

EXPLORING NONLINEAR DIMENSION REDUCTION WITH DIFFUSION MAPS

Susan Oluwatominiyi Kadri^{1,2}, Ranjini Ghosh^{1,2}

¹Group 15, Mathematical Foundations of Data Science, ² Mathematics Department, Georgia Institute of Technology

ABSTRACT

Traditional linear dimensionality reduction methods, such as Principal Component Analysis (PCA), are often limited in their ability to capture nonlinear structure in high-dimensional data. This underscores the need to use non-linear techniques such as Diffusion Maps (DM), Isometric Mapping (Isomap), and Uniform Manifold Approximation and Projection (UMAP). This report explores the use of DM and evaluates its performance using simulated and real-world datasets. The results show that, in some cases, diffusion maps are more effective at revealing nonlinear structure and can outperform PCA. However, in some cases, combining both PCA and Diffusion Maps provided better results.

INTRODUCTION

Dimensionality reduction is a fundamental problem in data analysis and machine learning, as modern datasets are often high-dimensional while their meaningful structure lies on a much lower-dimensional manifold (Lee and Verleysen, 2007; van der Maaten et al., 2009). The goal of dimensionality reduction is to recover compact representations that preserve essential geometric and statistical properties.

Principal Component Analysis (PCA) is one of the most widely used techniques for this purpose. Initially introduced by Hotelling (1933) and later formalized by Jolliffe and Cadima (2016), PCA identifies orthogonal directions of maximum variance and projects the data onto a low-dimensional linear subspace. PCA is computationally efficient and widely used for visualization, denoising, and preprocessing. However, because PCA is inherently linear, it often fails to preserve intrinsic geometry when data lie on nonlinear manifolds.

To address these limitations, nonlinear manifold learning methods have been developed. Among the most prominent are Isomap, Laplacian Eigenmaps, and Diffusion Maps (Tenenbaum et al., 2000; Belkin and Niyogi, 2003; Coifman and Lafon, 2006). These methods rely on graph-based representations and spectral decompositions to preserve local neighborhood structure rather than global linear variance.

Isomap preserves global nonlinear structure by approximating geodesic distances using shortest paths on a neighborhood graph (Tenenbaum et al., 2000). While effective under ideal sampling conditions, Isomap is sensitive to noise and graph connectivity (de Silva and Tenenbaum, 2003). In contrast, Diffusion Maps construct a Markov random walk on a similarity graph and derive a low-dimensional embedding from the eigenvectors of the diffusion operator (Coifman and Lafon, 2006). The resulting diffusion distance measures connectivity through all possible paths and is therefore more robust to noise than shortest-path methods (Nadler et al., 2006). Diffusion Maps also provide a multiscale representation by varying the diffusion time parameter.

Recent work has demonstrated the effectiveness of Diffusion Maps in complex real-world settings, including biological and image-based data (Coifman et al., 2005; Moon et al., 2019). In contrast, PCA remains purely variance-based and does not incorporate neighborhood geometry, limiting its ability to capture nonlinear transitions.

In this work, we compared PCA, Diffusion Maps, and Isomap using simulated manifold datasets and real-world classification datasets. The quality of the embeddings was evaluated through visualization, k-Nearest Neighbors (kNN) classification accuracy, and computational efficiency. Our goal was to identify when nonlinear methods, particularly Diffusion Maps, provide clear advantages over linear dimensionality reduction.

DATASETS AND EXPERIMENTAL PROCEDURES

1. SIMULATED DATASETS :

a. Swiss Roll Dataset

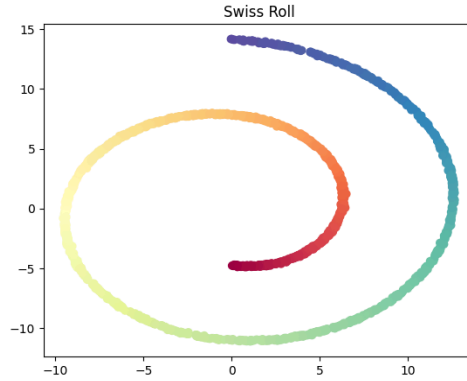


Figure 1. A standardized concentric circle illustrated two nested nonlinear manifolds

The Swiss Roll dataset was generated using the `make_swiss_roll` function from Scikit-Learn. It is a 3D nonlinear manifold commonly used to benchmark nonlinear dimensionality reduction methods. A total of 2000 samples were generated with a noise level of 0.05 to introduce moderate variability along the spiral structure. Each data point lies in \mathbb{R}^3 , but the intrinsic geometry is one-dimensional. To create class labels for evaluation, the intrinsic parameter t returned by `make_swiss_roll` was thresholded at its median, yielding a binary classification problem in which points on one half of the roll are labeled as class 0 and those on the other as class 1. Before applying any distance-based or kernel-based methods, all features were standardized using z-score normalization:

Standardization ensures equal scaling across dimensions and stabilizes Gaussian kernels (used in diffusion maps). The dataset was then split into 80% for training and 20% for testing using stratified sampling to preserve class balance. The same splits were used across all embeddings (raw, PCA, DM complete, and DM kNN) to ensure a fair comparison.

1. ai Dimensionality Reduction

Three methods were applied. PCA was applied to the standardized 3D dataset. Diffusion Maps (Full Gaussian Kernel) were implemented using the whole Gaussian kernel:

$$K_{ij} = \exp\left(\frac{(-||x_i - x_j||^2)}{\epsilon}\right)$$

A kernel scale ϵ was selected based on the median pairwise distance. The kernel was row-normalized to create a Markov transition matrix representing a random walk over the data graph. The dominant non-trivial eigenvectors of this transition matrix produce a nonlinear embedding that preserves diffusion distance, i.e., connectivity over many random walk paths. A kNN graph version was also implemented using $k = 12$ nearest neighbors. The Gaussian kernel was applied only along kNN edges and symmetrized.

1. aii Classification & Runtime Evaluation

Classification was performed using k-nearest neighbors ($k = 5$) across the four representations of the data: the raw 2D projection using the first two standardized features, the PCA embedding, the full-kernel diffusion map embedding, and the kNN kernel diffusion map embedding. Performance was evaluated using stratified 5-fold cross-validation for both training and prediction phases. Confusion matrices were also generated for each method to examine class-specific behavior and identify misclassification locations.

b. S-Curve Dataset

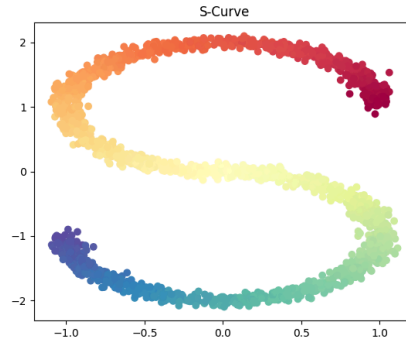


Figure 1.b Standardized S-curve dataset illustrating a smoothly varying nonlinear manifold

The S-Curve dataset was generated using Scikit-Learn's `make_s_curve` function with 2000 samples and a noise level of 0.05. This dataset lies in \mathbb{R}^3 and forms a smooth, nonlinear 2D surface. Class labels were created by thresholding the intrinsic parameter returned by the generator, yielding a balanced binary classification problem. All features were standardized via z-score normalization to stabilize distance metrics and kernel evaluations.

1. bi Dimensionality Reduction

The same three dimensionality reduction techniques were applied as in the Swiss Roll case.

1. bii Classification & Runtime Evaluation

The classification pipeline mirrors the approach used for the Swiss Roll experiment. A kNN classifier with $k=5$ is applied to four different representations of the dataset. Performance is evaluated using 5-fold stratified cross-validation, and confusion matrices and runtime bar charts summarize the results for accuracy comparisons.

c. Concentric Circles:

The final simulated dataset analyzed was the concentric circles. Concentric circles were generated using the `make_circles` function from Scikit-Learn to create a synthetic 2D nonlinear manifold. A sample of 3000 was generated with a noise level of 0.08 to introduce moderate overlap between the classes, and an inner-circle radius factor of 0.5, yielding two nested circular clusters.

Each data point lies in (i.e, a 2D dataset), and the class labels correspond to the inner and outer circles. The data set was then converted to a single-precision floating-point format. All features were standardized using z-score normalization to achieve a mean of 0 and unit variance.

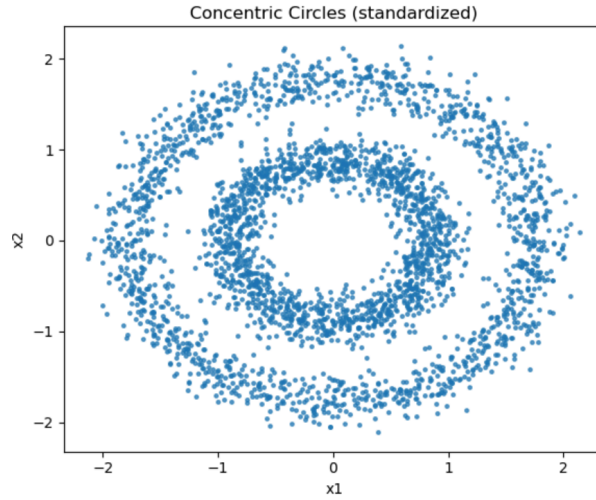


Figure 1.c Standardized concentric circle illustrates two nested nonlinear manifolds

1. ci Dimensionality Reduction

PCA, DM, and a hybrid PCA were applied to this data set. Isomap was also used for geometric visualization. PCA was applied to the standardized data, reducing the dimensionality to two principal components. Since the original data lies in two dimensions, PCA primarily performs a rotation of the original coordinate system. As expected, PCA does not separate two concentric circles because the mapping is nonlinear. DM was then implemented using an adaptive Gaussian kernel. First, the pairwise squared Euclidean distance matrix was computed:

$$D_{ij} = ||x_i - x_j||^2$$

Local kernel bandwidths were estimated from the distance to the seventh-nearest neighbor for each point. The similarity column was defined as:

$$K_{ij} = e^{\left(-\frac{D_{ij}}{\sqrt{\epsilon_i \epsilon_j}}\right)}$$

The kernel was row-normalized to construct a Markov chain transition matrix, thereby defining a random walk on the data graph. The eigenvalues and eigenvectors of this Matrix were computed, and the first two non-trivial diffusion coordinates were used as the non-linear embedding. This construction allowed the

embedding to preserve the fusion distance, which measures connectivity through all possible random marks rather than the direct Euclidean distance.

Isomap was implemented from scratch using a k-nearest neighbor graph with $k=10$. Geodesic distances were approximated using repeated Dijkstra shortest-path computations, followed by classical multidimensional scaling to obtain a two-dimensional Isomap embedding.

1. cii Classification and runtime evaluation

Classification performance was evaluated using KNN with k equal to 5. The classifier was trained and tested on the raw standardized features, PCA-reduced features, and diffusion Maps embedding. Classification accuracy on the test sets, training runtime, and prediction runtime were recorded. All run times were measured using high-resolution performance counters. Confusion matrices were computed to analyze class-specific prediction Behavior.

2. REAL DATASETS:

a. MNIST DIGITS

2. ai Data Description and Preprocessing

The MNIST datasets consist of grayscale images of handwritten digits 0 to 9, each 28×28 pixels. For this project, we used a subset of the full dataset (denoted X_{subset} Y_{subset}), in which each image was flattened into a 784-dimensional vector.

$$X_{raw} \in \mathbb{R}^{n \times 784}$$

If the input array was three-dimensional, it was reshaped to a two-dimensional matrix using:

$$X_{raw} = \text{reshape}(n, 28 \times 28)$$

No additional pixel-level normalization beyond the MNIST scaling provided by the data set loader was applied to the code. The data was finally split into 80% for training and 20% for testing, using stratified sampling to preserve the class distribution across splits. the same train test indices that we use for all representations to enable fair comparison

2. aii Nonlinear Dimension Reduction:

Following the same experimental pipeline established for the simulated dataset, PCA and Diffusion Maps were applied to the MNIST images, with dataset-specific parameter settings. PCA was first applied to the flattened 784-dimensional digit images, yielding a 50-dimensional embedding that served as the primary linear baseline for classification. A second PCA with 100 components was used exclusively as a preprocessing stage for Diffusion Maps to reduce computational cost while preserving most of the image structure.

Diffusion Maps were then applied to the PCA-reduced MNIST data using an adaptive local-scale kernel with $k = 15$. The resulting nonlinear embedding was constructed using the first 10 non-trivial diffusion coordinates, forming a 10-dimensional manifold representation of the digit data.

2. aiii Classification of Runtime Evaluation

To evaluate the quality of each representation, K-nearest Neighbors classification with $k=5$ was applied to the Ross pixel space, PCA space, and diffusion map space. A common stratified train-test was used for all three representations. For each case, the following performance metrics were recorded.

b. EMOJI Dataset

The emoji dataset was constructed from the OpenMoji library of colored emoji icons. All PNG files in the selected folder were loaded and preprocessed using a custom load. It was then opened in RGB format, resized to 32×32 pixels, converted to a Numpy array, and flattened into a 3072-dimensional vector ($32 \times 32 \times 3$). Finally, it was normalized to the range $[0,1]$ by dividing pixel values by 255. The resulting data matrix has the form:

$$X \in \mathbb{R}^{n \times 3072}$$

Where $n < 2000$ is the number of emojis loaded

Class labels were derived from the Unicode hex code in each filename. The first code point was extracted and mapped into coarse semantic categories such as smiley symbols, flags, and others. This produced a label vector

2. bi Non-Linear Dimension Reduction

PCA was applied directly to the flattened emoji vectors, reducing the data to a two-dimensional embedding that was used for both visualization and as a linear baseline for classification.

Diffusion Maps were applied manually to the high-dimensional emoji features using a global Gaussian kernel, with the kernel bandwidth ϵ set to the median of the off-diagonal squared distances. The resulting nonlinear embedding was constructed using the first two non-trivial diffusion coordinates, producing a two-dimensional diffusion representation.

Isomap was implemented from scratch using a kNN with $k=10$. Geodesic distances were approximated using repeated Dijkstra shortest-path computations, followed by classical multidimensional scaling to obtain a two-dimensional Isomap embedding.

2. bii Classification and Evaluation

To compare the representations, a k-NN classifier with $k=5$ was implemented manually. For each representation (PCA, Diffusion Maps, Isomap), the same train-test indices were used. Classification was performed in the corresponding 2D embedding space, and accuracy was computed as the proportion of correctly classified test samples.

c. Breast cancer

2. ci Data Description and Preprocessing

The Breast Cancer Wisconsin (Diagnostic) dataset was obtained using Scikit-Learn's built-in `load_breast_cancer()` function. The dataset contains 569 samples. Each corresponds to a digitized image

of a breast mass. Here, each sample is represented by 30 real-valued features extracted from the original medical photos. These features include radius, texture, smoothness, compactness, and concavity. Thus, each patient sample lies in \mathbb{R}^{30} . Therefore, it is a moderately high-dimensional tabular dataset. The target labels are binary, distinguishing between *malignant* and *benign* tumors.

Before applying any dimensionality reduction or classification, all features were standardized using z-score normalization:

$$X_{std} = \frac{(X-\mu)}{\sigma}$$

The standardized dataset was used consistently across all embeddings.

2. cii Nonlinear Dimension Reduction

Three embeddings were computed exactly as in the synthetic datasets: PCA, diffusion maps with a whole Gaussian kernel and diffusion maps with a sparse kNN kernel. PCA was applied to the 30-dimensional feature vectors. This reduced them to a 2-dimensional embedding for visualization and classification.

PCA was used as a baseline embedding for classification, a linear benchmark against nonlinear methods and a way to inspect the global variance structure in the data. Diffusion Maps were applied directly to the standardized 30-dimensional data using a Gaussian kernel.

To correct for sampling density bias, we applied α -normalization with $\alpha = 0.5$:

$$K_{\alpha}(i, j) = \frac{K_{ij}}{q_i^{\alpha} q_j^{\alpha}}, \quad q_i = \sum_j K_{ij}$$

The normalized kernel was row-normalized to form a Markov transition matrix:

$$P = \frac{K_{\alpha}}{\sum_j K_{\alpha}(i, j)}$$

Eigen-decomposition of P yields diffusion eigenvalues $\lambda \ell$ and eigenvectors $\psi \ell$:

$$P \psi \ell = \lambda \ell \psi \ell$$

The constant eigenvector $\psi 0$ is removed, and the subsequent two non-trivial eigenvectors were used to build the 2-dimensional diffusion embedding. A second nonlinear embedding was constructed by constructing a symmetric k-nearest-neighbor graph ($k = 12$) before applying the Gaussian kernel. The same α -normalization and row-stochastic normalization steps were used to produce a diffusion operator whose eigenvectors form the kNN diffusion embedding.

2. ciii Classification and Runtime Evaluation

To assess the effectiveness of each representation, k-nearest neighbors classification ($k = 5$) was applied to 3 embeddings. A stratified 5-fold cross-validation procedure was used to evaluate classification in a controlled and consistent manner. For each embedding, we recorded mean cross-validated accuracy, confusion matrices, and average training and testing time (fit time for kNN).

RESULTS AND ANALYSIS

1. SIMULATED DATASETS :

a. Swiss Roll Dataset

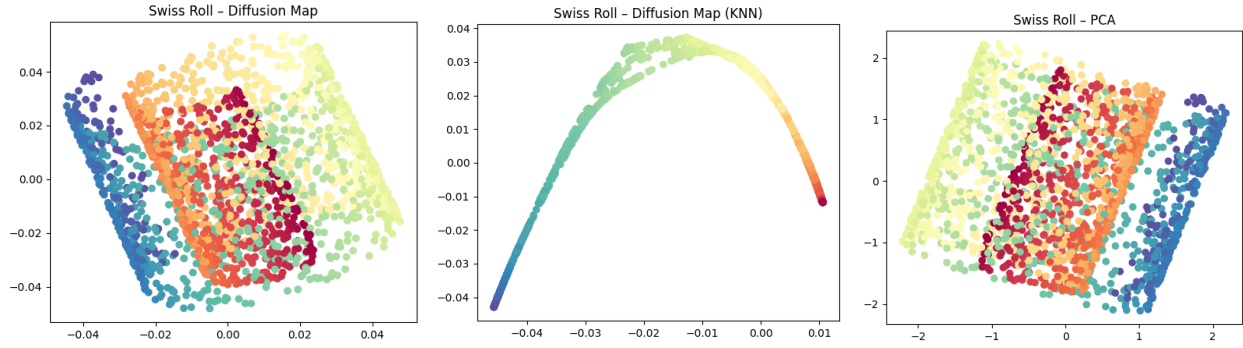


Figure 1.ai diffusion map (full kernel), diffusion map (KNN) and PCS for swiss roll

The diffusion map scatterplot shows the first two non-trivial diffusion coordinates computed using the whole Gaussian kernel. This embedding partially unrolls the Swiss Roll. However, there is some color-band mixing and blurring. We observe that full-kernel diffusion maps tend to oversmooth the geometry because every point influences every other point.

The kNN-based diffusion map, by contrast, employs a locally restricted kernel. It cleanly unwraps the roll into a smooth 1D curve, thus preserving its structure. This locally adaptive approach avoids long-range kernel noise and captures the manifold structure more accurately, which explains why DM-kNN delivers the strongest classification performance and the fastest runtime.

PCA cannot recover its nonlinear geometry, as we see significant color mixing. This demonstrates that PCA fails to preserve intrinsic manifold structure. This confirms that PCA is inadequate for nonlinear manifolds and it emphasizes the need for diffusion maps in such scenarios.

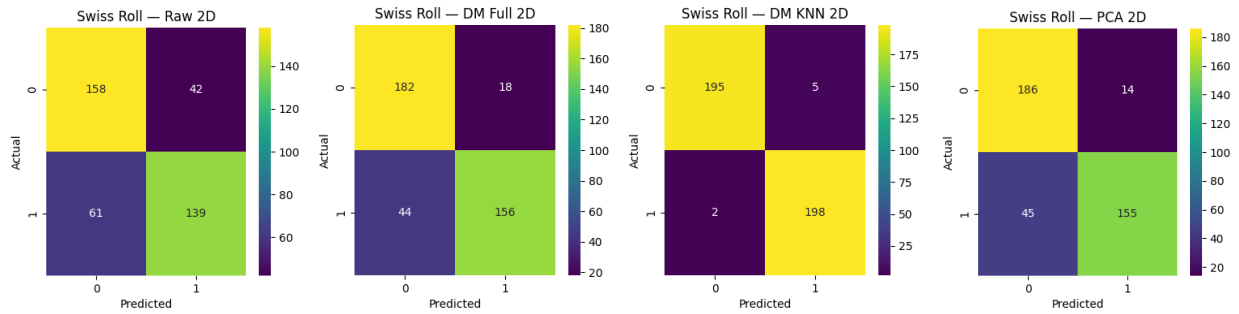


Figure 1.a.ii confusion matrices (raw data, diffusion map-full, diffusion map-KNN, PCA) for swiss roll

These confusion matrices compare the performance of KNN in classifying the two Swiss Roll classes across different embeddings. In the raw 2D projection, accuracy is lowest because the manifold is still curved. Points that are far apart on the surface appear close in Euclidean space, leading to class mixing. With the full-kernel diffusion map, misclassification drops. The kNN-based diffusion map performs even better. It produces the strongest separation with extremely low error. PCA shows improved performance relative to the raw projection; however, because it is a linear technique, some class overlap remains. We conclude that diffusion maps (especially the kNN kernel) yield the most pronounced class separation.

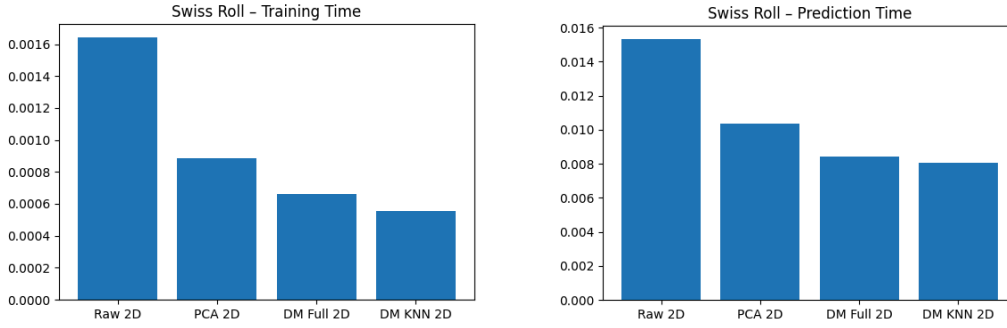


Figure 1.biii comparing kNN training and prediction times for swiss roll

When it comes to KNN training time, raw 2D is slowest because the curved Swiss Roll geometry creates confusing neighborhoods. Diffusion map embeddings are fastest since they smooth and unroll the data.

For prediction time, raw 2D again performs worst due to the dataset's nonlinear shape. PCA improves speed but still carries curvature issues. Both diffusion-map embeddings are fastest because unrolling the manifold makes Euclidean distances meaningful, simplifying nearest-neighbor search. Overall, nonlinear embeddings such as diffusion maps not only improve accuracy but also reduce KNN computation time.

b. S-Curve Dataset

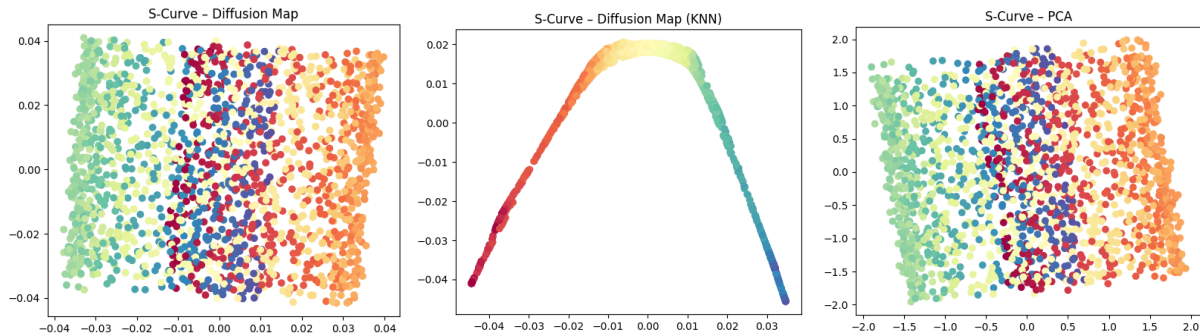


Figure 1.bi diffusion map (full kernel), diffusion map (KNN), and PCS for S-curve

The diffusion map embedding (KNN) successfully unfolds the S-curve into a smooth, continuous low-dimensional structure, whereas PCA fails to capture the nonlinear curvature, as we see an overlapping and distorted projection.

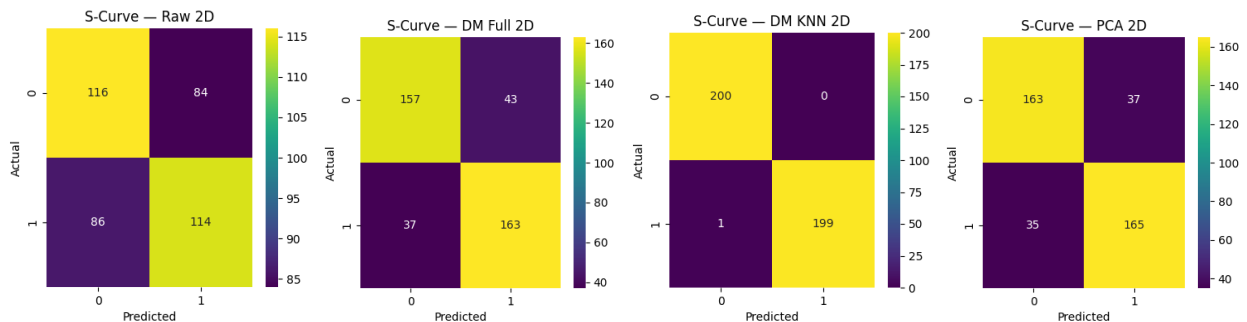


Figure 1.bii confusion matrices (raw data, diffusion map-full, diffusion map-KNN, PCA) for S-curve

The diffusion-based embeddings show more precise class boundaries, whereas raw 2D and PCA embeddings show more misclassifications.

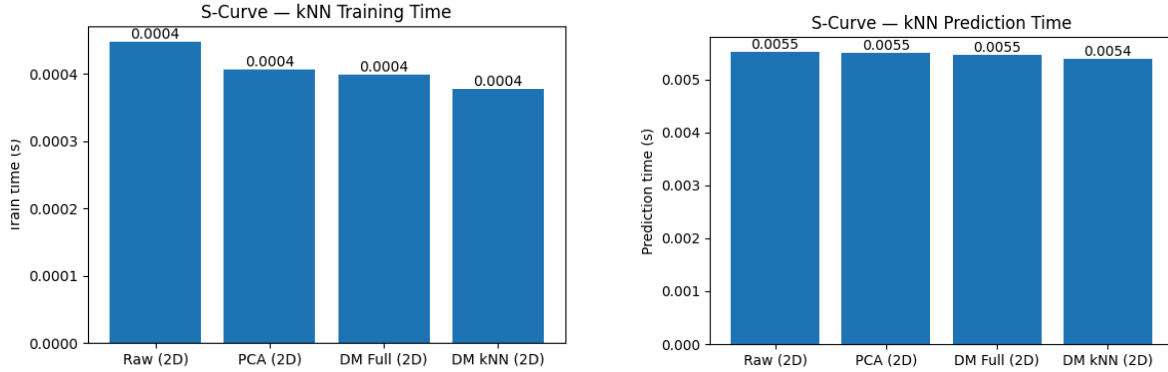


Figure 1.biii comparing kNN training and prediction times for S-curve

c. Concentric Circles:

The PCA embedding preserves global variance but fails to unfold the nonlinear structure of the data fully. As shown in Figure 1.ci, the two classes remain overlapping in the projected space, confirming that linear projections cannot separate nested manifolds.

In contrast, the PCA → Adaptive Diffusion Maps embedding successfully recovers the intrinsic geometry of the dataset. In Figure 1. ci, the inner and outer rings become cleanly separated in diffusion space. This demonstrates that Diffusion Maps correctly preserve geodesic connectivity and unfold the nonlinear manifold structure.

Isomap also separates the two rings, to the left of Figure 1. ci, but the resulting embedding exhibits noticeable geometric distortion and irregular point spacing reflecting Isomap's sensitivity to graph construction and its reliance on shortest-path estimates, which can become unstable in the presence of noise.

Overall, Diffusion Maps produce the most geometrically faithful nonlinear embedding among the three methods.

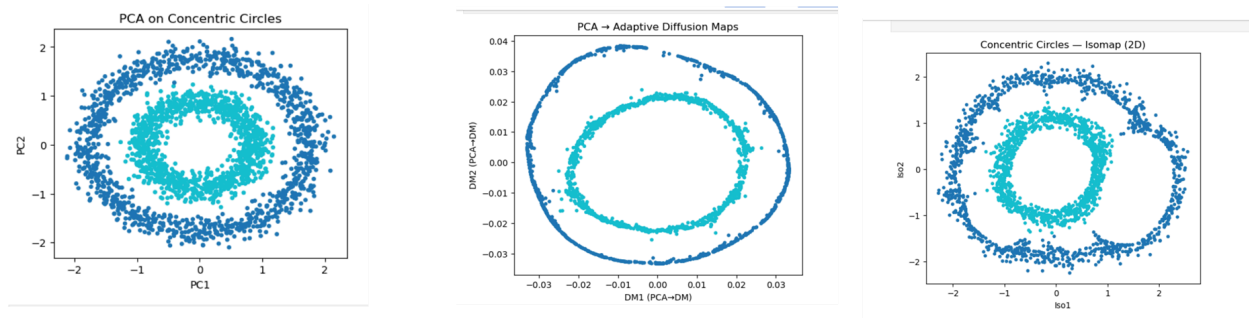


Figure 1. ci shows PCA, PCA- Adaptive Diffusion Maps, and Isomap for concentric circles.

The confusion matrices for k-Nearest Neighbors classification ($k = 5$) in the raw 2D space, PCA space, and Diffusion Maps space are shown in Figure 1. cii. All three representations achieve perfect classification accuracy (100%), with no misclassifications in either class. This result indicates that, despite PCA's geometric limitations, the concentric circles dataset is sufficiently structured to allow kNN to separate the classes even in the linear projection. However, this perfect accuracy does not reflect the quality of geometric manifold recovery, which is more accurately captured by Diffusion Maps.

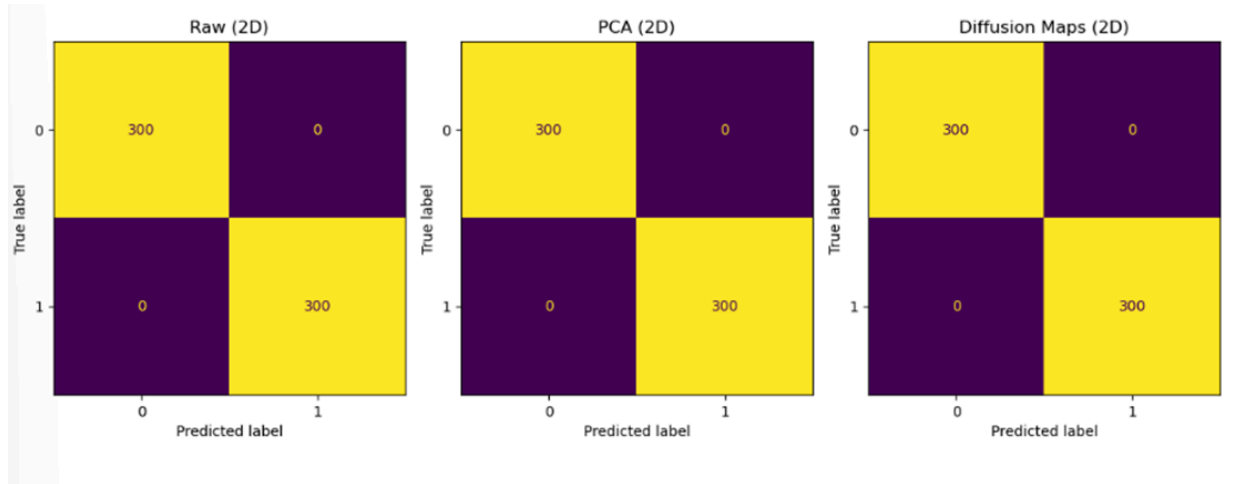


Figure 1. cii shows the confusion matrix for the Original Dataset, PCA, and Diffusion Maps for concentric circles.

Figure 1. ciii reports the kNN prediction and training times for the three representations. The raw representation exhibits the highest computational cost for both training and prediction. PCA and Diffusion Maps substantially reduce computational time due to dimensionality reduction.

Among the reduced representations, Diffusion Maps achieve the shortest prediction time, whereas PCA exhibits slightly shorter training time. These results demonstrate that nonlinear embeddings can improve computational efficiency without sacrificing classification performance.

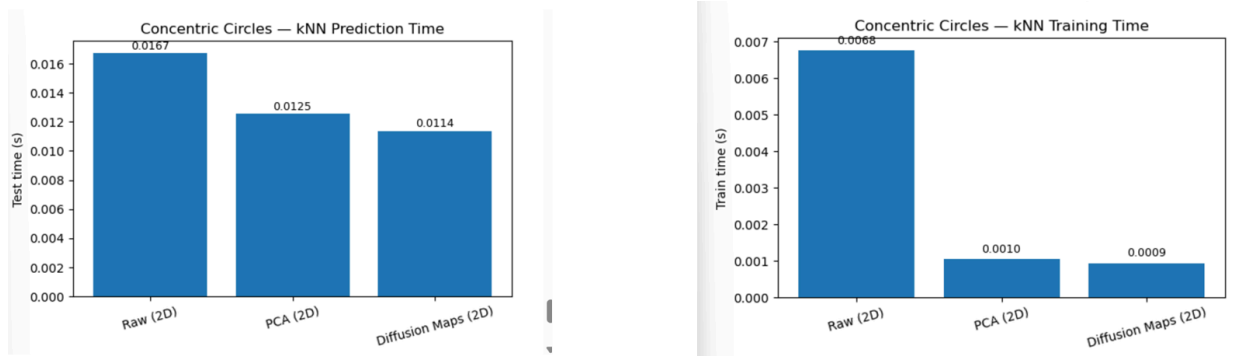


Figure 1. ciii shows the barplot of kNN prediction time and kNN training time for Concentric Circles.

2. REAL DATASETS :

a. MNIST DIGITS:

The PCA projection onto the first two principal components, shown to the left of Figure 2ai , preserves global variance but exhibits substantial class overlap. While some weak clustering of digits is visible, most digit classes remain highly entangled, indicating that PCA fails to capture the nonlinear structure of the digit manifold.

The PCA -Adaptive Diffusion Maps embedding is shown in the middle of Figure 2.ai. ai reveals a significant improvement in class organization. Digits organize along smooth curved trajectories, with several classes becoming visibly separated. This indicates that diffusion geometry successfully captures transitions between digit shapes that are invisible to linear projections.

When Diffusion Maps are applied directly without PCA preprocessing at the far right of Figure 2.ai, the embedding becomes highly distorted. The geometry collapses into elongated structures with excessive curvature and uneven density, indicating kernel instability and poor scaling in the original 784-dimensional pixel space, confirming that PCA preprocessing is essential for stable diffusion behavior on high-dimensional image data.

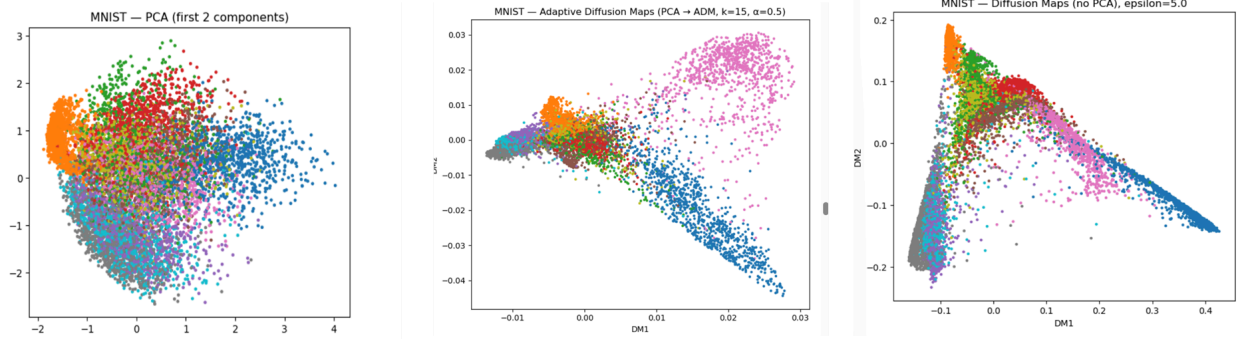


Figure 2 ai shows PCA, PCA- Adaptive Diffusion Maps, and Diffusion maps for MNIST Digits

Figure 2. aii displays the first three nontrivial diffusion coordinates (ψ_1, ψ_2, ψ_3). Each coordinate reveals a smooth gradient across the data cloud, indicating gradual variation in digit structure rather than abrupt class boundaries. This confirms that Diffusion Maps encode continuous morphological transitions between digits rather than merely discrete class separation.

Figure 2. aii presents the eigenvalue spectrum of the diffusion operator. The rapidly decaying eigenvalues indicate that only a small number of diffusion coordinates capture the dominant intrinsic structure of the dataset. This supports the use of low-dimensional diffusion embeddings for downstream tasks.

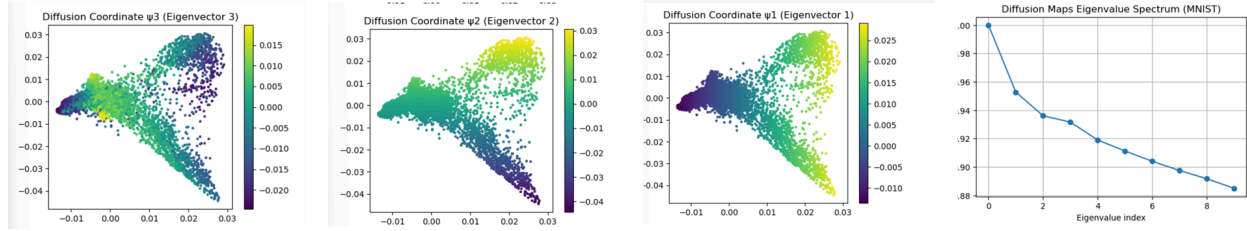


Figure 2 ai shows the diffusion coordinates 1-3 and the Eigenvalue Spectrum.

b. EMOJI Dataset:

The PCA embedding on the left side of Figure 2.bi spreads the data along the two directions with the most significant variance, but the overall structure remains highly mixed. Emoji categories overlap almost entirely, showing that PCA does not capture the nonlinear visual features that distinguish different groups.

The Diffusion Maps embedding in the middle of Figure 2.bi produces a noticeably clearer organization. Several categories begin to cluster together, and the layout reflects similarities in visual design rather than simple pixel intensity. Although the separation is not perfect, the embedding captures more meaningful relationships among emojis compared to PCA.

The Isomap result to the right of Figure 2bi is more irregular. The embedding is stretched along one axis, and the category structure is less stable. This is consistent with the fact that Isomap relies heavily on the quality of the neighborhood graph, which is challenging to construct accurately in very high-dimensional image space.

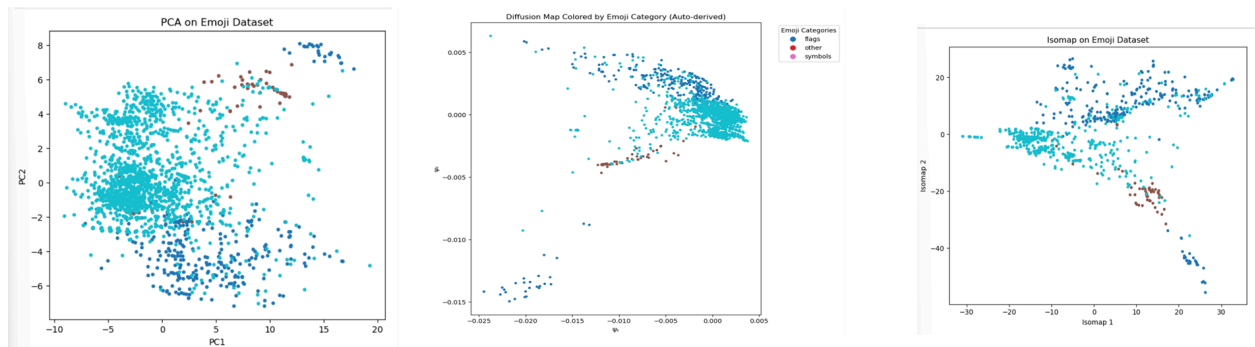


Figure 2. ai shows PCA, Diffusion Maps, and Isomap for the Emoji Dataset

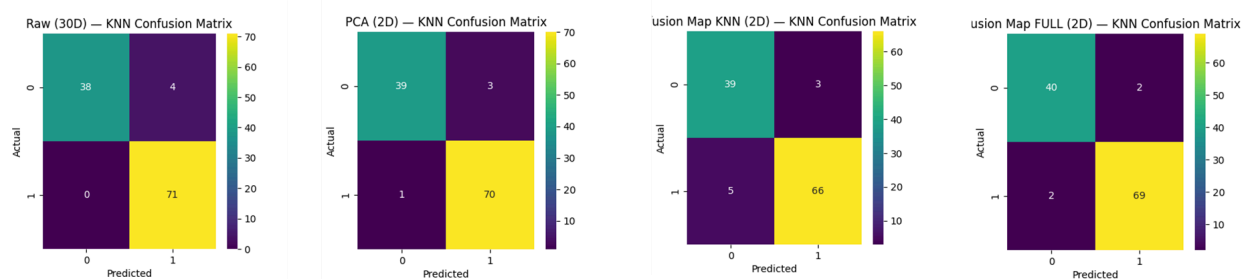


Figure 2. bi i shows the Confusion Matrix for the dimension reduction

Using the raw pixel features, kNN achieves strong accuracy, but a few misclassifications remain likely due to similarities between certain emoji types.

After applying PCA, the classification accuracy remains close to that of the raw features, suggesting that PCA preserves sufficient information for coarse-level classification, even though it does not improve the geometric separation observed in the visualization.

The Diffusion Maps embedding exhibits slightly higher confusion for one category. This is expected because diffusion maps emphasize manifold geometry rather than class boundaries.

When the complete set of diffusion coordinates is used, the classifier performs more consistently, producing one of the cleanest diagonal confusion matrices. This indicates that additional diffusion coordinates facilitate the recovery of proper structure for classification.

c. Breast cancer:

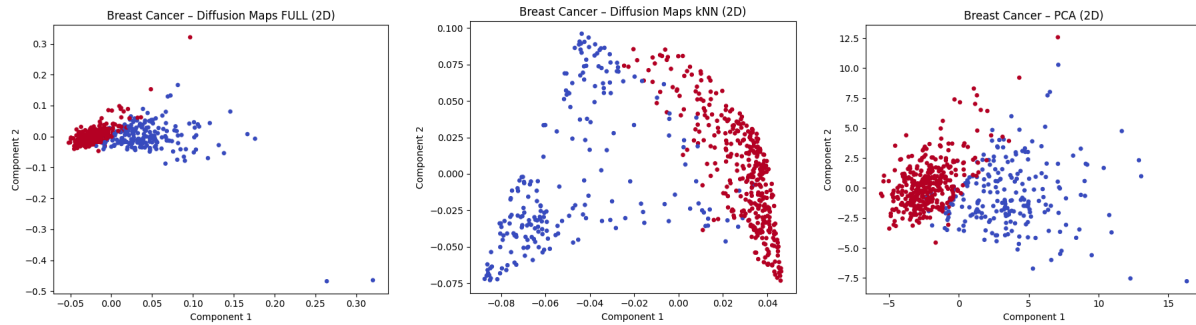


Figure 2.ci diffusion map (full kernel), diffusion map (kNN), PCA for breast cancer dataset

The first graph shows the 2D embedding produced by the full diffusion map kernel. Here, the breast cancer data forms two separated clusters, although the structure is slightly noisy. The kNN version produces a similar overall shape but with more spread along the axes, resulting in somewhat less stable grouping. PCA actually performs quite well. This is because the breast cancer dataset is less nonlinear than datasets such as Swiss Roll or S-Curve.

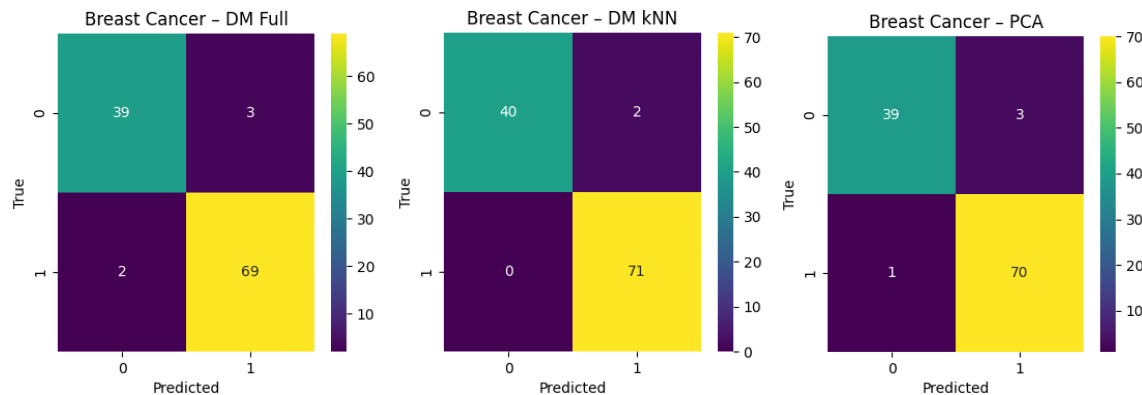


Figure 2.cii confusion matrices (raw data, diffusion map-full, diffusion map-KNN, PCA) for S-curve

The confusion matrix for the full diffusion map embedding shows strong classification performance with only a few misclassifications. The kNN diffusion map confusion matrix performs slightly better. The PCA confusion matrix also shows high accuracy. Again, this indicates that the breast cancer dataset is closer to linearly separable, which is why PCA is more effective here than on strongly nonlinear datasets.

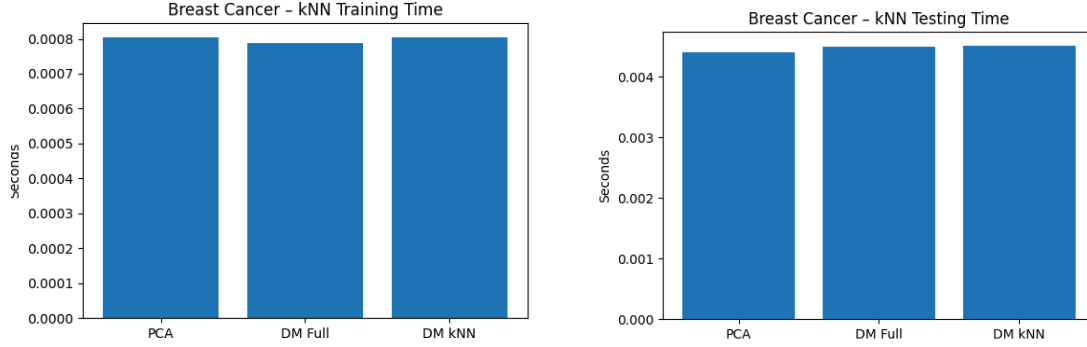
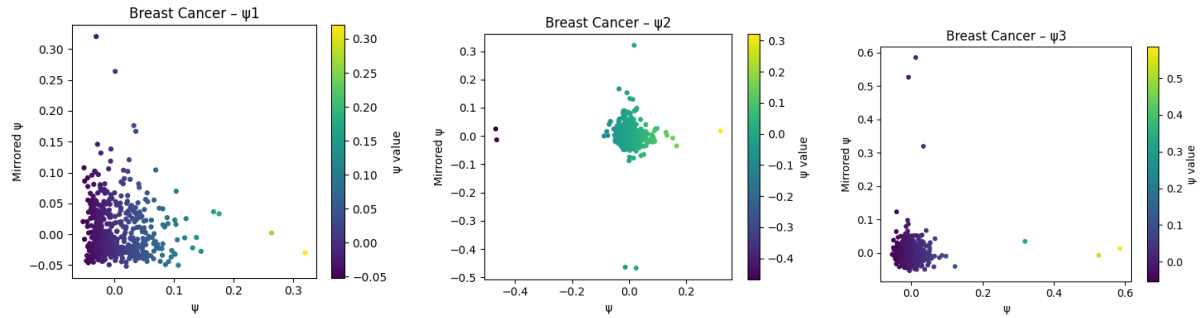
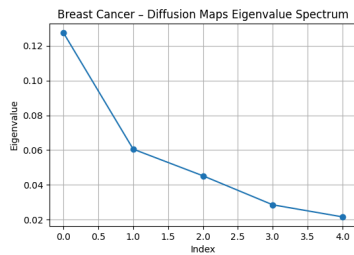


Figure 2.ciii comparing kNN training and prediction times for S-curve

All three methods allow extremely fast training and prediction.



ψ_1 is the first non-trivial diffusion coordinate and represents the strongest nonlinear direction in the dataset. The plot shows a smooth spread of values, meaning ψ_1 captures a major global trend in the data. ψ_2 captures a second independent direction of variation. Most points cluster more tightly here. Thus, ψ_2 represents finer-scale structure than ψ_1 . ψ_3 represents an even subtler geometric pattern. The values are more compressed. So, ψ_3 carries less essential information than ψ_1 and ψ_2 .



The eigenvalue spectrum drops quickly after the first few eigenvalues, showing that only a small number of diffusion coordinates, mainly ψ_1 and ψ_2 are needed to represent the dataset's intrinsic structure. This validates the use of a 2D diffusion map embedding.

CONCLUSION

Overall, our findings show that diffusion maps are a powerful nonlinear dimensionality-reduction method that consistently outperforms PCA when the underlying data lie on a curved or nonlinear manifold. Diffusion maps capture nonlinear structure more effectively and preserve the intrinsic geometry of the data even when reduced to very low dimensions. We also observed that the quality of the diffusion embedding is sensitive to the complexity and noise level of the dataset; clean, well-structured manifolds produce clearer embeddings, while noisier datasets require more careful parameter choices. Classification accuracy closely mirrored the extent to which each method represented the manifold, with stronger embeddings directly leading to better performance. However, diffusion maps are limited by high computational cost for large datasets, primarily because they require computing all pairwise distances.

TASK DIVISION

- 1) Emoji, MNIST, Concentric Circles - Susan Kadri
- 2) Swiss Roll, Scurve, Breast Cancer - Ranjini Ghosh

Statement of AI use in Report and Code

Generative AI tools were used to assist with debugging code, explaining algorithms, and formatting report sections. All experiments and implementations were performed manually.

REFERENCES

1. *Sklearn.datasets.load_breast_cancer* — Scikit-Learn 0.24.1 Documentation.” Scikit-Learn.org, scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html.
2. “Digit Recognizer.” Kaggle.com, www.kaggle.com/c/digit-recognizer/data. Subinium.
3. “[Emoji] FULL EMOJI DATASET 🇲🇪 🇲🇪 🇲🇪” Kaggle.com, Kaggle, 3 Apr. 2021, www.kaggle.com/code/subinium/emoji-full-emoji-dataset. Accessed 23 Sept. 2025.
4. “Make_swiss_roll.” Scikit-Learn, 2025, scikit-learn.org/stable/modules/generated/sklearn.datasets.make_swiss_roll.html.
“Make_s_curve.” Scikit-Learn, 2025, scikit-learn.org/stable/modules/generated/sklearn.datasets.make_s_curve.html.
5. “Sklearn.datasets.make_circles.” Scikit-Learn, scikit-learn.org/stable/modules/generated/sklearn.datasets.make_circles.html.
6. Coifman, R. R., & Lafon, S. (2006). Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1), 5–30.
7. Belkin, Mikhail, and Partha Niyogi. “Laplacian Eigenmaps for Dimensionality Reduction and Data Representation.” *Neural Computation*, vol. 15, no. 6, June 2003, pp.

1373–1396, www2.imm.dtu.dk/projects/manifold/Papers/Laplacian.pdf,
<https://doi.org/10.1162/089976603321780317>.

8. Coifman, R. R., et al. “Geometric Diffusions as a Tool for Harmonic Analysis and Structure Definition of Data: Diffusion Maps.” *Proceedings of the National Academy of Sciences*, vol. 102, no. 21, 17 May 2005, pp. 7426–7431, <https://doi.org/10.1073/pnas.0500334102>.
9. Wang, Qinggang, and Jianwei Li. “Combining Local and Global Information for Nonlinear Dimensionality Reduction.” *Neurocomputing*, vol. 72, no. 10-12, June 2009, pp. 2235–2241, <https://doi.org/10.1016/j.neucom.2009.01.006>.
10. Tenenbaum, J. B. “A Global Geometric Framework for Nonlinear Dimensionality Reduction.” *Science*, vol. 290, no. 5500, 22 Dec. 2000, pp. 2319–2323, <https://doi.org/10.1126/science.290.5500.2319>. Accessed 5 July 2020.
11. Nadler, Boaz, et al. “Diffusion Maps, Spectral Clustering and Reaction Coordinates of Dynamical Systems.” *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, July 2006, pp. 113–127, <https://doi.org/10.1016/j.acha.2005.07.004>. Accessed 27 Oct. 2021.
12. Moon, Kevin R., et al. “Visualizing Structure and Transitions in High-Dimensional Biological Data.” *Nature Biotechnology*, vol. 37, no. 12, 1 Dec. 2019, pp. 1482–1492, www.nature.com/articles/s41587-019-0336-3,
<https://doi.org/10.1038/s41587-019-0336-3>. Accessed 16 Feb. 2021.
13. Jolliffe, Ian T., and Jorge Cadima. “Principal Component Analysis: A Review and Recent Developments.” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, 13 Apr. 2016, p. 20150202, royalsocietypublishing.org/doi/10.1098/rsta.2015.0202,
<https://doi.org/10.1098/rsta.2015.0202>.
14. OpenAI. *ChatGPT*, version GPT-5.1, OpenAI, 2025, <https://chat.openai.com/>.

