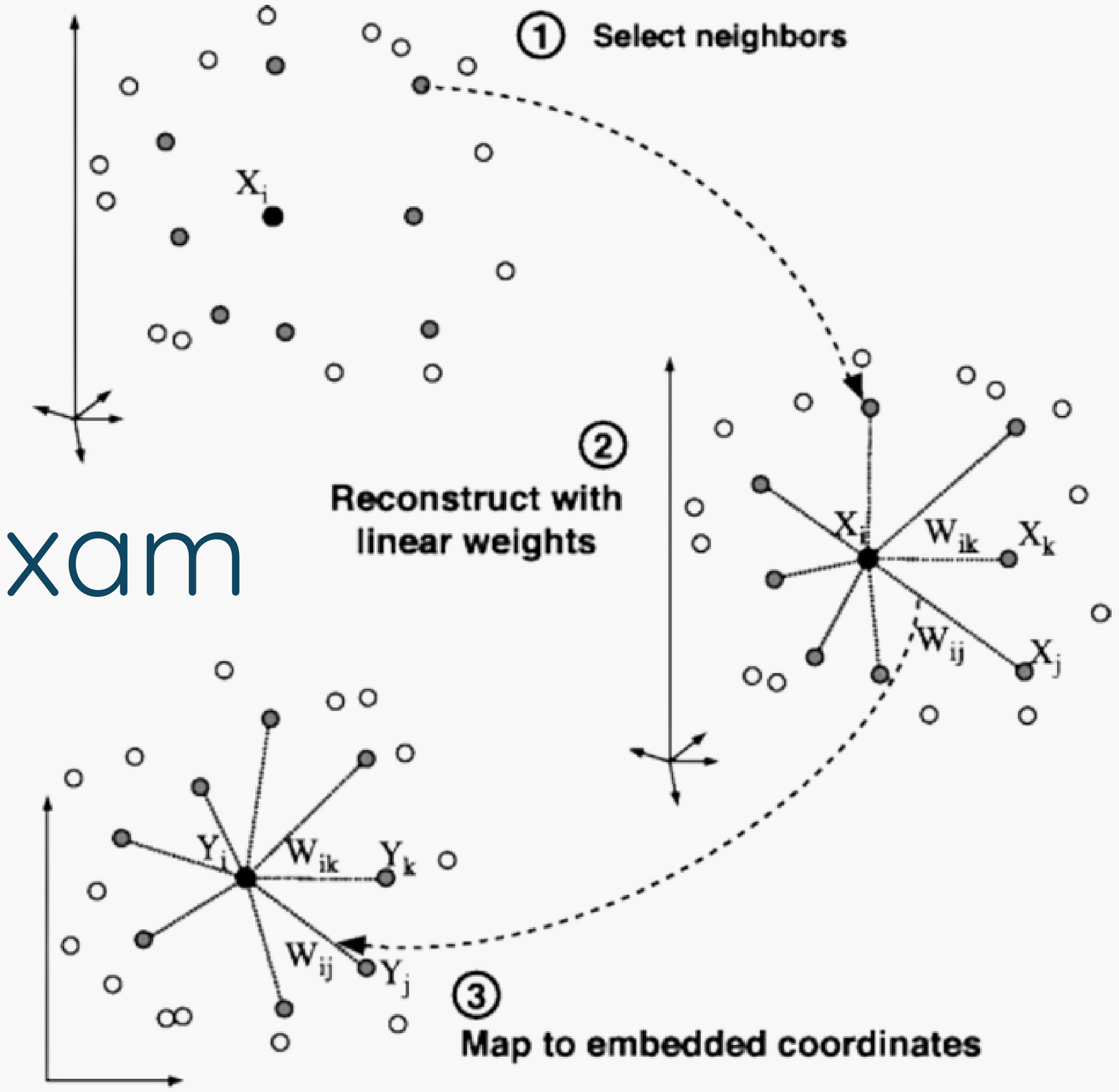
Locally Linear Embedding

LLE



Unsupervised Learning Final Exam

Tanja Derin

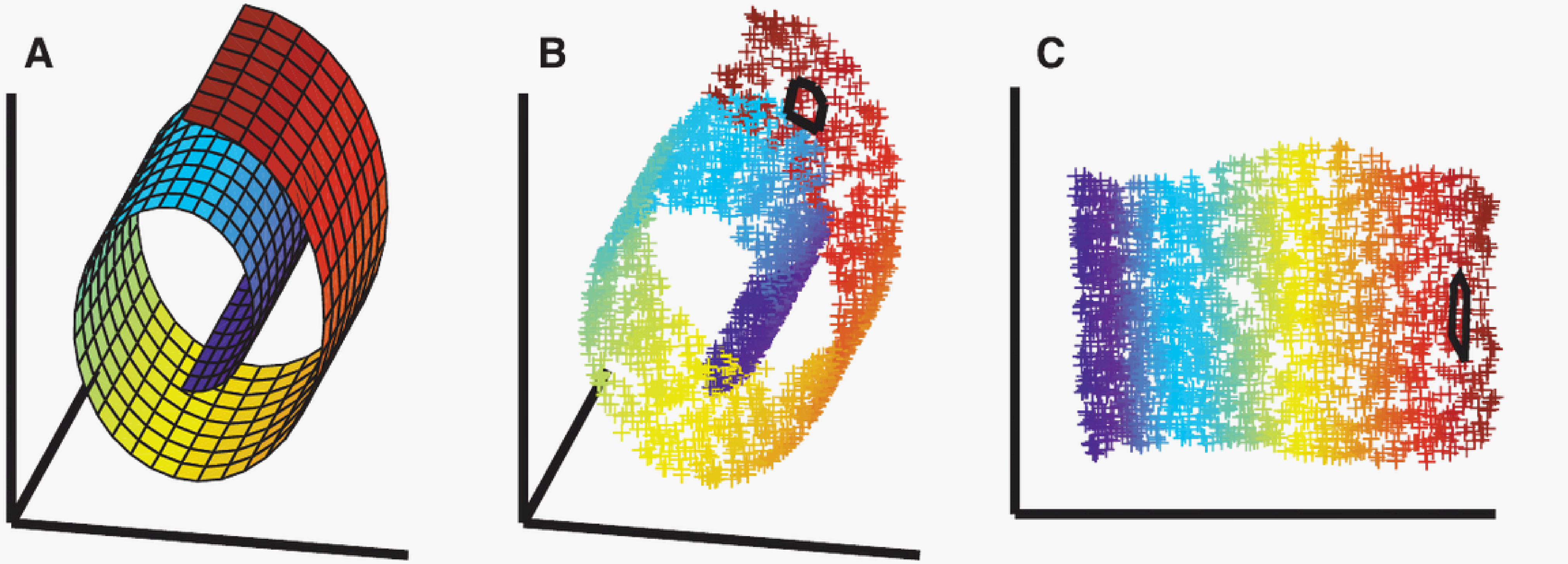
UNIVERSITÀ DEGLI STUDI DI TRIESTE

January, 2025

Introduction

Locally Linear Embedding (LLE)

- A **nonlinear dimensionality reduction** technique tailored for data on a curved surface.
- Preserves **local relationships** between data points.
- Approach:
    - Leverages local symmetries of linear reconstructions.
    - Uncovers the global structure of the manifold **without explicitly defining a distance** metric in the embedded space.
- **Assumption: Each data point and its neighbors lie on or near a locally linear patch of the manifold.**



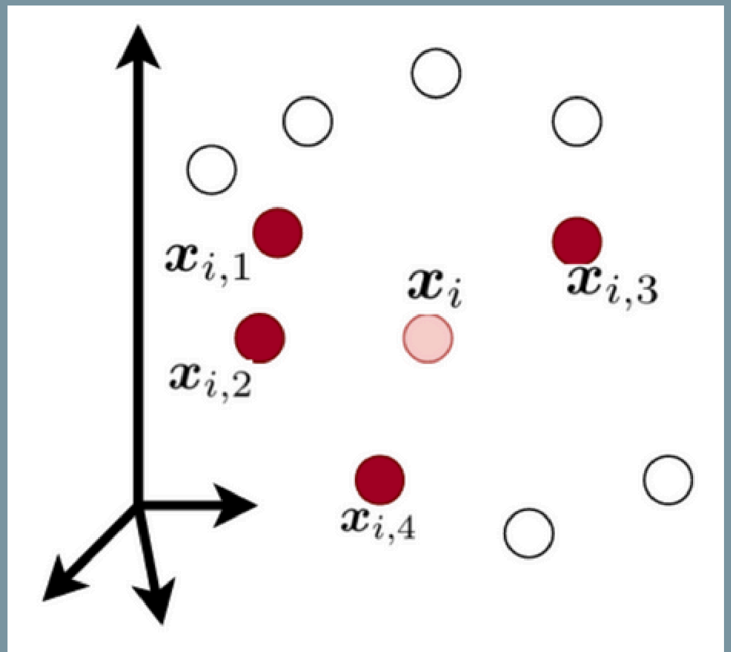
Algorithm

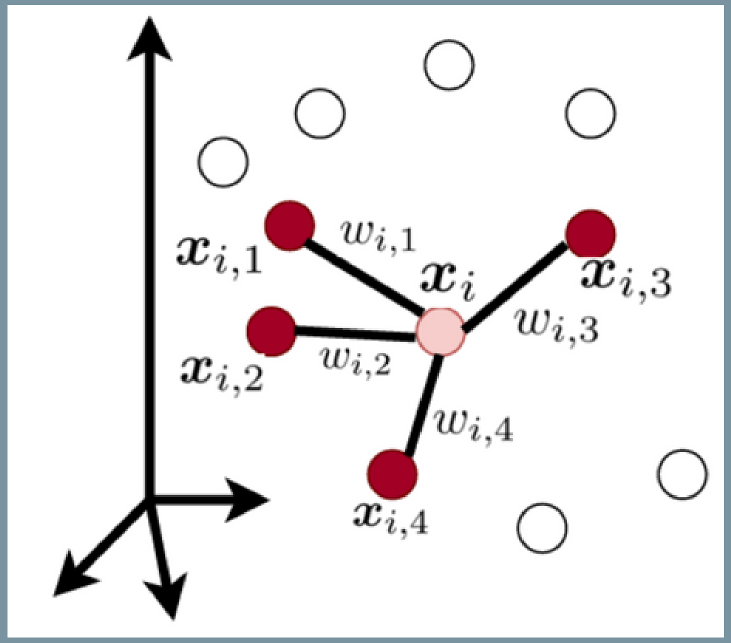
Three Main Steps

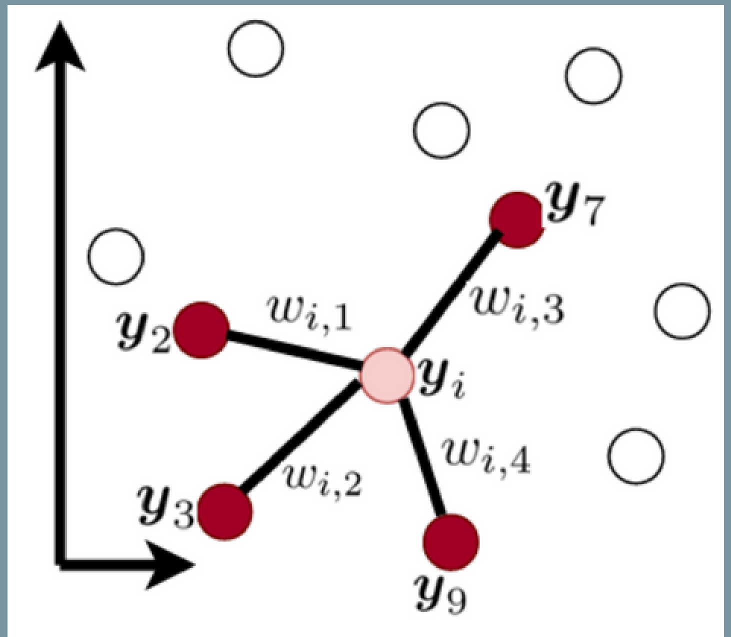
1. **Find the k-Nearest Neighbors**
   - Identify the k-nearest neighbors for each data point in the high-dimensional space.
   - This step captures the **local geometry** of the data.

2. **Compute Reconstruction Weights**
   - Represent each data **point as a linear combination of its k-nearest neighbors**.
   - Minimize reconstruction error while preserving local linearity.

3. **Compute the Low-Dimensional Embedding**
   - Map data points to a low-dimensional space.
   - Ensure the **reconstruction weights are preserved, maintaining the local structure.**





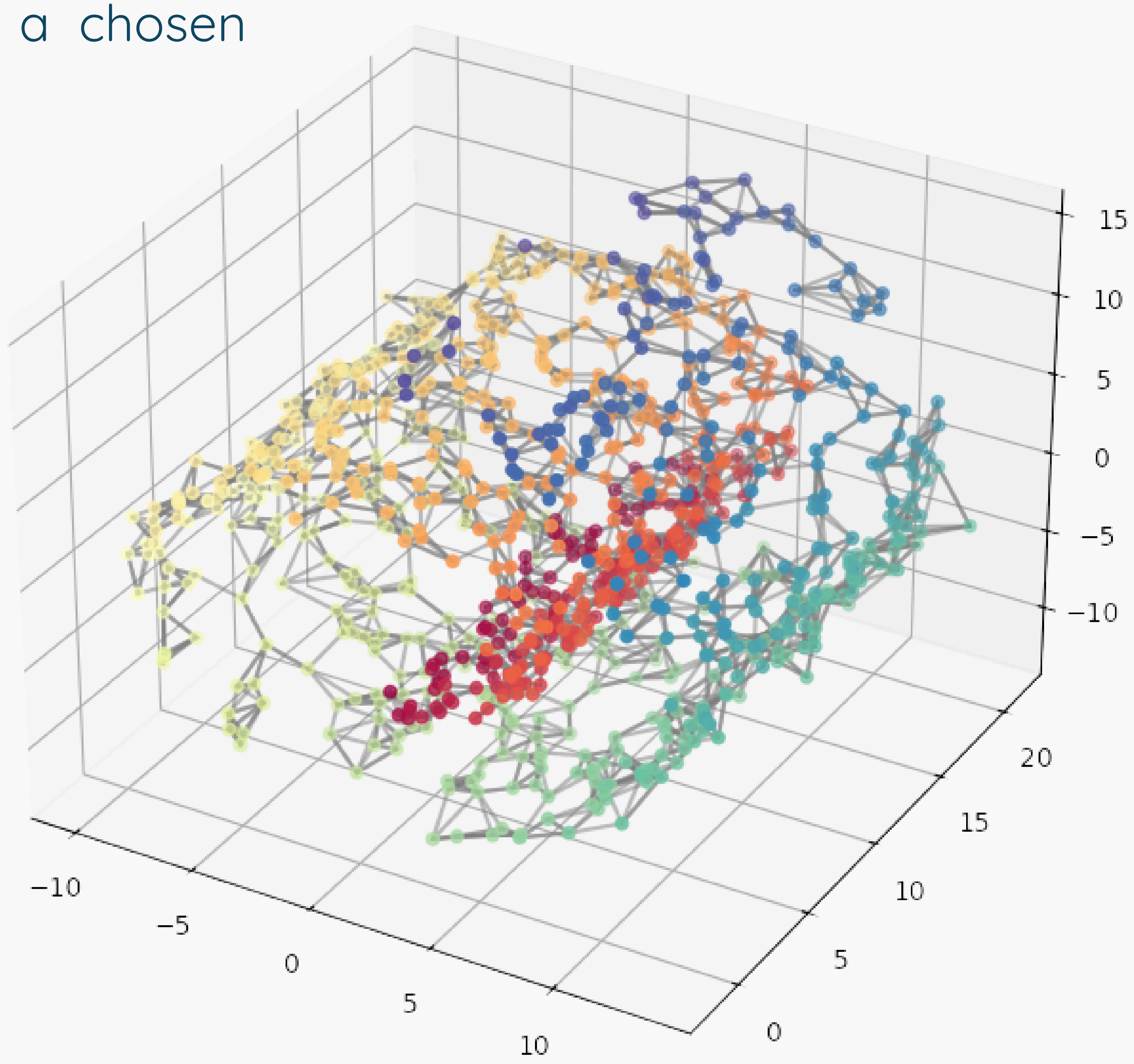


Step 1: Defining Neighborhoods

K-Nearest Neighbors (k-NN) Method

Identifies the k closest data points to xi based on a chosen distance metric, Euclidean distance.

- **validate the neighborhood implementation** using a simple example
- generate a **Swiss roll dataset** and identify the neighbors of each data point using the k-nearest neighbors



Step 2: Find the weights

The weights are crucial as they are used to **preserve the geometric configurations** of the data points by **summing to one** (convex combination), ensuring invariance to rotations, rescalings, and translations while maintaining the local structure of the data.

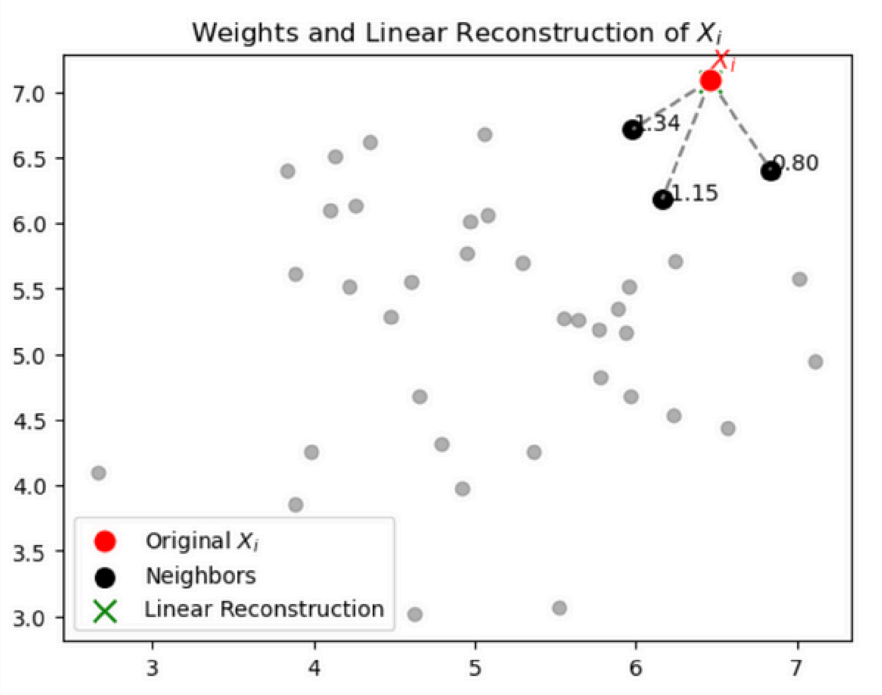
$$\sum_{j=1}^{k} w_{ij} = 1$$

The objective is to express **xi** as a linear combination of its **k** nearest neighbors

$$x_i \approx \sum_{j=1}^{k} w_{ij} x_{ij}$$

Minimize the reconstruction error, defined as the squared Euclidean distance between xi and its reconstruction from the neighbors

$$\epsilon(W) = \sum_{i=1}^{n} \|x_i - X_i w_i\|^2 = \|x_i \mathbf{1} \tilde{w}_i - X_i \tilde{w}_i\|^2 = \|(x_i \mathbf{1}^T - X_i)\tilde{w}_i\|^2 = \tilde{w}_i^T \Big(\overbrace{(x_i \mathbf{1}^T - X_i)^T (x_i \mathbf{1}^T - X_i)}^{\boxed{G_i}}\Big)\tilde{w}_i$$



Gram matrix, capturing the local geometry of the data

$$\text{minimize} \sum_{i=1}^{n} w_i^T G_i w_i \quad \text{subject to } \mathbf{1}^T w_i = 1$$

To solve this, we use a Lagrange multiplier.

Implementation involves solving the linear system avoiding matrix inversion by using numerical solvers

$$G_i w_i = \lambda_i \mathbf{1},$$

Step 3: Embedding

In the final step we map the data points into a **lower-dimensional** space while preserving local neighborhood relations.

The goal is to **minimize the reconstruction error in the embedded space,** defined as:

$$\Phi(Y) = \sum_{i=1}^{n} \|y_i - \sum_{j=1}^{n} w_{ij} y_j\|^2$$

To ensure the embedding is well-defined, the following **constraints** are imposed:

1. **The embedded points must be centered at the origin**
2. **The covariance of the embedded points must be the identity matrix**

- $$\sum_{i=1}^{n} y_i = 0$$
- $$\frac{1}{n} \sum_{i=1}^{n} y_i y_i^T = I$$

$$\Phi(Y) = \|Y^T (I - W)^T\|_F^2$$

$$\Phi(Y) = \mathrm{tr}(Y^T (I - W)^T (I - W) Y)$$

$$M = (I - W)^T (I - W)$$

(I−W) represents the Laplacian matrix of the weight matrix W.

M is the Gram matrix

$$\text{minimize}_Y \quad \mathrm{tr}(Y^T M Y)$$

$$\frac{1}{n} Y^T Y = I$$

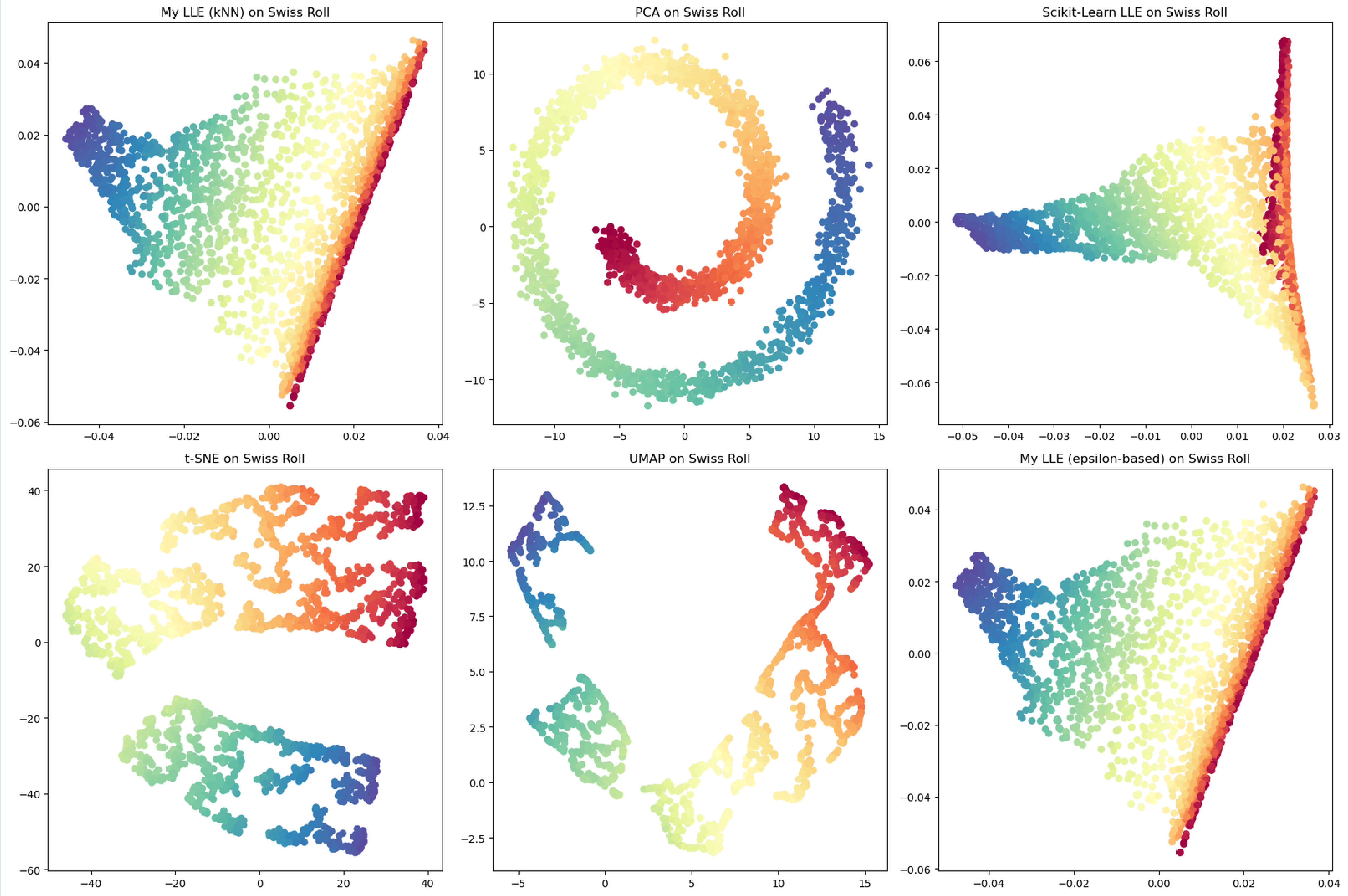
$$Y^T \mathbf{1} = 0$$

The matrix M is symmetric, making it suitable for eigen decomposition.

- Compute the **eigenvalues and eigenvectors of M.**
- The new embedded space is defined by the **eigenvectors corresponding to the smallest non-zero eigenvalues** of M.

SWISS ROLL

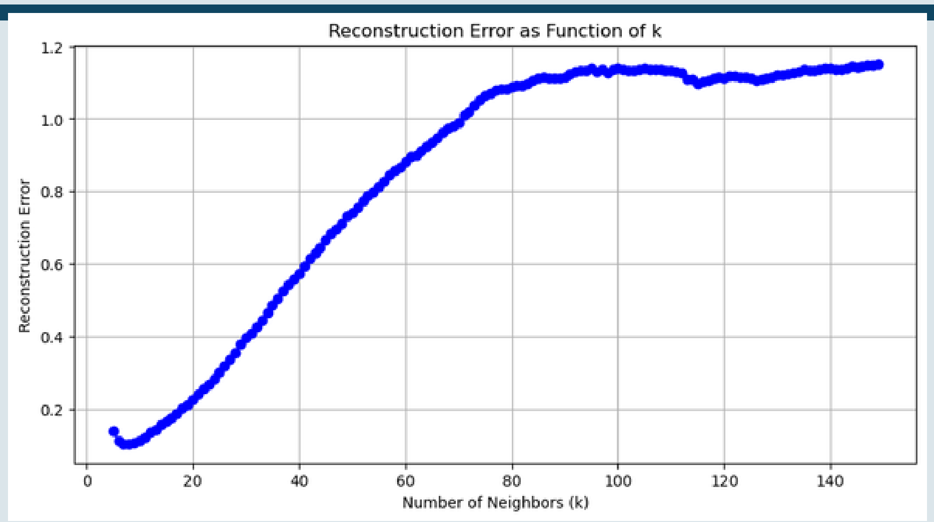
- LLE **unravels** and visualizes the complex **2D manifold** within the 3D Swiss Roll.
- Custom **LLE implementations align closely** with Scikit-Learn's
- **t-SNE and UMAP**-successfully unroll the Swiss Roll but result in **less smooth surfaces.**
- PCA- **Fails to unroll** the Swiss Roll due to its linear nature,



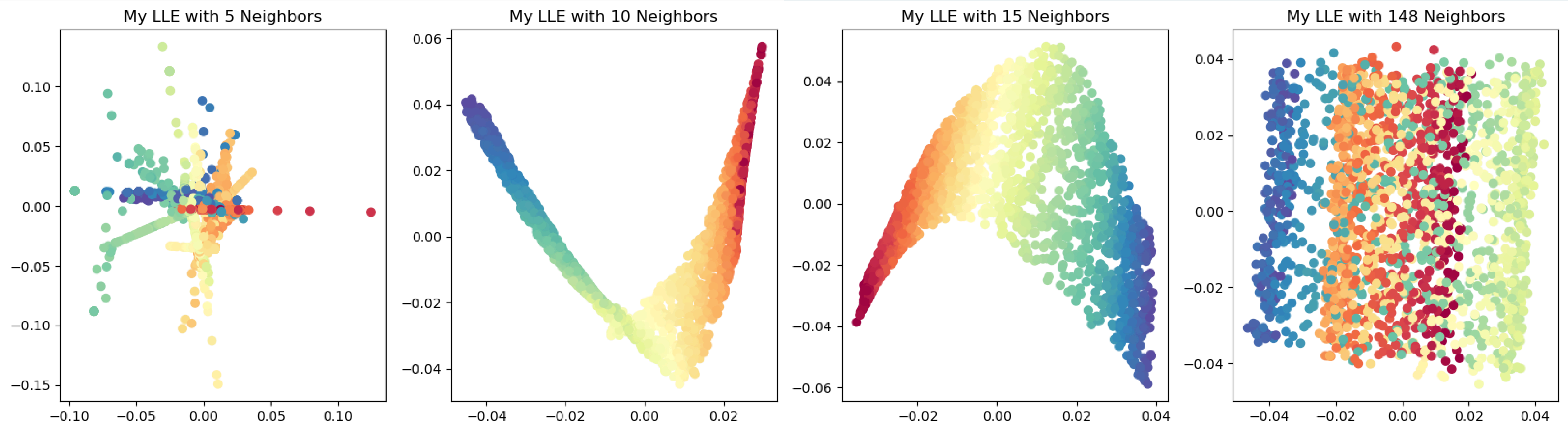
Parameter

The LLE algorithm relies on the Number of Neighbors. It determines how many neighbors each data point considers for reconstructing itself.

- **A smaller (k)** focuses on very local structures, which can lead to overfitting or **disconnected embeddings.**
- **A larger (k)** captures more global structures, but it may **blur fine details** and lead to higher reconstruction errors.

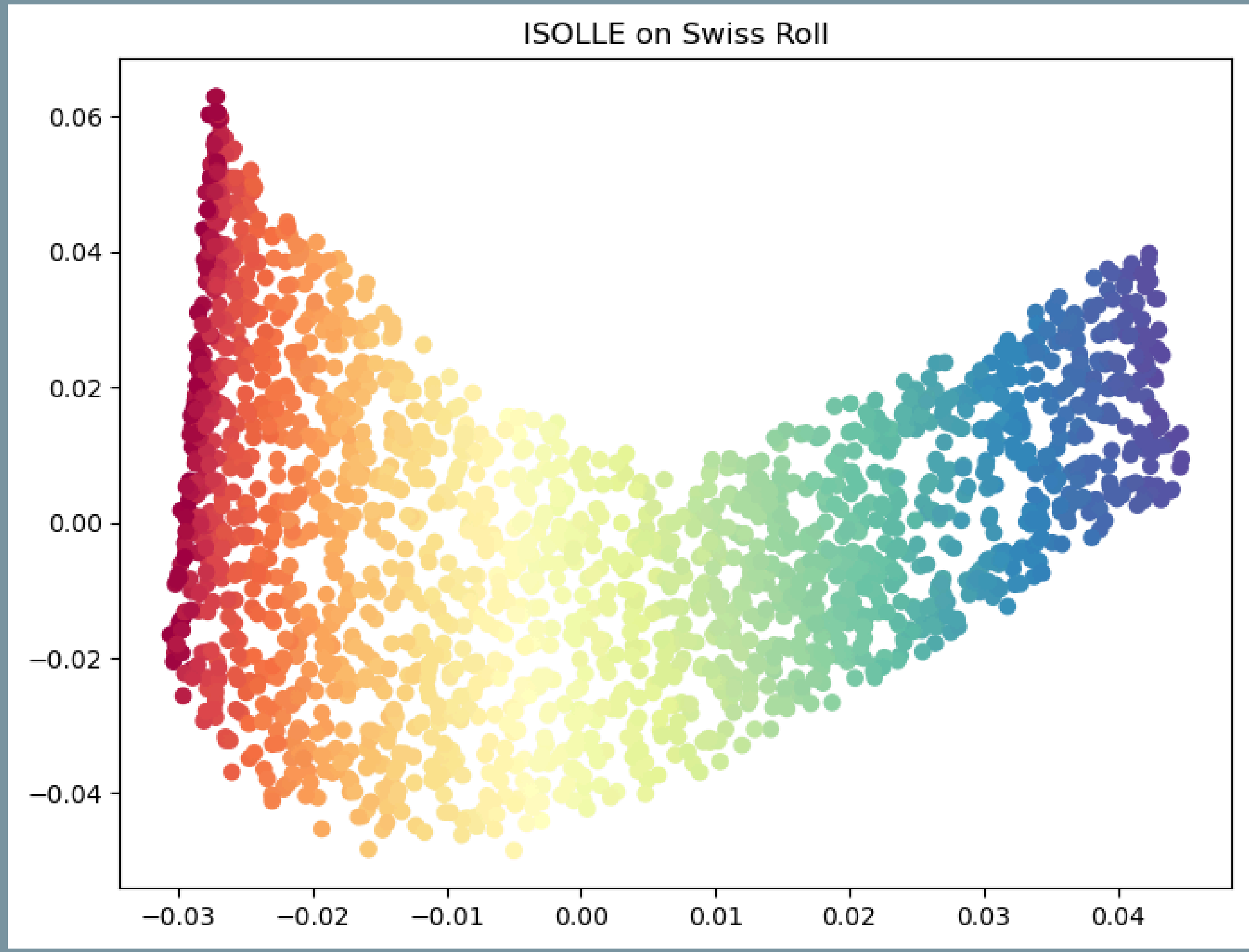


The optimal number of neighbors on the Swiss roll dataset appears to be between 10 and 15. This range shows a balance between sufficient detail and smooth transitions

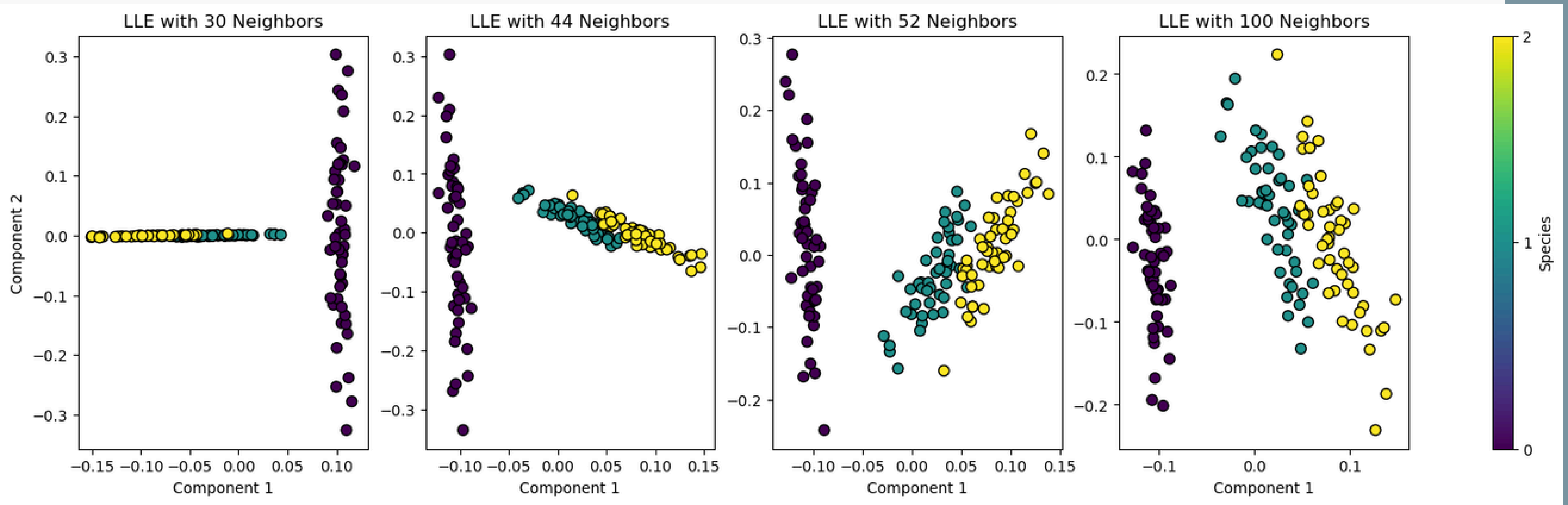


Fuse LLE with Isomap as ISOLLE

Fusing LLE with Isomap involves using **geodesic distances** rather than simple Euclidean distances to construct the k-NN graph.



The Iris dataset, which consists of 4 features and 3 species, is used to demonstrate how ISOLLE works with different neighbor settings.



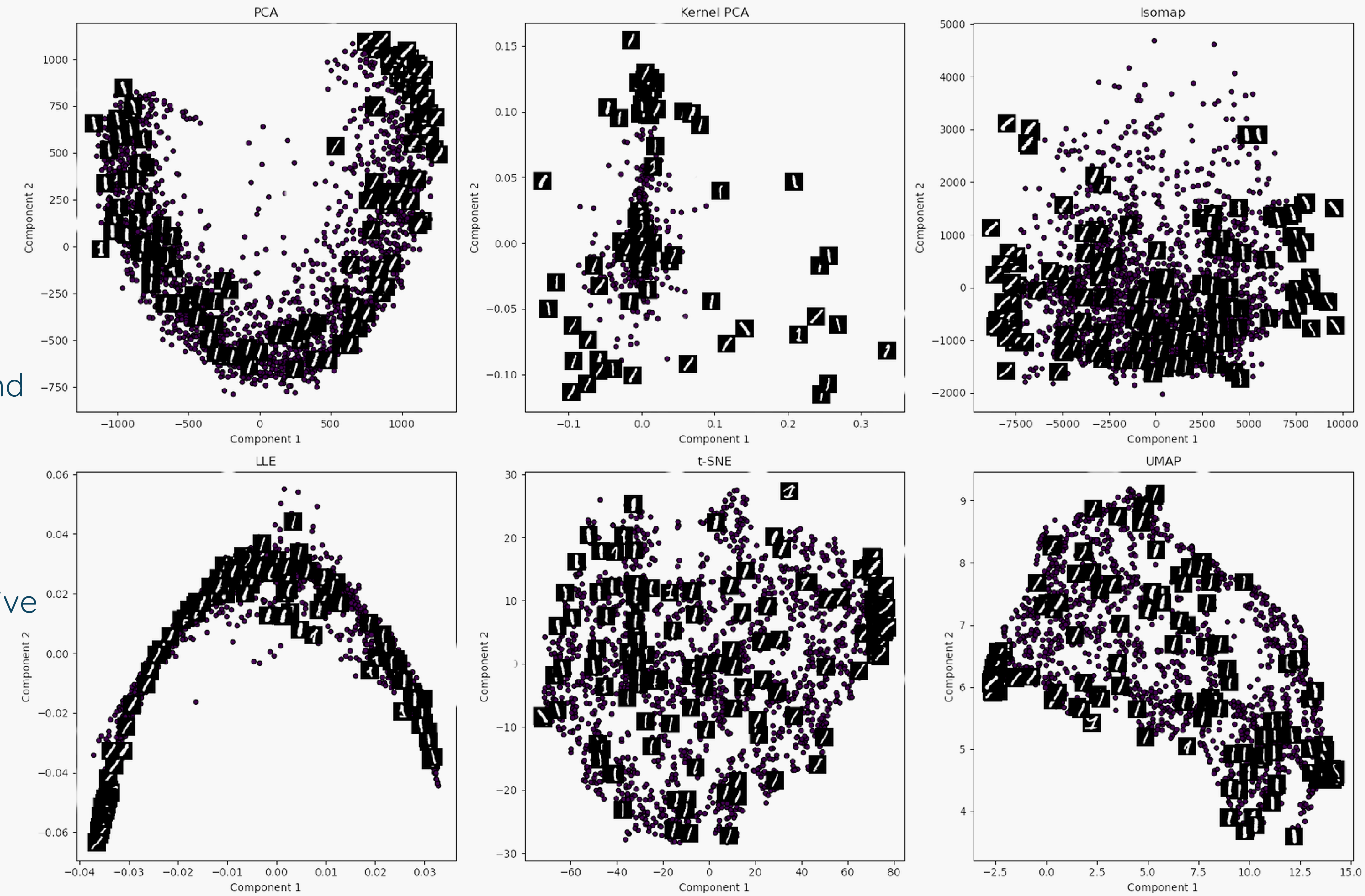
The embedding for small neighbor sizes (30) fails to preserve the overall structure, Larger neighbor sizes (100) oversmooth the local details, potentially reducing the interpretability**. Using 52 neighbors appears to provide the best trade-off** for the Iris dataset.

Image Processing

LLE to the **MNIST dataset**

(focusing on digit '1')

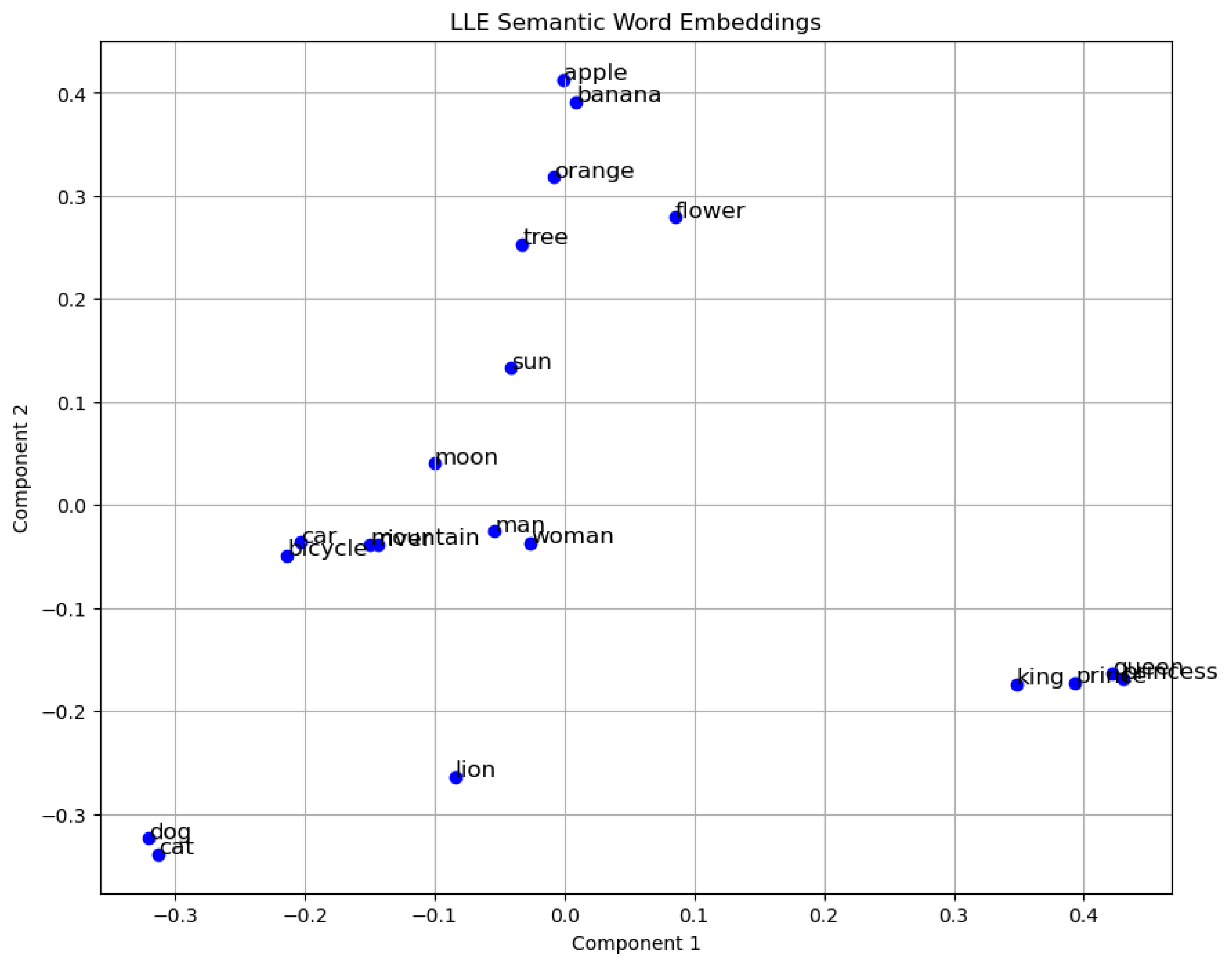
- LLE clusters **similar shapes** and styles of the digit '1', revealing variations in handwriting.
- LLE provides the most distinctive and **structured visualization**



Semantic Word Embedding

Visualize relationships between high-dimensional **word vectors (from Google News Word2Vec) in a 2D space.** Procedure:

1. **Load Word Vectors:** Use pre-trained word embeddings.
2. **Select Words:** Choose a subset of interest
3. **Apply LLE:** Reduce dimensionality to 2D while preserving local relationships.



Plot of the reduced embeddings reveals **semantic clusters** (e.g., 'king', 'queen', 'prince'; 'cat', 'dog').

LLE maintains semantic relationships, showing distinct clusters and uncovering the **underlying structure of word embeddings.**

Conclusion

Advantages

- Preserves **Local Structures:** Retains geometrical properties of nonlinear manifolds.
- Handles **Non-Linearity**: Suitable for curved or twisted distributions.
- **Facilitates Visualization:** Reduces dimensions while maintaining core characteristics.

Disvantages

- **Curse of Dimensionality**: Struggles with extremely high-dimensional spaces.
- **Resource-Intensive**: High memory and computational demands for adjacency matrix and eigenvalue decomposition.
- **Sensitive to Noise and Outliers**: Disrupts local linearity assumptions, reducing embedding accuracy.

In practice, LLE shines in controlled datasets with moderate size and low noise levels, where uncovering local manifold structures is the priority. Its utility can be **extended through hybrids like `Hessian LLE` or `modified LLE`**, which address some of its limitations.

Thank you