Supplementary Information for: Learning two-phase microstructure evolution using DeepONet and autoencoder architectures

Vivek Oommen[1], Khemraj Shukla[2], Somdatta Goswami[2], Rémi Dingreville[3,*], and George Em Karniadakis[1,2,*]

[1]School of Engineering, Brown University
[2]Division of Applied Mathematics, Brown University
[3]Center for Integrated Nanotechnologies, Sandia National Laboratories, Albuquerque, NM, 87185, USA
[*]rdingre@sandia.gov, george_karniadakis@brown.edu

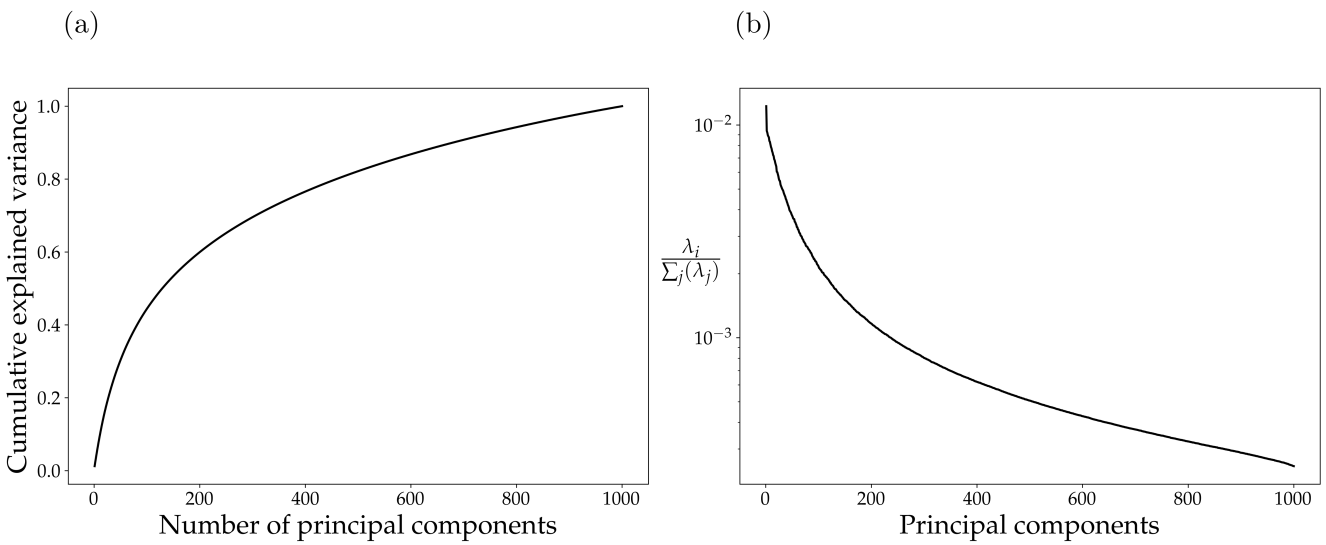
ABSTRACT

This supplementary information contains details on: (i) Principal Component Analysis (PCA) of microstructure dataset, (ii) experiment on number of training samples required, (iii) experiment on choice of training loss function, (iv) details on error in forecasting, and (v) sensitivity to noise.

Contents

Supplementary Note 1: Principal Component Analysis (PCA) of microstructure dataset

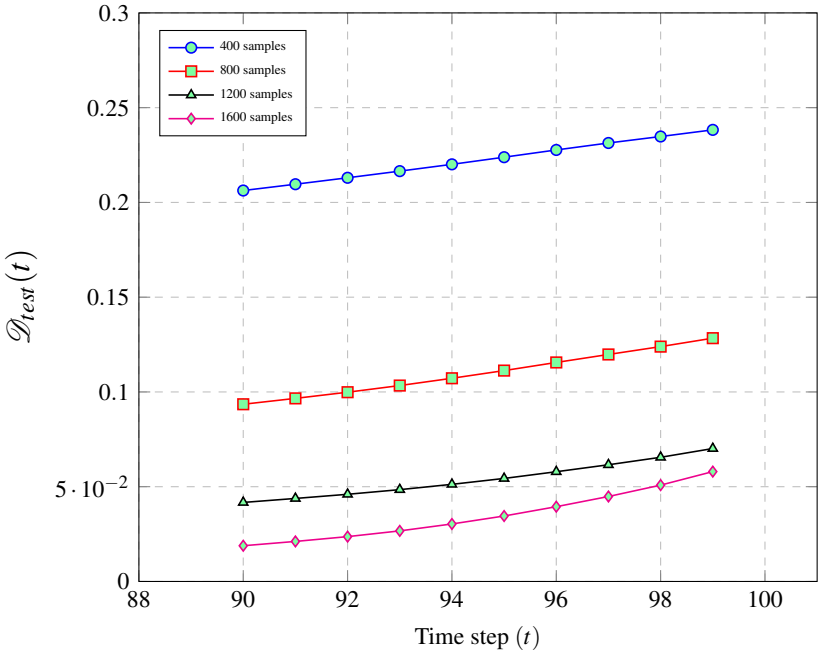
To circumvent issues pertaining to the data dimensionality and preparing the phase-field microstructure data for DeepONet training, we explored a couple of options. First, we tried using Principal Component Analysis (PCA) with linear kernel, for reducing the dimensionality of the data[1–3]. The low-dimensional representation of the data obtained from PCA is a linear transformation of the high-dimensional data and discards the insignificant modes (eigen/singular) corresponding to the lower eigen/singular values (λ_i). However, the system considered here is not diffusive and this is shown in Suppl. Fig. 1. Therefore, using PCA for reducing the dimensionality of the microstructure description could result in the loss of valuable information, if only a convenient low number of principal components are considered. Learning a non-linear mapping from a high dimensional to a low-dimensional latent space is one way to compress data without losing as much information as in the PCA.



Supplementary Figure 1. (a) The cumulative explained variance curve on the dominant 1000 eigenvalues of the phase-field microstructure (b) The energy curve on the dominant 1000 eigenvalues of the phase-field microstructure. The two plots suggest that microstructure data represent a non-diffusive system.

Supplementary Note 2: Experiment on number of training samples required

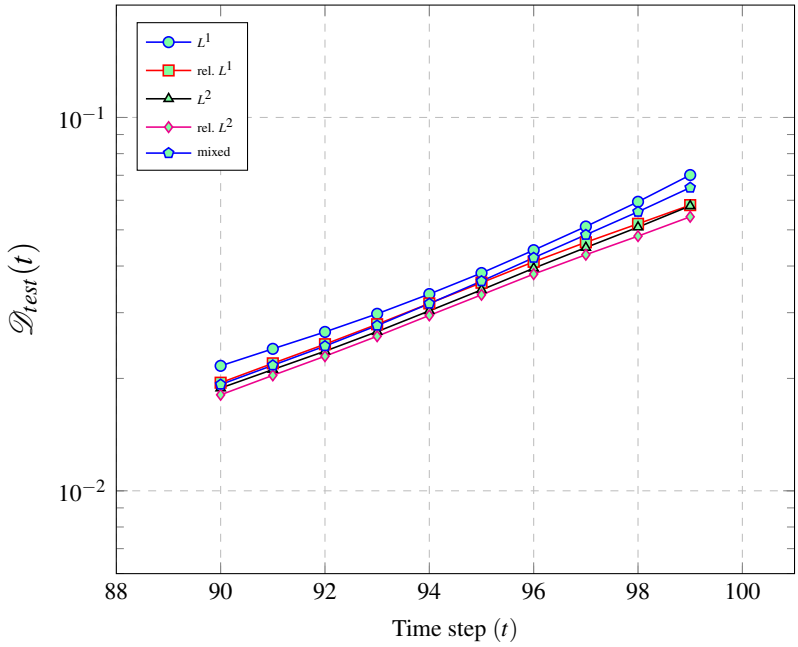
We carried out a computational experiment to analyze the sensitivity of the proposed approach to the number of samples used for training. We consider training datasets with 25%, 50%, 75% and all the 1600 training samples. We trained separate Autoencoder-DeepONet models on each of the four datasets. The model performance is evaluated on the basis of forecasting errors on test data, shown in Suppl. Fig. 2. We observe better accuracy in model's predictions while increasing the number of training samples. The model trained with 1200 data samples shows similar accuracy as the best model trained with 1600 data samples.



Supplementary Figure 2. Sensitivity of the proposed surrogate model to the number samples in training data. $\mathscr{D}_{test}(t)$ represents the relative squared error computed across the samples in test dataset at different time steps. We analyze the performance of each model on the basis of $\mathscr{D}_{test}(t)$ at forecasting time steps.

Supplementary Note 3: Experiment on choice of training loss function

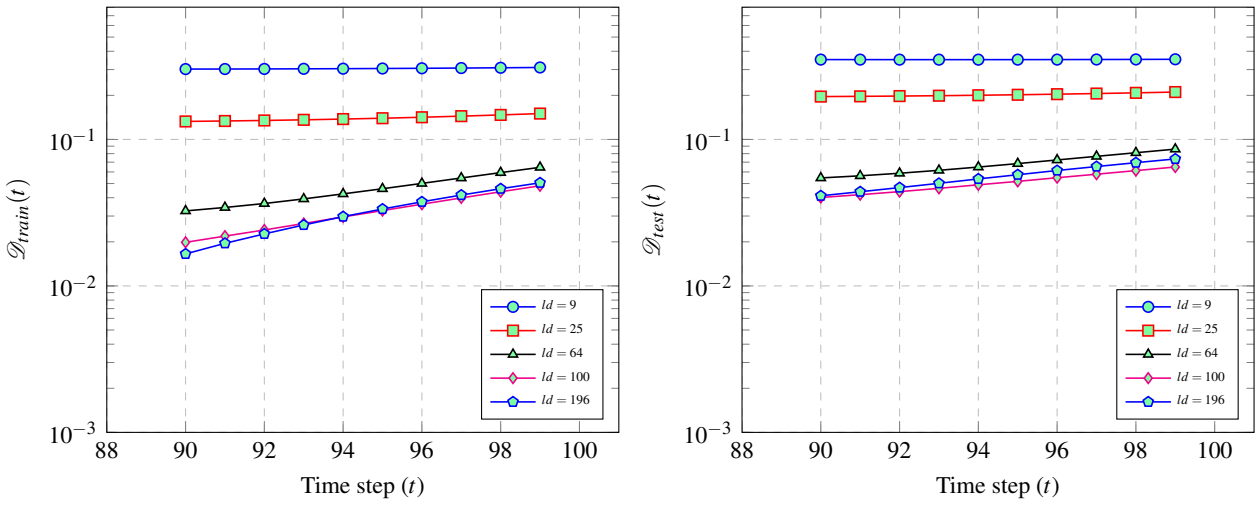
We investigate the effect of using different loss functions on training of the autoencoder models. Specifically, we trained autoencoders by minimizing (1) L^1 loss, (2) relative L^1 loss, (3) L^2 loss, (4) relative L^2 loss, and (5) mixed loss (L^2 loss for the initial 5000 epochs and L^1 loss for the remaining epochs). Now, (i) we train DeepONet model with architecture 1 on each of the learned latent microstructure data and (ii) re-transform the DeepONet predictions using pre-trained decoder to retrieve the microstructure. Finally, we analyze the model performance by computing the forecasting error, $\mathscr{D}_{test}(t)$, on unseen test data, shown in Suppl. Fig. 3. We observe that, models whose autoencoder is trained on L^2 loss is better than L^1 loss. Similarly, mean values of relative L^1 and L^2 are a better choice for autoencoder loss, $\mathscr{L}_{ae}$, than the mean of L^1 and L^2, respectively. In general, all the models perform well consistently.



Supplementary Figure 3. Variation of forecasting error on test data for models with different autoencoder loss functions, $\mathscr{L}_{ae}$, specifically L^1, relative L^1, L^2, relative L^2 and mixed loss(L_2 loss for the initial 5000 epochs and L^1 loss for the remaining epochs). $\mathscr{D}_{test}(t)$ represents the relative L^2 error computed across the samples in test dataset at different time steps.

Supplementary Note 4: Error in forecasting

A closer look at the forecasting predictions offers further insights on the DeepONet predictive performance. We computed the mean of the relative L^2 error, $\mathscr{D}(t)$ for forecasting time steps $t = 90, 91, ..., 99$, across the training and the testing datasets for all the latent dimensions considered in this study and plotted these values in Suppl. Fig. 4. We observe that the mean relative L^2 error reduces by increasing the latent dimension of the autoencoder model. In other words, the model with larger latent spaces is able to predict microstructures better while forecasting. This is intuitive because a larger dimension of the latent space implies that there are more basis functions to express the encoded information about the microstructure and its evolution, and therefore the network has an improved representation capability. However, this trend seems to saturate beyond $l_d = 100$. For the model with $ld = 100,\ 196$, the forecasting error is always less than 6%. The logarithm of the relative L^2- error linearly increases for $l_d = 64,\ 100$ and 196 for forecasting time step, whereas for $l_d = 9$ and $ld = 25$, the error is high and remains constant.



(a) Forecasting error on train dataset.

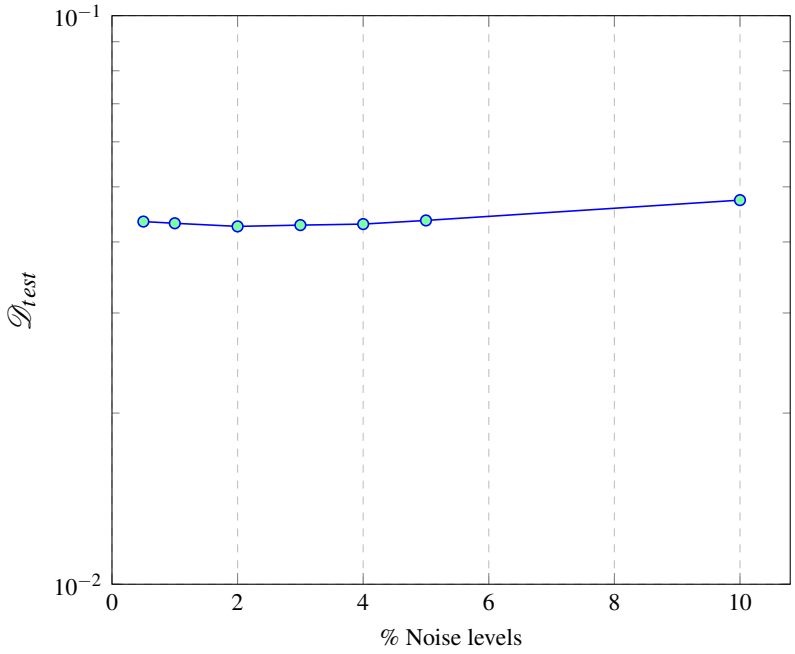
(b) Forecasting error on test dataset.

Supplementary Figure 4. (a) and (b) represents the relative L^2 error for predictions on the train and the test dataset. $\mathscr{D}_{train}(t)$, $\mathscr{D}_{test}(t)$ represents the mean of relative L^2 error computed on train and test datasets respectively.

Supplementary Note 4: Sensitivity to noise

We investigate how the accuracy of prediction varies by systematically increasing the noise levels in the model input. For this analysis, we considered the best model with $l_d = 196$ and DeepONet architecture 1 with sin activation functions. We added a Gaussian white noise with mean zero and standard deviations, $\sigma = 0.5\%, 1\%, 2\%, 3\%, 4\%, 5\%, 10\%$ to the input. To evaluate the model performance, we used the relative L^2 norm, $\mathscr{D}$, computed across all the time steps. $\mathscr{D}$ is calculated across the samples present in the test dataset.

From Suppl. Fig. 5, the relative L^2 norm does not increase much when noise is added to the model input. Previous studies[4–6] illustrated the capability of the autoencoders to denoise noisy images. The transformation to a low-dimensional latent space forces the autoencoders to retain the dominant features alone. The convolutional autoencoder used in this approach does exactly that by de-noising the noisy microstructure input. The output of the convolutional encoder is almost in its pure form, free from noise and therefore enables the DeepONet to make stable predictions. The decoder accurately reconstructs the microstructure from the predictions made by DeepONet in the latent space.



Supplementary Figure 5. Sensitivity of the proposed surrogate model to different levels of noise added to the test data. $\mathscr{D}_{test}$: represents the mean of relative L^2 error computed on test dataset for all time steps.

Acknowledgements

RD and GEK acknowledge funding under the *Beyond*Fingerprinting Sandia Grand Challenge Laboratory Directed Research and Development (GC LDRD) program. The phase-field framework is supported by the Center for Integrated Nanotechnologies (CINT), an Office of Science user facility operated for the U.S. Department of Energy. This research was conducted using computational resources and services at the Center for Computation and Visualization, Brown University. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

Supplementary References

1. Brough, D. B., Kannan, A., Haaland, B., Bucknall, D. G. & Kalidindi, S. R. Extraction of process-structure evolution linkages from x-ray scattering measurements using dimensionality reduction and time series analysis. *Integrating Materials and Manufacturing Innovation* **6**, 147–159 (2017).

2. Niezgoda, S. R., Kanjarla, A. K. & Kalidindi, S. R. Novel microstructure quantification framework for databasing, visualization, and analysis of microstructure data. *Integrating Materials and Manufacturing Innovation* **2**, 54–80 (2013).

3. Gupta, A., Cecen, A., Goyal, S., Singh, A. K. & Kalidindi, S. R. Structure–property linkages using a data science approach: application to a non-metallic inclusion/steel composite system. *Acta Materialia* **91**, 239–254 (2015).

4. Vincent, P., Larochelle, H., Bengio, Y. & Manzagol, P.-A. Extracting and Composing Robust Features with Denoising Autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, 1096–1103 (2008).

5. Vincent, P. *et al.* Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research* **11** (2010).

6. Gondara, L. Medical Image Denoising Using Convolutional Denoising Autoencoders. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, 241–246 (IEEE, 2016).