LEARNING REDUCED ORDER DYNAMICS VIA GEOMETRIC REPRESENTATIONS

Imran Nasim
IBM Research Europe, UK
imran.nasim@ibm.com

Melanie Weber
Harvard University, Cambridge MA, USA
mweber@seas.harvard.edu

ABSTRACT

Geometric Representation Learning is a cornerstone of (unsupervised) learning and as such has been widely applied across domains. In this work, we consider the problem of data-driven discovery of system dynamics from spatial-temporal data. We propose to encode similarity structure in such data via a spatial-temporal proximity graph; we then apply a range of manifold learning and deep-learning based approaches to recover reduced order dynamics. We perform a systematic analysis of the learned representations, comparing the different approaches' ability to capture local and global geometric features of the system dynamics.

Keywords graphs · dynamical systems · manifold learning · graph neural networks

1 Introduction

The problem of learning the dynamics of a complex system from spatial-temporal data arises in applications across the Sciences and Engineering. Often such data has a low-dimensional structure, usually induced by inherent symmetries in the underlying system. In the dynamical systems literature, there is a growing body of work on learning dynamics from data, e.g., [7, 6, 19]. A related body of literature considers the problem of identifying and characterizing low-dimensional structure in data. Such methods range from classical manifold learning approaches [15, 24, 1] to more recent deep learning based methods [21, 8, 12], which are typically applied to static (non-time-dependent) data. The question of which method performs best in a given setting is highly data- and domain-dependent. In this work, we propose a framework for investigating the utility of such geometric representation learning approaches for the recovery of reduced order dynamics from *spatial-temporal* data. We consider data that is generated by physical systems that can be modeled via classical Partial Differential Equations. Our framework recovers the system dynamics in a purely data-driven manner, relying on a spatial-temporal proximity graph that encodes similarity structure in the data. We perform a systematic qualitative and quantitative analysis of the different approaches with the aim of identifying methods that can reliably recover the reduced order dynamics. Our analysis sheds light on the advantages and disadvantages of each method with respect to their ability to capture local and global geometric features in spatial-temporal data.

2 Methods

2.1 Learning manifolds from data

In this work, we assume that data lies on or near a low-dimensional manifold, an assumption commonly known as the *manifold hypothesis*. A large body of literature has evolved around the problem of reconstructing manifolds from data (*manifold learning*). Formally, let $\mathcal{X} \times (0,T) \subseteq \mathbb{R}^{d+1}$ denote time series data measured in a physical system. In practise, these measurements may be inaccurate (e.g., affected by noise) or incomplete (e.g., not representative of the low-dimensional manifold). We want to learn a low-dimensional representation of the system's dynamics, which allows for characterizing its local and global geometric properties, characterized by a *representation function* $\phi : \mathcal{X} \times (0,T) \to \mathcal{M}$, where $\mathcal{M} \subset \mathbb{R}^{d+1}$ (with dimension $\dim(\mathcal{M}) \ll d$). To learn the representation function ϕ, we will utilize classical manifold learning [15, 24, 22], as well as deep learning based approaches that emerged more recently in the literature [21, 12, 25]. In this section, we review common building blocks of such methods (Fig. 1).

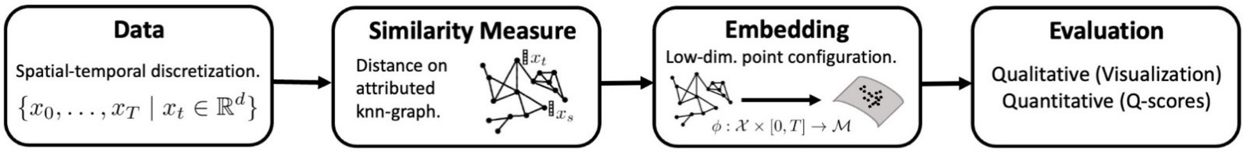
Data
Spatial-temporal discretization.
$\{x_0, \ldots, x_T \mid x_t \in \mathbb{R}^d\}$
Similarity Measure
Distance on attributed knn-graph.
x_t
x_s
Embedding
Low-dim. point configuration.
$\phi : \mathcal{X} \times [0,T] \to \mathcal{M}$
Evaluation
Qualitative (Visualization)
Quantitative (Q-scores)

Figure 1: Overview of our Representation Learning framework.

Similarity measure. A key component of every representation learning algorithm is a *similarity measure* imposed on the data space, which provides a means for quantifying the similarity or dissimilarity of pairs of data points. In the algorithms considered here, we encounter three types of similarity measures: (i) *Euclidean distances*, which compute pairwise distances in the ambient space; (ii) *geodesic distances*, which are approximated from a proximity graph; (iii) *probabilistic distances*, which are approximated via the likelihood that two nodes co-occur in a random walk on a proximity graph. The choice of similarity measure is often not unique and represents a crucial design choice. A key subroutine of many similarity measures is the construction of a *proximity graph*. Unlike in the typical manifold learning setting, our graph is *time-dependent*, with nodes representing time points; the spatial discretization is encoded in the node attributes. To construct a *spatial-temporal proximity graph*, nodes are connected by edges, if they are k-nearest neighbors with respect to the similarity of the node attributes. The construction requires a careful choice of the parameters k, which may require a grid search in practise. The actual similarities (i)-(iii) are estimated from the proximity graph.

Embedding. Assuming access to a similarity measure, we can compute a point configuration on a low-dimensional manifold. This process implicitly approximates a representation function $\phi : \mathcal{X} \times (0, T) \to \mathcal{M}$, which embeds the data $\mathcal{X}$ into the target manifold $\mathcal{M}$. In classical manifold learning methods, the point configuration is usually computed by solving an eigenvalue problem. Shallow graph embedding are optimization-based, i.e., a loss function, which is defined based on the chosen similarity measure, is optimized to compute a point configuration. In deep learning-based embeddings, point configurations are trained via message-passing.

Evaluation. Once a point configuration is computed, we evaluate its utility. We propose to measure the quality of a learned representation qualitatively via visualization and quantitatively via local and global Q-scores, which are classical measures of embedding quality [17]. We defer all formal definitions to Apx. B.

2.2 Algorithms

In this section we briefly review representation learning algorithms considered in this study. A taxonomy is given in Tab. 2 in Apx. A.

Classical Manifold Learning algorithms. Arguably the simplest approach for inferring low-dimensional structure in data is via linear projections, e.g., using PCA. The most common approach for learning non-linear subspaces from data are manifold learning algorithms. At the heart of those algorithms lies often an eigenvalue problem, where utilizing the eigenvectors associated with the top eigenvalues leads to projections that preserve pairwise distances globally (*global methods*), whereas projections based on the eigenvectors corresponding to the bottom eigenvalues preserves local pairwise distances (*local methods*). Examples of global methods are Isomap [24] and MDS [15]. We further consider LLE [22] as an example of a local method. Additionally we consider SE, which utilizes a probabilistic similarity measure. More details on all three approaches can be found in Apx. A.

Deep Learning on Point Clouds. One of the first Representation Learning approaches that utilized probabilistic similarity measures was DeepWalk[21], a shallow node embedding method. More recently, approaches based on Graph Neural Network (GNN) architectures have been utilized for graph embeddings. Here, we consider GraphSage [9], GCN [12] and GAT [25]. GCN and GraphSage are early GNN architectures, which extend convolutional neural networks to graph domains. GAT [25] implements the idea of graph attention, i.e., it translates the idea of *transformers* to the graph setting. More details on all four approaches can be found in Apx. A.

3 Experiments

3.1 Experimental setup

Case studies. The forced Korteweg–de Vries (fKdV) equation is an integrable non-linear PDE which is commonly used to model the weakly non-linear flow problem [3, 2]. In this study we consider the fKdV equation under the assumption of no disturbance where the equation can be written as [2]:

$$6u_t + u_{xxx} + (9u - 6(F-1))u_x = 0, \tag{1}$$

where F is the depth-based Froude number. This model has been shown to exhibit both periodic travelling wave and soliton dynamics [2].
The second model we consider is the Kuramoto–Sivashinksy equation (KS hereafter). The KS equation is a fourth-order non-integrable nonlinear PDE that can be used to model the pattern formation for several physical systems [16]. The viscous form of the equation can be written as

$$u_t + uu_x + u_{xx} + \nu u_{xxxx} = 0, \tag{2}$$

where ν is a coefficient of viscosity. The KS equation is a highly complex model which captures the dynamics of spatio-temporal instabilities. The KS model also displays chaotic motion, and due to its non-integrability, gives rise to a rich array of solution types depending on the value of the viscosity parameter ν. Most pertinent to this study, in the case where $\nu = \frac{16}{71}$ the KS equation exhibits bursting dynamics [13].
In addition to the two models described above, we consider the Sine Gordon (SG) equation which is an integrable PDE. The SG equation gained popularity for the ability to exhibit soliton solutions [11] and can be written as

$$u_{tt} - u_{xx} + \sin x = 0. \tag{3}$$

Data generation. To obtain the dynamics data we simulated the fKdV and KS equation on a grid of 64 points on a periodic domain of $-\pi \leq x \leq \pi$. The Froude number in fKdV was assumed to be $F = 1.5$. The initial wave profiles are of the form $u(x, 0) = A\cos(kx + \phi)$ where we fix $A = 0.5$, $k = 1$ and $\phi = 1$. The trajectory data is collected from time $t = 300$ with $\Delta t = 0.1$ yielding 1000 time points, which captures the dynamics well after the initial transients have died out. Numerical integrations of the fKdV and KS equations were performed using an explicit RK finite-difference scheme with a tolerance of 10^{-6} which we compared to a psuedo-spectral method to ensure accuracy.

For the SG equation, we used a grid of 64 points and solved the model using an adaptive 5th order exponential time differencing method [28], resulting in 1362 time points.

Implementation and Training. We construct a time-dependent proximity graph, where nodes represent time points with d-dimensional node attributes, which encode the spatial discretization. Connectivity is inferred via proximity, i.e., each node is connected to its k nearest neighbors. We then utilize geometric representation learning to learn low-dimensional node representations. We run the deep learning models for 1000 epochs using the SGD optimizer having a learning rate of 10^{-2} with a negative sampling loss. For GCN, GRAPHSAGE and GAT we use two hidden layers having 256 and 512 units. For both the classical and deep learning methods, we adopt the same graph construction method using a k nearest neighbours approach setting $k = 20$. For ease of visualization, we show results in three dimensions. Further implementation details and hyperparameter choices can be found in Apx. D.

3.2 Results

Table 1: Numerical results. Q-scores range from 0 (worst) to 1 (best).

	fKdV			KS			SG		
	Q_{local}	Q_{global}	$K_{\max}$	Q_{local}	Q_{global}	$K_{\max}$	Q_{local}	Q_{global}	$K_{\max}$
PCA	0.55	0.91	3	**0.99**	**0.95**	3	**0.79**	0.94	80
MDS	**0.75**	0.95	214	0.96	0.94	6	0.74	0.93	19
ISOMAP	0.63	**0.97**	247	0.84	0.94	1	0.77	**0.95**	71
LLE	0.44	0.68	4	0.64	0.90	302	0.44	0.76	3
SE	0.61	0.84	201	0.76	0.72	5	0.48	0.79	23
DW	0.52 (0.06)	0.76 (0.06)	156.1 (33.96)	0.38 (0.02)	0.57 (0.01)	10.7 (0.90)	0.39 (0.03)	0.71 (0.03)	63.7 (11.2)
GCN	0.47 (0.06)	0.73 (0.05)	41.5 (56.2)	0.50 (0.043)	0.77 (0.07)	40.4 (107.9)	0.43 (0.03)	0.72 (0.05)	48.3 (16.3)
GS	0.37 (0.08)	0.70 (0.07)	91.9 (90.3)	0.82 (0.02)	0.81 (0.05)	4.0 (0.0)	0.35 (0.06)	0.66 (0.05)	30.0 (20.8)
GAT	0.43 (0.03)	0.68 (0.03)	17.2 (3.18)	0.79 (0.011)	0.84 (0.03)	4.0 (0.0)	0.45 (0.02)	0.73 (0.04)	19.8 (0.75)

Qualitative and Quantitative Analysis. A crucial aspect of any reduced-order dynamical model is to ensure the geometry of the higher dimensional data is faithfully conserved, i.e., is capturing the underlying dynamics of the system.

To compare the classical and deep learning based methods, we apply all approaches to the KS bursting dynamics data and the travelling wave data from both the fKdV and SG equations. To probe the interpretability of the representations we consider three dimensional reduced order dynamics and compare these to the representations obtained by a Fourier projection on the two dominant modes of the raw data (see Fig. 2); a comprehensive comparison of all methods across all three case studies can be found in Appendix E. We also quantitatively measure the quality of the reduced order dynamics using the local and global Q-score metrics, which is presented in Table 1. For all the deep learning models we run 10 different initializations with different seed numbers giving the mean and standard deviation values, where the latter is the value within the parenthesis in Table 1.

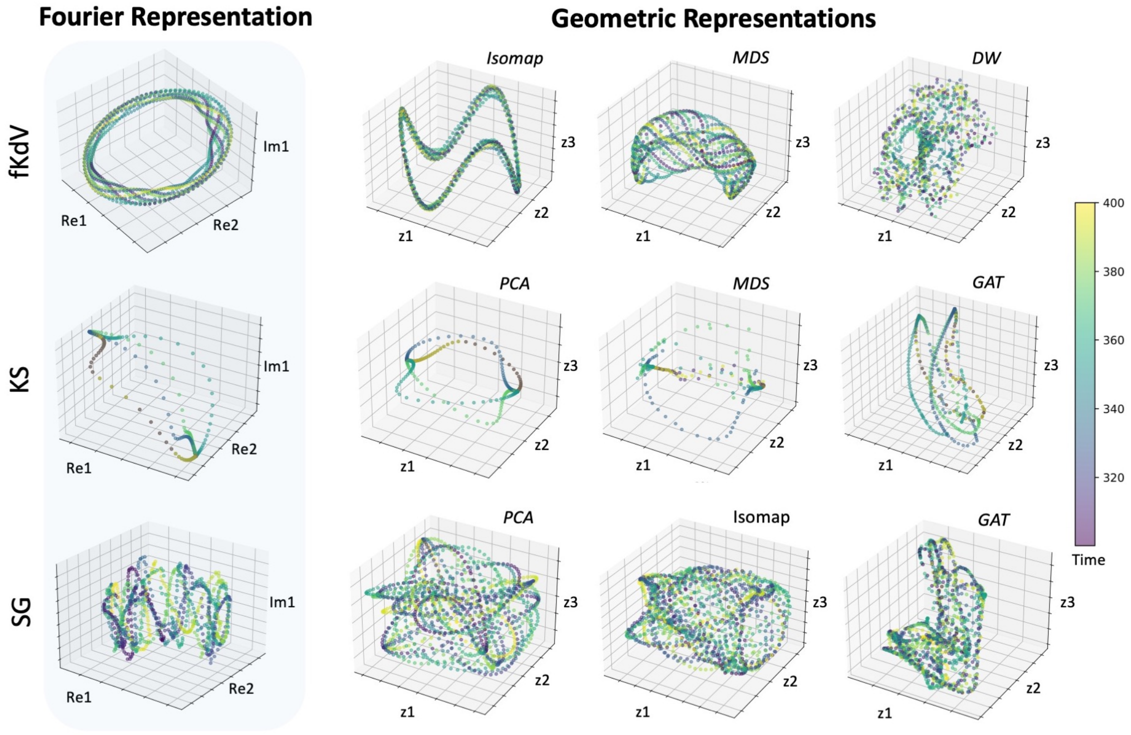

Figure 2: Three-dimensional visualizations of the reduced order dynamics (top row: forced Korteweg–de Vries (fKdV), middle row: Kuramoto–Sivashinksy (KS), bottom row: Sine Gordon (SG)). Left: Ground truth dynamics in Fourier space. Right: Geometric representations learned with top two classical and top deep-learning based approach.

Characterizing local and global geometric features. Comparing the learned dynamics both qualitatively and quantitatively, we notice that PCA and MDS perform best with respect to local and global Q-scores, a result that may indicate that variation in the data can be captured well with linear subspaces. ISOMAP, the third global manifold learning approach tested here, performs competitively; the local methods (LLE and SE) perform low with respect to all evaluation criteria. This suggest that a successful approach needs to accurately capture both similarity between nearby points and dissimilarity between far distant points. The inability of random-walk based similarity measures to capture global pairwise distances may stem from approximation errors inherent to their probabilistic nature, but further analysis is needed to explain the performance trade-offs between classical and deep learning based methods.

4 Impact statement

We have explored the utility of geometric representation learning approaches for recovering reduced-order dynamics in spatial-temporal data. Our systematic quantitative and qualitative study of representation trade-offs suggests that classical manifold learning approaches appear to be superior to deep learning approaches. In particular, simple approaches, such as PCA and MDS seem to extract the underlying dynamics best and also yield interpretable representations. Our analysis utilized a spatial-temporal proximity graph, which allows for extending classical representation learning approaches from static to dynamic data. We aim to extend our study to a broader range of case studies and representation learning methods with the aim of understanding trade-offs and commonalities between these approaches more comprehensively. We further aim to investigate the effects of non-uniform sampling to gain better insight into the applicability of the tested approaches to real-world data.

References

[1] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[2] Benjamin J. Binder. Steady two-dimensional free-surface flow past disturbances in an open channel: Solutions of the korteweg–de vries equation and analysis of the weakly nonlinear phase space. *Fluids*, 4(1), 2019.

[3] BJ Binder and J-M Vanden-Broeck. Free surface flows past surfboards and sluice gates. *European Journal of Applied Mathematics*, 16(5):601–619, 2005.

[4] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences*, 103(5):1168–1172, 2006.

[5] Frédéric Chazal and Bertrand Michel. An introduction to topological data analysis: Fundamental and practical aspects for data scientists. *Frontiers in Artificial Intelligence*, 4, 2021.

[6] Oluwadamilola Fasina, Smita Krishnaswamy, and Aditi Krishnapriyan. Geometric NeuralPDE (GNPnet) Models for Learning Dynamics. 2022.

[7] Feng Gao, Guy Wolf, and Matthew Hirn. Geometric Scattering for Graph Data Analysis. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2122–2131, 2019.

[8] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[9] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[10] Geoffrey E Hinton and Sam Roweis. Stochastic neighbor embedding. *Advances in neural information processing systems*, 15, 2002.

[11] Ryogo Hirota. Exact solution of the sine-gordon equation for multiple collisions of solitons. *Journal of the Physical Society of Japan*, 33(5):1459–1463, 1972.

[12] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[13] Michael Kirby and Dieter Armbruster. Reconstructing phase space from pde simulations. *Zeitschrift für angewandte Mathematik und Physik ZAMP*, 43:999–1022, 1992.

[14] Anna Klimovskaia, David Lopez-Paz, Léon Bottou, and Maximilian Nickel. Poincaré maps for analyzing complex hierarchies in single-cell data. *Nature communications*, 11(1):2966, 2020.

[15] Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[16] Yoshiki Kuramoto. Diffusion-Induced Chaos in Reaction Systems. *Progress of Theoretical Physics Supplement*, 64:346–367, 02 1978.

[17] John A. Lee and Michel Verleysen. Scale-independent quality criteria for dimensionality reduction. *Pattern Recognition Letters*, 31(14):2248–2257, October 2010.

[18] Shane Lubold, Arun G Chandrasekhar, and Tyler H McCormick. Identifying the latent space geometry of network models through analysis of curvature. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(2):240–292, 2023.

[19] Samuel E. Otto and Clarence W. Rowley. Linearly Recurrent Autoencoder Networks for Learning Dynamics. *SIAM Journal on Applied Dynamical Systems*, 18, January 2019.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[21] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 701–710. Association for Computing Machinery, 2014.

[22] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[23] Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar. Karate Club: An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*, page 3125–3132. ACM, 2020.

[24] Joshua B Tenenbaum, Vin de Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[25] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. Graph attention networks. *stat*, 1050(20):10–48550, 2017.

[26] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019.

[27] Melanie Weber. Neighborhood growth determines geometric priors for relational representation learning. In *International Conference on Artificial Intelligence and Statistics*, volume 108, pages 266–276, 2020.

[28] Patrick Whalen, Moysey Brio, and Jerome V Moloney. Exponential time-differencing with embedded runge–kutta adaptive step control. *Journal of Computational Physics*, 280:579–601, 2015.

A Methods for Geometric Representation Learning

Table 2: Overview of Geometric Representation Learning methods.

Algorithm	Type	Geometry preserved	Method	Embedding Dimension
PCA	spectral	global proximity	top eigenvalues	learnt
MDS	spectral	global isometry	top eigenvalues	learnt
ISOMAP	spectral	global isometry	top eigenvalues	learnt
LLE	spectral	conformal, local poximity	bottom eigenvalues	hyperparameter
SE	shallow	local proximity	probabilistic	hyperparameter
DEEPWALK	shallow	local proximity	probabilistic	hyperparameter
GCN	deep	local proximity	probabilistic	hyperparameter
GRAPHSAGE	deep	local proximity	probabilistic	hyperparameter
GAT	deep	local proximity	probabilistic	hyperparameter

The methods considered in this study identify geometric structure in a purely data-driven manner, without prior characterization of the data's geometry. We note that there is a complementary body of literature, which provides tools for characterizing data geometry intrinsically, independent of learned data representations [5, 18, 27].

A.1 Classical Manifold Learning algorithms

Multi-Dimensional Scaling (MDS). Multi-dimensional scaling is a popular method for Representation Learning in both Euclidean (cMDS [15]) and non-Euclidean spaces (gMDS [4]). It computes a map ϕ by directly minimizing the representation error, i.e., the difference between pairwise distances $d_{\mathcal{X}}(x, y)$ in the data space and the distances of the corresponding manifold representations $d_{\widetilde{\mathcal{M}}}(\phi(x), \phi(y))$ over all pairs $x, y \in \mathcal{X}$.

Isometric Feature Mapping (Isomap). Isomap [24] is an extension of of MDS, which approximates geodesic distances via shortest paths on the constructed proximity graph and then performs an MDS step.

Locally Linear Embeddings (LLE). The focus of local manifold learning methods, such as *Locally Linear Embeddings* (LLE) [22], is on preserving local similarity structure. A local similarity matrix is constructed with respect to the k nearest neighbors of a data point from which local structure is extracted using spectral decomposition.

Stochastic Neighbor Embeddings (SE). SE [10] aims to preserve the local neighborhood stucture of a point from data space to embedding space. Similarities are measured probabilistically.

A.2 Deep Learning approaches

DeepWalk. DeepWalk [21] is an early approach for shallow graph embeddings. It learns a latent space representation of nodes based utilizing a random walk based similarity measure.

Graph Convolutional Networks and GraphSage. One of the early and still widely popular GNN architectures are GCN [12] and GraphSage [9], both of which extend the idea of convolutional neural networks to the graph domain. Previous applications of graph convolutions to point clouds were considered in [26].

Graph Attention. Graph attention, which translates the idea of *transformers* to Graph Neural Networks, was first proposed by [25]. The most popular variants of Graph Attention Networks are based on this architecture.

B Evaluation Representation Quality

Q-scores, originally introduced by [17], are ranking losses which evaluate how well pairwise distances are preserved locally and globally by an embedding ϕ. Q-scores are computed with respect to a *co-ranking matrix* $Q = \{q_{kl}\}_{k,l \in [N]}$, defined by

$$
\begin{aligned}
q_{kl} &:= |\{(k,l):\ \mu_{ij} = k, \nu_{ij} = l\}| \\
\mu_{ij} &:= |\{k:\ d_{\mathcal{X}}(x_i, x_k) < d_{\mathcal{X}}(x_i, x_j) \text{ and } \{d_{\mathcal{X}}(x_i, x_k) = d_{\mathcal{X}}(x_i, x_j) \text{ if } 1 \leq k < j \leq N\}\}| \\
\nu_{ij} &:= |\{k:\ d_{\mathcal{M}}(x_i, x_k) < d_{\mathcal{M}}(\phi(x_i), \phi(x_j)), \{d_{\mathcal{M}}(\phi(x_i), \phi(x_k)) = d_{\mathcal{M}}(\phi(x_i), \phi(x_j))\}_{1\leq k<j\leq N}|
\end{aligned}
$$

As usual, $|I|$, denotes the cardinality of a set I. We define Q-curves as

$$Q(\phi, K) := \frac{1}{KN} \sum_{(k,l)\in I_K} q_{kl} \ , \tag{4}$$

which are computed over blocks $I_K := \{1, \dots, K\} \times \{1, \dots, K\}$ of the co-ranking matrix. Note that we only need to consider the upper half of I_K, due to symmetry. With respect to Q-curves, we compute local and global Q-scores of ϕ:

$$K_{\max} := \operatorname{argmax}_K \left(Q(\phi, K) - \frac{K}{N-1} \right) \tag{5}$$

$$Q_{local}(\phi) := \frac{1}{K_{\max}} \sum_{K=1}^{K_{\max}} Q(\phi, K) \tag{6}$$

$$Q_{global}(\phi) := \frac{1}{N - K_{\max}} \sum_{K=K_{\max}}^{N-1} Q(\phi, K) \ . \tag{7}$$

Here, the local and global regimes are distinguished by a split of the Q-curve at $K_{\max}$. Q_{local} and Q_{global} attain values from 0 (worst) to 1 (best). In practise to obtain these metric values we first compute the co-ranking matrix Q which is based on the apparoach used in [14].

C Implementation of Classical Manifold Learning

All of the classical models considered in this study (LLE, ISOMAP, SPECTRAL EMBEDDING, MDS, PCA) were implemented using the SCIKIT-LEARN python library [20]. The proximity graph was constructed using a k nearest neighbours approach setting $k = 20$.

D Graph based Deep Learning model architecture and parameters

Table 3: Parameters used in GCN, GRAPHSAGE and GAT methods where d_i is the input dimension and d_z is the latent dimension. The "Dimension" column describes the dimension of each layer and the "Activation" column describes the activation functions between layers. The "Epochs" column describes the number of epochs used for training.

Method	Dimension	Activations	Optimizer	Learning Rate	Epochs
GCN	$d_i : 256 : 512 : d_z$	ReLU: ReLU	SGD	10^{-2}	1000
GRAPHSAGE	$d_i : 256 : 512 : d_z$	ReLU : ReLU	SGD	10^{-2}	1000
GAT	$d_i : 256 : 512 : d_z$	ReLU : ReLU	SGD	10^{-2}	1000

For GCN, GRAPHSAGE and GAT we choose a fixed walk length of 5 and use a single negative sample. For GAT we use a single attention head. The parameters of the GNN based methods are given in Table 3. Through our experiments we found that the SGD optimizer yielded more accurate results compared to Adam and found the commonly used learning rate of 10^{-2} to work well for our applications. The DEEPWALK method was implemented via KATECLUB package [23] which we ran for 1000 epochs with a walk length of 50 and a window size of 5. We found that increasing the walk length yielded a better metric score but was more computationally expensive. We did not observe a significant increase in the performance for walk lengths greater than 50.

E Additional Experimental Results

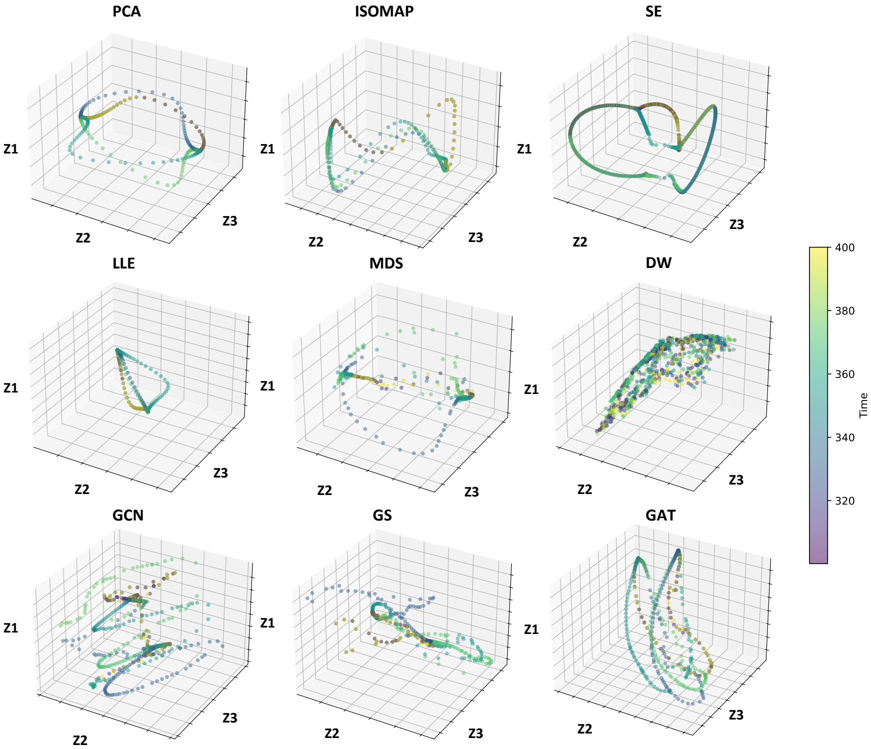

Figure 3: Three-dimensional visualizations of the reduced order dynamics for the KS equation for all of the techniques presented in this study.

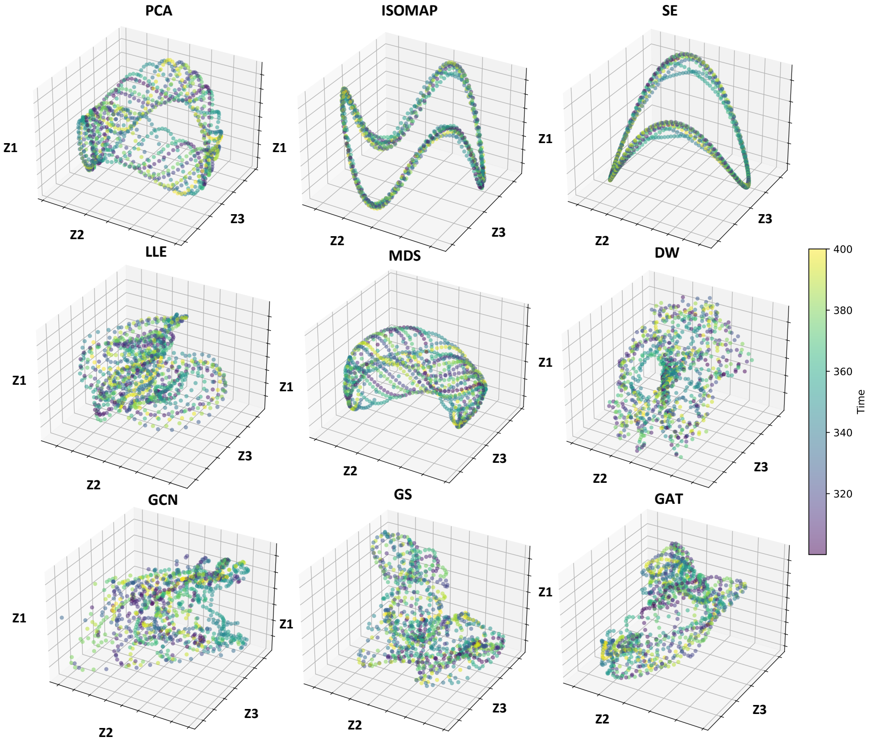

Figure 4: Three-dimensional visualizations of the reduced order dynamics for the fKdV equation for all of the techniques presented in this study.

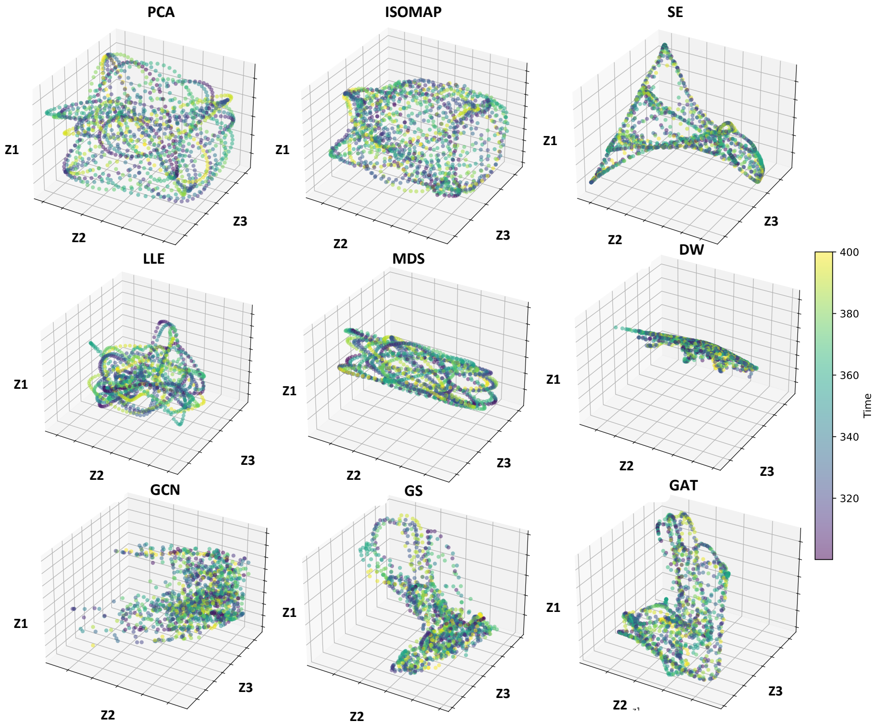

Figure 5: Three-dimensional visualizations of the reduced order dynamics for the SG equation for all of the techniques presented in this study.