

# Proyecto 7: Laplacian Eigenmaps para reducción de dimensión y embeddings

Asignatura: Métodos Diferenciales para la IA

Sergio Herreros Pérez, Mario Kroll Merino, Carlos Mazuecos Reíllo,  
José Andrés Ridruejo Tuñón

# Why Laplacian Eigenmaps?

- Classical dimensionality reduction techniques (PCA, MDS) are **linear** and fail to capture non-linear structures.
- Real datasets often lie near a **non-linear manifold** embedded in high dimensions.
- **Laplacian Eigenmaps** aim to preserve the **local geometry** of the data.

## Key Idea

Build a graph that captures local relationships and use the eigenvectors of the **graph Laplacian** to obtain a smooth low-dimensional embedding.

# Laplacian Eigenmaps: Core Idea

- 1 Build a **neighborhood graph** using k-NN or a kernel.
- 2 Assign similarity weights  $w_{ij}$  between connected points.
- 3 Construct the graph Laplacian:  $L = D - A$ .
- 4 Solve the spectral problem:

$$Lf = \lambda f.$$

## Final Embedding

The eigenvectors associated with the smallest non-zero eigenvalues provide the embedding coordinates.

# Geometric Interpretation

- The graph Laplacian measures the **smoothness** of a function on the graph.
- Minimizing  $f^\top Lf = \sum_{i,j} w_{ij}(f_i - f_j)^2$  encourages nearby nodes to have similar values.
- The smallest-energy eigenvectors correspond to the "smoothest" functions on the graph.

## Why this reduces dimension

These smooth eigenfunctions capture the local geometric structure of the underlying manifold.

# The Continuous Problem: Laplace–Beltrami

On a continuous domain or manifold:

$$\min_{u: \|u\|=1} \int_{\Omega} |\nabla u|^2 dx$$

leads to the eigenvalue problem:

$$-\Delta u = \lambda u.$$

- The eigenvalues encode the **geometry** of the manifold.
- The eigenfunctions are smooth modes with minimal oscillation.

# From Continuous to Discrete: The Key Connection

## Continuous Laplacian

$$-\Delta u = \lambda u$$

$$\int |\nabla u|^2$$

## Graph Laplacian

$$Lf = \lambda f$$

$$f^\top Lf = \sum w_{ij}(f_i - f_j)^2$$

## Conceptual Correspondence

- $|\nabla u|^2 \longleftrightarrow (f_i - f_j)^2$
- Laplace–Beltrami operator  $\longleftrightarrow$  Graph Laplacian
- Smoothness on the manifold  $\longleftrightarrow$  Smoothness on the graph

# Laplacian Eigenmaps in Graph-Based Learning

- Laplacian Eigenmaps provide the spectral foundation for many graph learning algorithms:
  - **Spectral Clustering**: uses the smallest non-zero Laplacian eigenvectors to partition graphs.
  - **Graph Convolutional Networks (GCNs)**: graph convolutions are defined using the Laplacian spectrum.
- The Laplacian encodes the local geometry of the dataset by capturing similarity relations through weights  $w_{ij}$ .
- Eigenvectors provide smooth representations that respect this geometry.

# Connection to the Finite Element Method (FEM)

## Shared Variational Principle

Both FEM and Laplacian Eigenmaps are based on minimizing an energy associated with the Laplacian:

$$\int_{\Omega} |\nabla u|^2 \quad \longleftrightarrow \quad \sum_{i,j} w_{ij} (f_i - f_j)^2 = f^{\top} L f.$$

- FEM approximates  $\Delta$  on a mesh; Laplacian Eigenmaps approximate  $\Delta$  on a **similarity graph**.
- In both cases, low-energy eigenfunctions correspond to the smoothest modes.
- This provides a rigorous bridge between PDE-based models and graph learning.

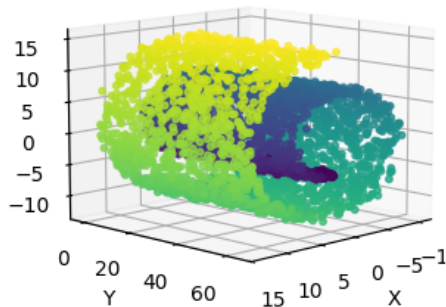


# From Continuous Geometry to Discrete Data

- Laplacian Eigenmaps allow transferring ideas from differential geometry to data analysis:
  - **Diffusion**: heat flow on a manifold  $\leftrightarrow$  diffusion processes on graphs.
  - **Smoothness**: low-oscillation eigenfunctions  $\leftrightarrow$  low-variation graph signals.
  - **Eigenmodes**: Laplace–Beltrami eigenfunctions  $\leftrightarrow$  graph Laplacian eigenvectors.
- This creates a unified framework connecting:
  - PDEs and variational principles,
  - manifold geometry,
  - and machine learning on graphs.
- The result: geometric structure of data becomes accessible even in discrete, high-dimensional settings.

# Swiss Roll Dataset

Swiss Roll - View 1



Swiss Roll - View 2

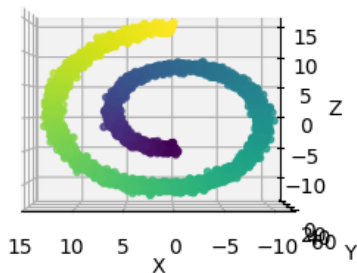


Figure: 3D Visualization of the Swiss Roll Dataset

# Laplacian Eigenmaps Embedding of Swiss Roll

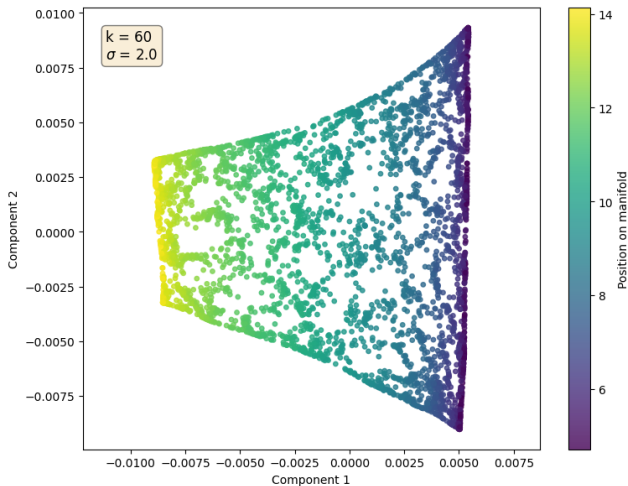


Figure: 2D Embedding of the Swiss Roll Dataset using Laplacian Eigenmaps

# Effect of Neighbors ( $k$ ) on Embedding

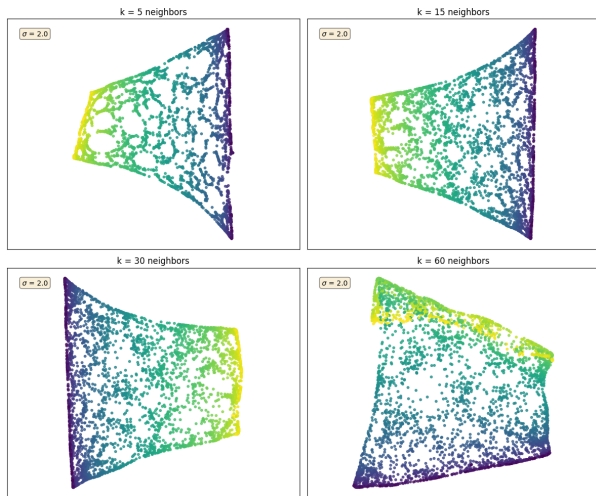


Figure: Effect of Varying Number of Neighbors ( $k$ ) on the Embedding

# Effect of Sigma (Kernel Bandwidth) on Embedding

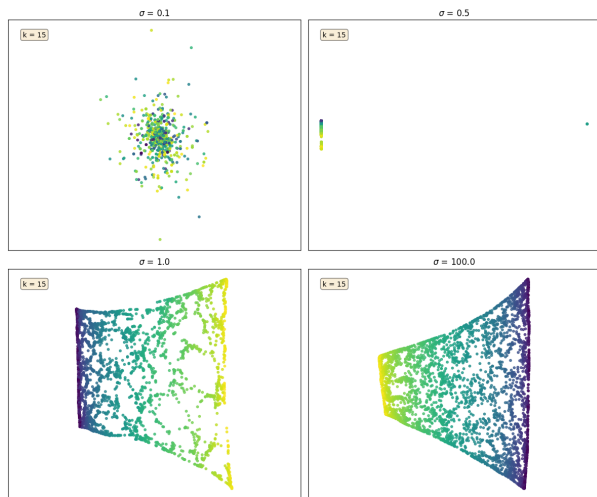


Figure: Effect of Varying Sigma (Kernel Bandwidth) on the Embedding

# PCA vs Laplacian Eigenmaps on Swiss Roll

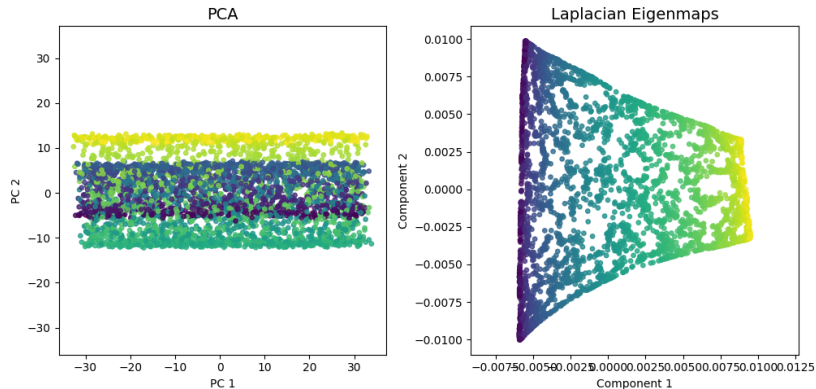


Figure: Comparison of PCA and Laplacian Eigenmaps on the Swiss Roll Dataset

# Mammoth Dataset

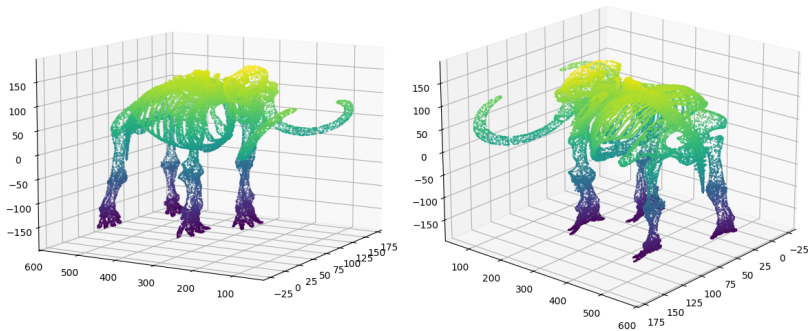
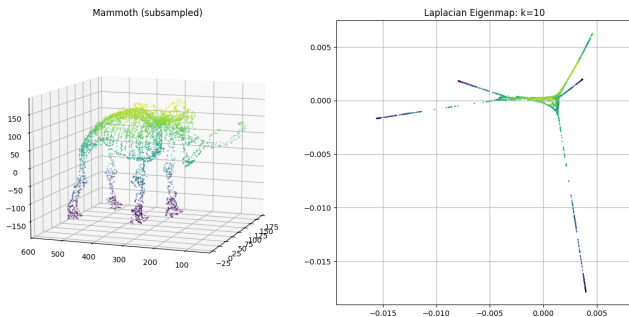


Figure: 3D Mammoth Dataset Visualization

# Laplacian Eigenmaps Embedding of Mammoth Dataset



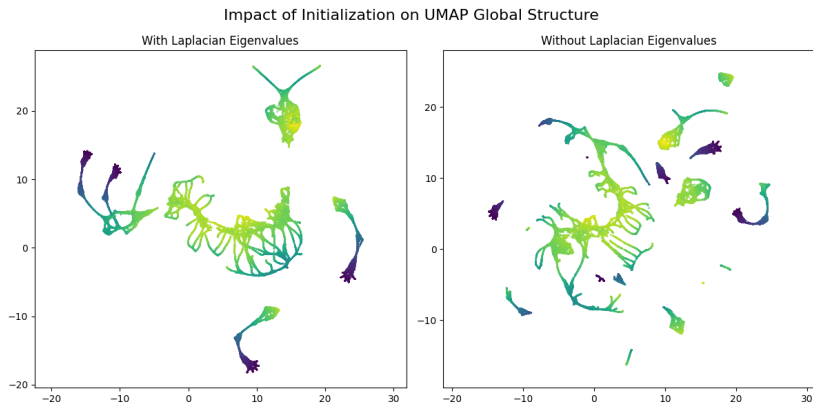
**Figure:** Laplacian Eigenmaps Embedding of the Mammoth Dataset

## Observation

The effect shown on the picture is called the starfish effect. It reflects perfectly how the Laplacian Eigenmaps preserves connectivity, but it fails to preserve global structure.



# Laplacian Eigenmaps as UMAP initialization



**Figure:** UMAP Embedding of the Mammoth Dataset with Laplacian Eigenmaps Initialization and Random Initialization