
Latent Nonlinear Wave Dynamics in Image Datasets and Autoencoder Reconstructions

Alexey Yermakov

Dept. of Electrical & Computer Engineering
University of Washington
Seattle, WA 98195
alexeyy@uw.edu

Lillian J. Ratliff

Dept. of Electrical & Computer Engineering
University of Washington
Seattle, WA 98195
ratliff@uw.edu

J. Nathan Kutz

Dept. of Applied Mathematics
University of Washington
Seattle, WA 98195
kutz@uw.edu

Abstract

Many deep learning models implicitly structure data by evolving it through layers in a way that resembles the flow of a dynamical system. In this work, we investigate the structure of classical image datasets through the lens of dynamical systems and partial differential equations (PDEs). We demonstrate that several widely used image datasets can be effectively modeled by a simple wave-like PDE in the latent embedding, revealing an underlying geometric and temporal coherence. Furthermore, we show that when these datasets are passed through a basic autoencoder, the reconstructed data preserves the wave-like structure up to a high-frequency cutoff. Remarkably, this indicates that neural networks inherently respect the dataset’s underlying dynamics, suggesting that PDE-inspired approaches can potentially guide the design of more structured and efficient representations. These findings offer new insights into the interplay between data geometry and deep learning, and suggest that viewing datasets through the framework of PDE dynamics may yield fruitful directions for representation learning. The data and codebase are available at the following link: https://github.com/yyexela/dataset_dynamics

1 Introduction

Understanding and exploiting the structure of high-dimensional data is central to modern machine learning. Many successful models—from word embeddings like `word2vec` to large language models and image diffusion models [1–5]—implicitly rely on structured latent spaces to extract and manipulate the underlying information in data. Natural images, in particular, exhibit rich statistical regularities, including scale invariance, spatial correlations, and predictable local structures [6–10], suggesting inherent geometric and temporal coherence. While prior work has characterized these statistical properties and explored spectral biases in neural networks [11, 12], less is known about how geometric and temporal regularities can simultaneously be captured through a dynamical systems framework—which could enable latent-space control, data generation, and exploration of underrepresented modes—and how such structures are preserved in learned latent neural representations. This paper starts to address this gap.

Motivated by a growing body of research on spectral analysis of deep networks, we focus on uncovering the latent dynamical structure of datasets themselves. Prior work has shown that neural

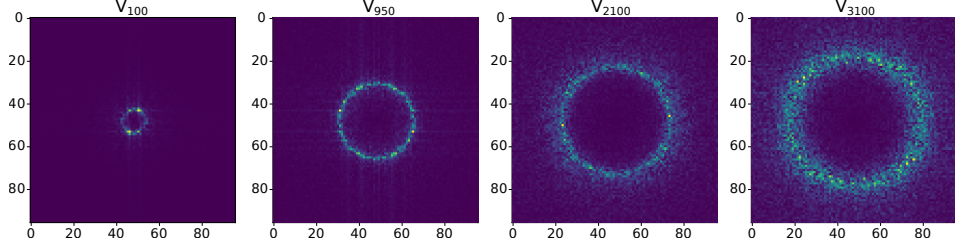


Figure 1: Spectral eigenimage evolution for the PCAM dataset. Shown are eigenimages corresponding to (sorted) indices $j \in \{100, 950, 2100, 3100\}$.

networks often learn low-frequency information first [11], create high-frequency barriers that separate classes [12], or can be modified to control spectral bias during training [13, 14]. Inspired by these insights, we study the spectral eigenimages of classical datasets and investigate how their underlying dynamical (wave-like) patterns are maintained through autoencoder reconstruction.

Our findings reveal a propagating wave structure that can be described by a simple partial differential equation (PDE) across multiple datasets. In particular, we find that different classical image datasets—e.g., MNIST and CIFAR—can be modeled by wave-like PDEs in latent space, revealing a common underlying structure despite differences in dataset-specific terms, and remarkably we observe a propagating wave in the datasets’ spectral eigenimages (cf. Figure 1). Importantly, this structure is largely *preserved by autoencoders*, thereby suggesting that neural networks inherently respect the underlying dynamical system representation of the data. This perspective opens the door to abstracting neural network architectures as task-dependent PDEs, analogous to how diffusion models use stochastic differential equation formulations. Such a framework can enable new approaches for representation learning, generative modeling, and even data-driven control, e.g., to detect underrepresented samples or guide optimization in structured latent spaces.

2 Methodology

We first describe the pipeline for converting a grayscale image dataset to a spectral representation visualized in Figure 2. Each image is first flattened into a vector, and each vector is stacked on top of one another to create a single 2-dimensional matrix for the entire dataset. This 2-dimensional matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is decomposed into a product of three matrices through Singular Value Decomposition (SVD): $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$, where $\mathbf{U} \in \mathbb{R}^{m \times n}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ are unitary and $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$ is diagonal, containing the singular values of \mathbf{A} in descending order (see, e.g., [15]). The vectors in \mathbf{V} form a sorted basis of flattened images that span the original dataset. Using the language of Navarrete and Ruiz-Del-Solar [16], we call these basis images *eigenimages*. The eigenimages are sorted by descending singular value, where larger singular values correspond to eigenimages capturing greater variance in the dataset. Each eigenimage is then reshaped into a 2-dimensional image and passed into the 2-dimensional FFT algorithm to obtain the spectral eigenimage. The magnitude of the sorted spectral eigenimages are used in our analysis.

In addition to obtaining a deeper understanding of the frequency information encoded in each studied dataset, we use this pipeline to observe the effects of how a simple MLP AE [17, 18] with a latent space bottleneck affects the spectral eigenimages, shown in Figure 3.

3 Results

We explore several common PyTorch Datasets which are listed in Table 2, Appendix A. For the first set of experiments, we simply take each image dataset and pass it directly into the spectral pipeline explained in Section 2. By combining the sorted

magnitudes of spectral eigenimages into a movie, one observes a propagating wave in for each dataset. This is shown in Figure 1 for the PCAM dataset, but the same applies to all the studied datasets. Visualizations of the j -th sorted spectral eigenimage are shown in Figure 4, where j was selected to demonstrate the unique wave composition of each dataset.

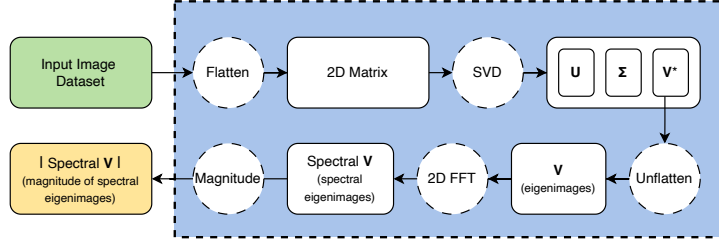


Figure 2: Grayscale image dataset spectral pipeline.

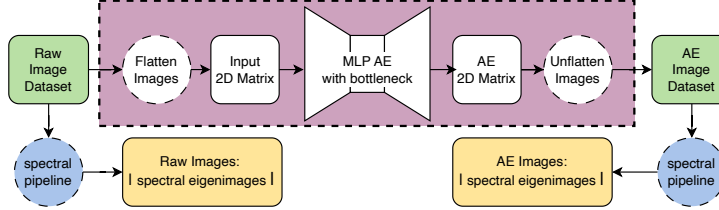


Figure 3: MLP AE pipeline.

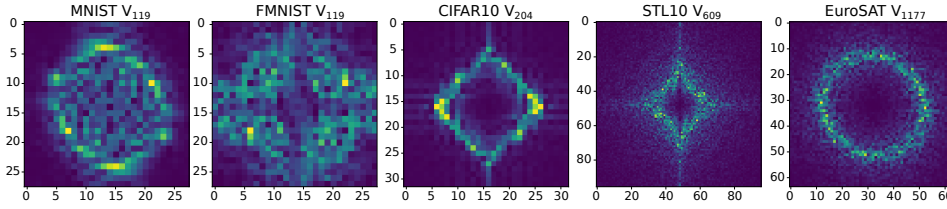


Figure 4: The j -th sorted spectral eigenimage of the remaining studied datasets. Shown are MNIST, FMNIST, CIFAR10, EuroSAT, and STL10. The value of j was determined heuristically to showcase the unique wave shape for each dataset.

Viewing the sorted spectral eigenimages as a propagating wave allows us to directly model the spectral information of each image dataset as a PDE in a purely data-driven manner. We pass this data into PySINDy: a Python library that contains a data-driven approach for discovering symbolic PDEs from raw data [19, 20]. The SINDy (sparse identification of nonlinear dynamics) algorithm is an efficient sparse regression algorithm for identifying underlying dynamical systems in data [21, 22]. The resulting equations describe the evolution of the wave front, shown in Table 1 with more details provided in Appendix B.

Table 1: Discovered PDEs

Dataset	p-norm	PDE
MNIST	2.000	$u_t = 0.010u_x u_{xxx} - 0.016u_x^2 u_{xxx}$
FMNIST	1.000	$u_t = 0.169u_{xx} - 1.034u_x u_{xx} + 1.214u_x^2 u_{xx}$
CIFAR10	1.273	$u_t = -0.560u_x - 1.587u^2 u_x + 0.175u^2 u_{xx}$
EuroSAT	1.492	$u_t = -13.008u^2 u_x - 0.007u^2 u_{xxx}$
STL10	0.318	$u_t = -6.207u^2 u_x - 0.002u^2 u_{xxx}$
PCAM	1.935	$u_t = -1.772u_x u_x - 11.196u_x^2 u_x - 0.003u_x^2 u_{xxx}$

These results show that complex high-dimensional datasets can be described in terms of a hierarchy of nonlinear PDE equations that exhibit shock front propagation in the latent embedding. Indeed, across data sets, generalizations of the underlying Burgers equation structure [23] with either nonlinear diffusive or dispersive regularization is observed. Burgers equation was one of the earliest models

used for the understanding of shock waves in the 1950s as humans approached super-sonic flight. In latent space, the model learns a propagating shock-like structure. Frequencies beyond the shock front are "filtered" so that the latent space acts as a nonlinear low-pass filter. The organization of the data into latent nonlinear wave dynamics is a remarkable observation that merits further exploration.

Viewing the MLP AE's spectral eigenimages alongside the original dataset shows that it prioritizes learning eigenimages corresponding to the largest variance directions (cf. Figure 5). In particular, the wave-front from the MLP AE spectral eigenimages closely matches the wave-front from the original data up to a specific frame/ j -value. After this j -value, the autoencoder's spectral eigenimages are noisy except for a low-frequency hole in the center of the spectral eigenimage, matching the size of the wave-front at the transition frame/ j -value. This hole persists for the remaining MLP AE spectral eigenimages. This reinforces previous results that MLP AEs work as a band-pass filter, prioritizing the learning of the low-frequency information of a dataset [12].

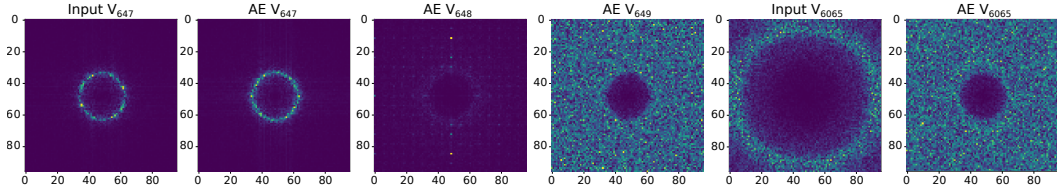


Figure 5: Evolution of spectral eigenimages of the original PCAM dataset (1,5) and the reconstructed PCAM dataset (2,3,4,6). Shown are $j = 647$ (1,2), $j = 648$ (3), $j = 649$ (4), and $j = 6,065$ (5,6). The MLP AE had a bottleneck size of 2,048 and was trained for 500 epochs with a batch size of 1,024, and a learning rate of 0.0001 using the Adam optimizer and the Mean Squared Error (MSE) loss function.

What this result also shows is that autoencoders preserve the dynamics of the spectral eigenimages. The governing equations describing the wave dynamics of spectral eigenimages are followed closely by the MLP AE. In fact, for a dataset containing M images of size $\ell \times w$, it appears that the first T spectral eigenimages of the input dataset are prioritized during training by the MLP AE (where $\ell \cdot w > T \geq 1$). We conjecture that the model bottleneck prevents the MLP AE from learning the high-frequency information for the remaining $j > T$ high-frequency information of the input dataset. Further analysis is presented in Appendix C. To the best of our understanding, this observation has not been shown before.

4 Conclusion

In this work, we study the frequency content of various image datasets and find that they can be described by a set of nonlinear shock-like PDEs with a common underlying structure. Each PDE is simple and some form of generalization of Burgers equation, containing no more than three terms, although the dataset it describes is up to $96^2 = 9,216$ dimensional. We also observe that MLP AEs mimic the low-frequency content of the underlying datasets by following the dynamics of the spectral eigenimages until a certain high-frequency limit. Thus the latent representation acts as a nonlinear low-pass filter, organizing the data into a propagating shock-structure.

Our work opens several promising avenues for future research by leveraging a structured dynamical systems perspective on image datasets. For instance, one possible direction is compression, where instead of storing the full dataset, a stored initial condition is evolved according to the dataset's PDE to reconstruct the original dataset. This approach could also enable synthetic data generation, producing samples consistent with the dataset's intrinsic dynamics. Another exciting possibility is using the PDE to detect out-of-distribution samples or underrepresented regions in the data. Finally, the wave-like dynamics uncovered via the proposed methodology could be incorporated into diffusion models, guiding latent space evolution to improve both in-class image synthesis and conditional generation. More broadly, this framework suggests a principled way to connect neural representations, dataset structure, and PDE-based modeling, potentially inspiring new architectures and analysis techniques across machine learning.

While our experiments do not directly establish a PDE structure for neural network architectures themselves, the fact that classical datasets exhibit wave-like PDE dynamics and these dynamics are largely preserved through autoencoder reconstruction suggests that analogous task-specific PDE abstractions for networks may be possible. Such a perspective, if realized, could enable the application of stochastic control and optimization techniques to guide network behavior, latent space evolution, and generative modeling.

Acknowledgments and Disclosure of Funding

Acknowledgements: AY thanks Jake Stevens-Haas (University of Washington) and Zachary Nicolaou (University of Washington) for their valuable discussions and input on PySINDy.

Funding: This work was supported in part by the US National Science Foundation (NSF) AI Institute for Dynamical Systems (dynamicsai.org), grant 2112085. JNK further acknowledges support from the Air Force Office of Scientific Research (FA9550-24-1-0141). AY is supported by the NSF Graduate Research Fellowship Program under Grant No. DGE-2140004. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsors.

References

- [1] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.
- [2] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- [3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [4] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In Katrin Erk and Noah A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL <https://aclanthology.org/P16-1162/>.
- [5] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.
- [6] David J Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, 4(12):2379–2394, 1987.
- [7] Daniel L Ruderman. The statistics of natural images. *Network: computation in neural systems*, 5(4):517, 1994.
- [8] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.
- [9] David Beymer and Tomaso Poggio. Image representations for visual learning. *Science*, 272(5270):1905–1909, 1996.
- [10] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828, 2013.
- [11] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pages 5301–5310. PMLR, 2019.

- [12] Sara Fridovich-Keil, Raphael Gontijo Lopes, and Rebecca Roelofs. Spectral bias in practice: The role of function frequency in generalization. *Advances in Neural Information Processing Systems*, 35:7368–7382, 2022.
- [13] Siavash Khodakarami, Vivek Oommen, Aniruddha Bora, and George Em Karniadakis. Mitigating spectral bias in neural operators via high-frequency scaling for physical systems. *arXiv preprint arXiv:2503.13695*, 2025.
- [14] Peihao Wang, Wenqing Zheng, Tianlong Chen, and Zhangyang Wang. Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=0476oWmiNNp>.
- [15] Jose Nathan Kutz. *Data-driven modeling & scientific computation: methods for complex systems & big data*, chapter 15.10, pages 413–417. OUP Oxford, 2013.
- [16] Pablo Navarrete and Javier Ruiz-Del-Solar. Analysis and comparison of eigenspace-based face recognition approaches. *International Journal of Pattern Recognition and Artificial Intelligence*, 16(07):817–830, 2002.
- [17] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
- [18] Dor Bank, Noam Koenigstein, and Raja Giryes. Autoencoders. *Machine learning for data science handbook: data mining and knowledge discovery handbook*, pages 353–374, 2023.
- [19] Alan A. Kaptanoglu, Brian M. de Silva, Urban Fasel, Kadierdan Kaheman, Andy J. Goldschmidt, Jared Callahan, Charles B. Delahunt, Zachary G. Nicolaou, Kathleen Champion, Jean-Christophe Loiseau, J. Nathan Kutz, and Steven L. Brunton. Pysindy: A comprehensive python package for robust sparse system identification. *Journal of Open Source Software*, 7(69):3994, 2022. doi: 10.21105/joss.03994. URL <https://doi.org/10.21105/joss.03994>.
- [20] Brian de Silva, Kathleen Champion, Markus Quade, Jean-Christophe Loiseau, J. Kutz, and Steven Brunton. Pysindy: A python package for the sparse identification of nonlinear dynamical systems from data. *Journal of Open Source Software*, 5(49):2104, 2020. doi: 10.21105/joss.02104. URL <https://doi.org/10.21105/joss.02104>.
- [21] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.
- [22] 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, 2017.
- [23] Mayur P Bonkile, Ashish Awasthi, C Lakshmi, Vijitha Mukundan, and VS Aswin. A systematic literature review of burgers’ equation with recent advances. *Pramana*, 90(6):69, 2018.
- [24] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.
- [25] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant CNNs for digital pathology. June 2018.
- [26] Yann LeCun. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>, 1998.
- [27] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.
- [28] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

- [29] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.
- [30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

A Datasets

We consider several common PyTorch Datasets which are listed in Table 2 [24–30]. We did not consider other interesting image datasets, such as the CelebA dataset, due to the full size of the dataset exceeding the 32 bit limit from the BLAS backend for computing the SVD in PyTorch.

Each dataset was converted to grayscale before being used in any of the aforementioned pipelines.

Table 2: Used PyTorch Datasets

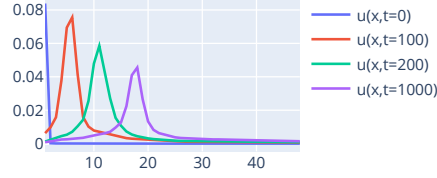
Dataset Name	Image Width	Image Height	Number of Images
MNIST	28	28	60,000
FMNIST	28	28	60,000
CIFAR10	32	32	50,000
EuroSAT	64	64	27,000
STL10	96	96	5,000
PCAM	96	96	262,144

B Governing Equations

We explain in more detail how to obtain the governing equations presented in Section 3. First, a chosen image dataset needs to be converted to a grayscale format so that the dataset can be represented as a single 2-dimensional matrix. This dataset needs to then be passed into the pipeline described in Figure 2 to obtain a sorted spectral eigenimage dataset. To determine the p-norm the propagating wave follows, pick one of these spectral eigenimages that has the most clear structure and perform a p-norm fit to the data. Afterwards, calculate the radial integrals using the provided p-norm for each spectral eigenimage. This whole process yields a 1+1-dimensional system evolving over time, where the spatial dimension is the p-norm distance from the origin and the temporal dimension is each successive spectral eigenimage.

Note that for many of the datasets there is a spike for the first few eigenimages, which need to be skipped in order to assist PySINDy in finding a set of equations (n_{skip}), shown in Figure 6 (left). Similarly, the behavior for the final few eigenimages is noisy, so they are skipped as well (n_{end}). This noise is shown in Figure 6 (right), which contains MNIST’s final spectral eigenimage. Clearly, a radial integral here would not be useful data for obtaining a PDE.

This trimmed 1+1-dimensional system can be passed directly into PDE-FIND in the PySINDy library to obtain a set of governing equations. For each dataset, we used the Sequentially Thresholded Least Squares (STLSQ) algorithm with a polynomial library of degree 2 and 3rd order derivatives. The PySINDy parameters used to generate Table 1 are presented in Table 3.

Time slices of $u(x,t)$ for PCAM

Final MNIST Spectral Eigenimage

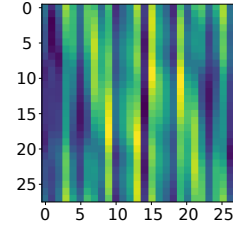


Figure 6: Visualization of time slices of the radial integrals of PCAM’s spectral eigenimages (left). Final ($j = 784$) MNIST spectral eigenimage (right).

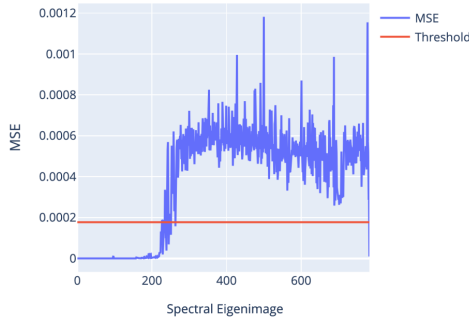
Table 3: PySINDy PDE-FIND Parameters

Dataset	p-norm	n_{start}	n_{end}	α	threshold
MNIST	2.000	10	150	$1e-5$	30
FMNIST	1.000	15	150	$1e-5$	62
CIFAR10	1.273	40	375	$1e-5$	20
EuroSAT	1.492	60	1500	$1e-5$	250
STL10	0.318	60	800	$1e-5$	292
PCAM	1.935	100	4000	$1e-5$	325

C Transition Frames

To study what would happen to the transition frame when varying the MLP model parameters, we automated the calculation of the transition frame by calculating when the MSE between the true spectral eigenimage dataset and the reconstructed spectral eigenimage dataset first exceeded 15% of the maximum difference, shown in Figure 7 (left). We then trained various model bottlenecks, from a bottleneck of size of 1 up to a bottleneck size of 28^2 , the original dimension of the MNIST dataset. Each MLP was trained 5 times, with a learning rate of 0.002 for 100 steps using the Adam optimizer. The results are shown in Figure 7 (right).

Transition Frame for MNIST



Transition Frame per Model Bottleneck (MNIST)

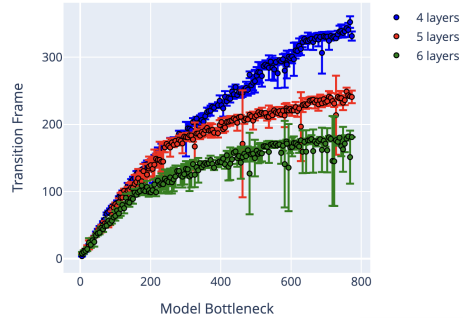


Figure 7: MSE between MNIST spectral eigenimages and reconstructed MNIST spectral eigenimages (left) and the transition frame across multiply varying model parameters (right). The scatter plot (right) has mean and standard deviation across five random initial model weights.

We observe that a smaller model bottleneck results in a smaller transition frame. This is likely due to the model capacity being smaller, forcing the MLP AE to learn a smaller amount of the low-frequency information that if the model were larger. Furthermore, we also observe that a smaller number of layers corresponds to a larger maximum transition frame. This is likely due to the MLP model

simultaneously having less weights than models with more layers while being trained for the same amount of steps with the same step-size. As a result, in the smaller MLP is able to reconstruct the dataset better over the same amount of steps.

In all cases, the MLP AE is learning the low-frequency information first. That is, if the transition frame is at $j = T$, then the MSE between the input spectral eigenimage and the reconstructed eigenimage is very small for all $j < T$. Then, for all $j > T$, the MSE between the input and the reconstructed spectral eigenimage is proportionally larger, corresponding to high-frequency information.