Bold EVRs indicate relevant PCA components, and a bold R^2 value represents the largest correlation of κ with one of the PCA components. Results are shown for the original noiseless dataset as well as with added $\sigma = 0.1$ Gaussian noise.

	No Noise				$\sigma = 0.1$ Noise			
Comp. #	EVR	R^2 (D)	R^2 (v_x)	R^2 (v_y)	EVR	R^2 (D)	R^2 (v_x)	R^2 (v_y)
1	**0.48**	0.01	**0.91**	0.06	**0.48**	0.00	0.02	**0.95**
2	**0.40**	0.00	0.04	**0.90**	**0.44**	0.00	**0.95**	0.01
3	**0.12**	**0.74**	0.01	0.00	**0.07**	**0.68**	0.00	0.01

Table 2: Explained variance ratios (EVRs) for each PCA component, and R^2 correlation coefficients for linear fits of the extracted PCA components with the true parameters (the diffusion constant D, and the velocity components v_x, v_y) in the convection–diffusion dataset. Bold EVRs indicate relevant PCA components, and a bold R^2 value represents the largest correlation of a particular true parameter with one of the PCA components.

4 Conclusion

The numerical experiments show that our VAE-based architecture for parameter extraction from PDE systems performs well on a variety of different 1D and 2D PDE datasets. The model can accurately identify relevant parameters and extract them from raw and even noisy PDE data (with roughly 10% added noise). These extracted parameters correlate well (in most cases, linearly with $R^2 > 0.9$) with the true parameters used to generate the datasets. Our method for discovering interpretable latent parameters in PDE systems will allow us to better analyze and understand real-world phenomena and datasets, which often have unknown and uncontrolled parameters that change the system dynamics and cause varying behaviors that are difficult to disentangle. In the future, we will study alternative geometries and boundary conditions, deal with spatial inhomogeneity and time-dependent parameters, consider 3D systems, and work with latent variable predictive models for datasets with incomplete information. These advances will allow us to apply our model to more complex applications.

Acknowledgments

This research is sponsored in part by the Army Research Office and was accomplished under Cooperative Agreement Number W911NF-18-2-0048. This work is also supported by the U.S. Department of Defense through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program and by the MIT–SenseTime Alliance on Artificial Intelligence.

References

[1] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. PDE-Net: Learning PDEs from Data, 2017.

[2] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, April 2017.

[3] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations, 2017.

[4] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics Informed Deep Learning (Part II): Data-driven Discovery of Nonlinear Partial Differential Equations, 2017.

[5] Jens Berg and Kaj Nyström. Data-driven discovery of PDEs in complex datasets, 2018.

[6] Samuel Rudy, Alessandro Alla, Steven L. Brunton, and J. Nathan Kutz. Data-driven identification of parametric partial differential equations, 2018.

[7] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes, 2013.

[8] David Zheng, Vinson Luo, Jiajun Wu, and Joshua B. Tenenbaum. Unsupervised Learning of Latent Physical Properties Using Perception-Prediction Networks, 2018.

[9] Tianfan Xue, Jiajun Wu, Katherine Bouman, and William Freeman. Visual Dynamics: Stochastic Future Generation via Layered Cross Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[10] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic Filter Networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 667–675. Curran Associates, Inc., 2016.