Dimensionality Reduction

Machine Learning and Deep Learning
Lesson #2

AImage Lab
University of Modena and Reggio Emilia

Multimedia DBs

- Many multimedia applications require efficient indexing in high-dimensions (time-series, images and videos, etc)

- Answering similarity queries in high-dimensions is a difficult problem due to "curse of dimensionality"

  A solution is to use **Dimensionality reduction**

- The main idea: reduce the dimensionality of the space.

- Project the d-dimensional points in a k-dimensional space so that:
  - k << d
  - distances are preserved as well as possible

- Solve the problem in low dimensions

Multi-Dimensional Scaling (MDS)

- Map the items in a k-dimensional space trying to minimize the **stress**

$$stress = \sqrt{\frac{\sum_{i,j}(\hat{d}_{i,j} - d_{i,j})^2}{\sum_{i,j} d_{i,j}^2}} \, with$$

$$d_{i,j} = | o_j - o_i |$$

$$\hat{d}_{i,j} = | \hat{o}_j - \hat{o}_i |$$

- **Steepest Descent algorithm:**
  - Start with an assignment
  - Minimize stress by moving points

- But the running time is $O(N^2)$ and O(N) to add a new item

Embeddings

- Given a metric distance matrix D, embed the objects in a k-dimensional vector space using a mapping F such that
  - D(i,j) is close to D'(F(i),F(j))

- Two types of mapping according to distances (Embedding):
  - **Isometric mapping:**
    - D'(F(i),F(j)) = D(i,j)
  - **Contractive mapping:**
    - D'(F(i),F(j)) <= D(i,j)

Where d' is some Lp measure

- **Two types of embeddings according to warping technique**
  - *Linear* -> data points are projected by a liear transformation (PCA)
  - *Non linear* -> data points are projected non linearly (Laplacian ISOMAP, T-sne)

PCA Algorithm

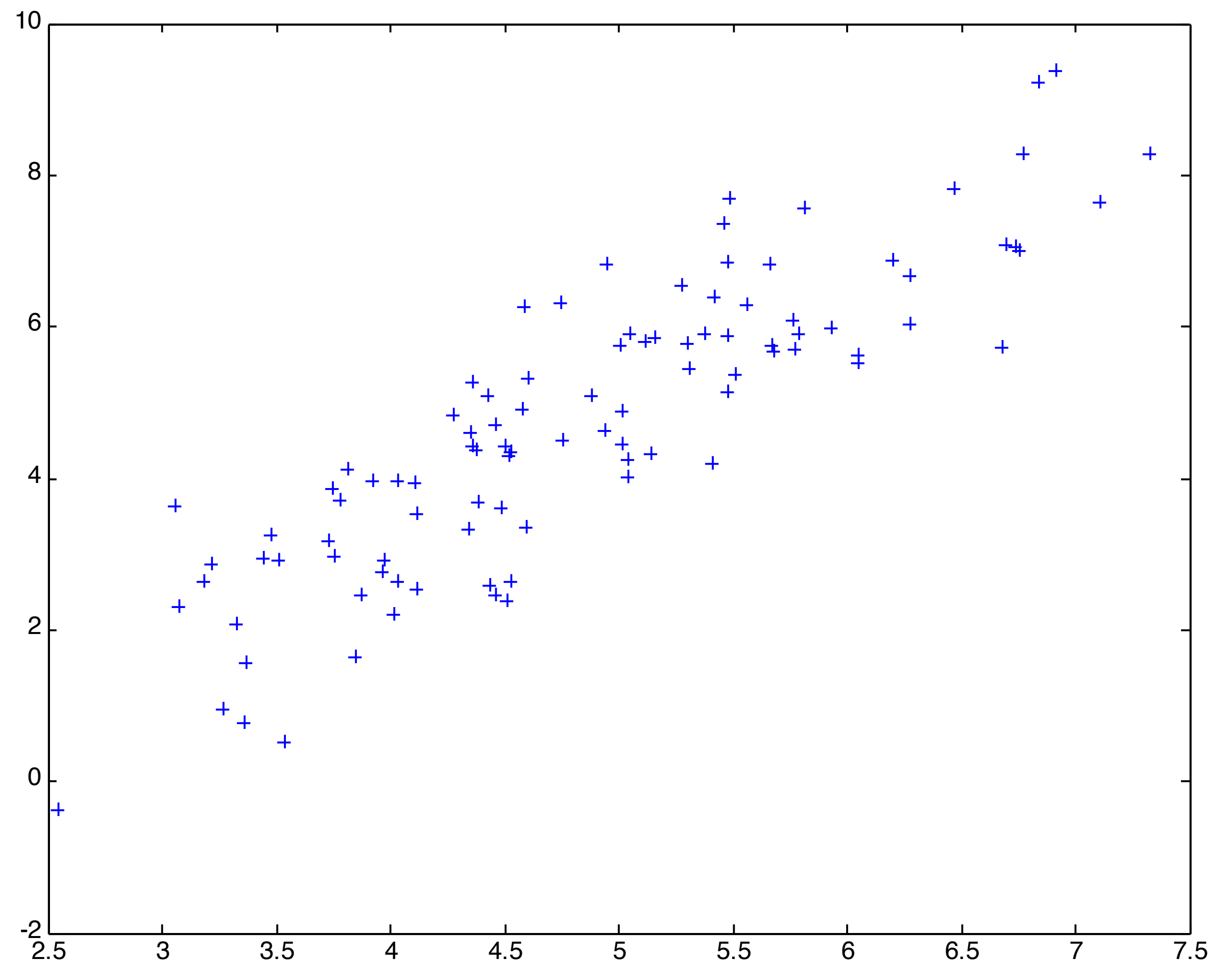
- PCA algorithm:
  - 1. X ← Create N x d data matrix, with one row vector x_n per data point
  - 2. X subtract mean *x* from each row vector x_n in X
  - 3. Σ ← covariance matrix of X
  - Find eigenvectors and eigenvalues of Σ
  - PC's ← the M eigenvectors with largest eigenvalues

Geometric Rationale of PCA

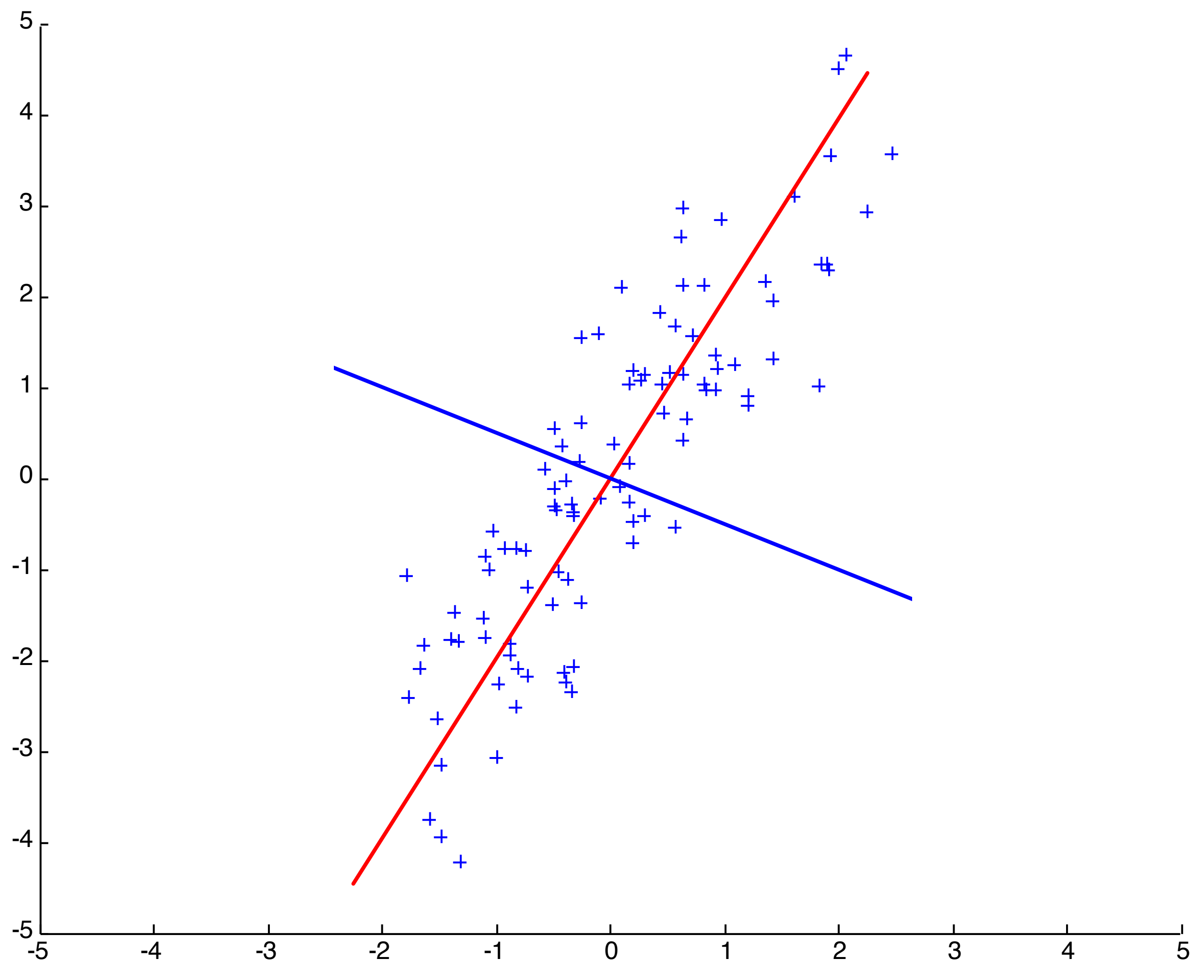
- objective of PCA is to rigidly rotate the axes of this p-dimensional space to new positions (principal axes) that have the following properties:

- ordered such that principal axis 1 has the highest variance, axis 2 has the next highest variance, ... , and axis p has the lowest variance

- covariance among each pair of the principal axes is zero (the principal axes are uncorrelated).

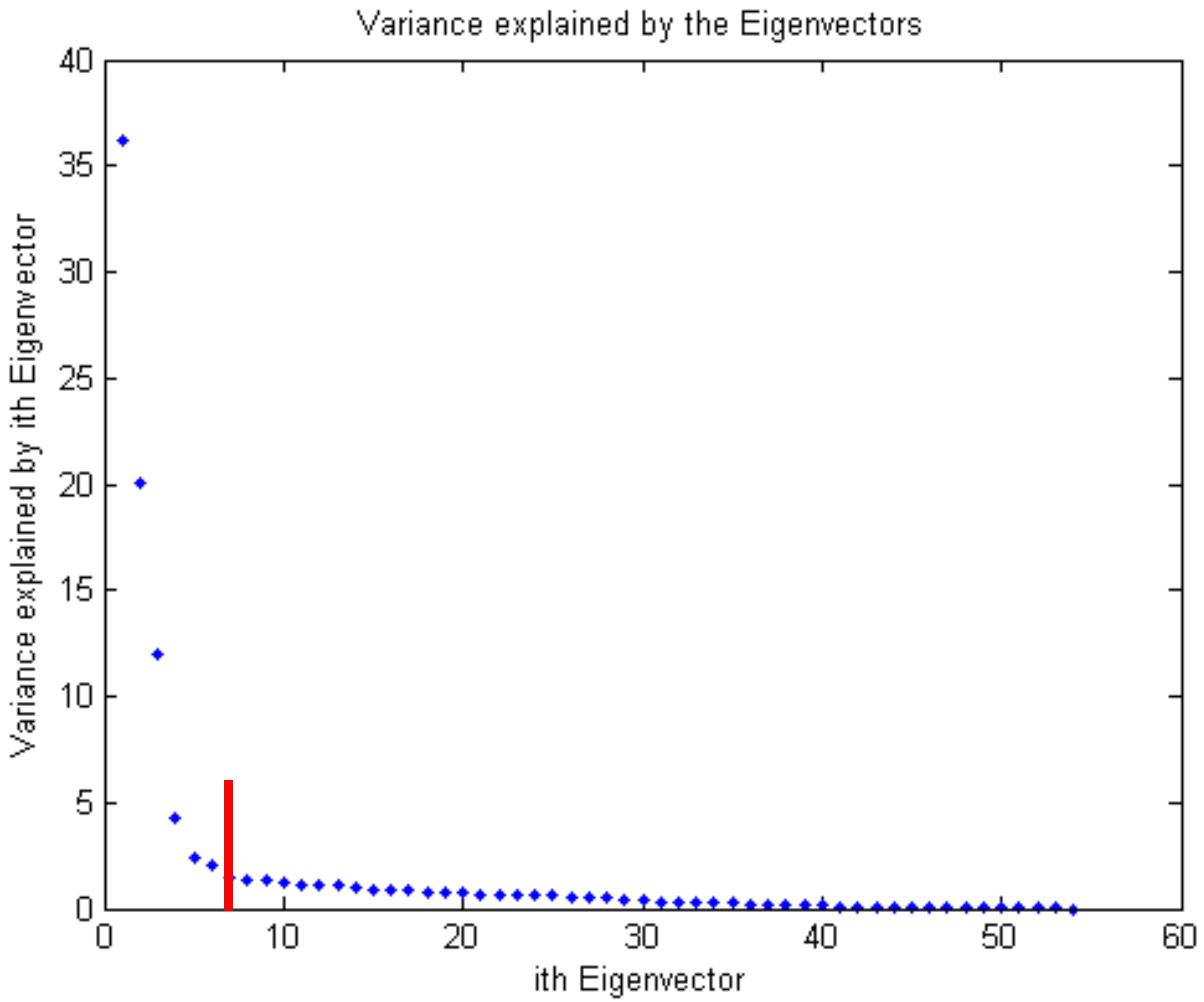


PCA principal AXIS

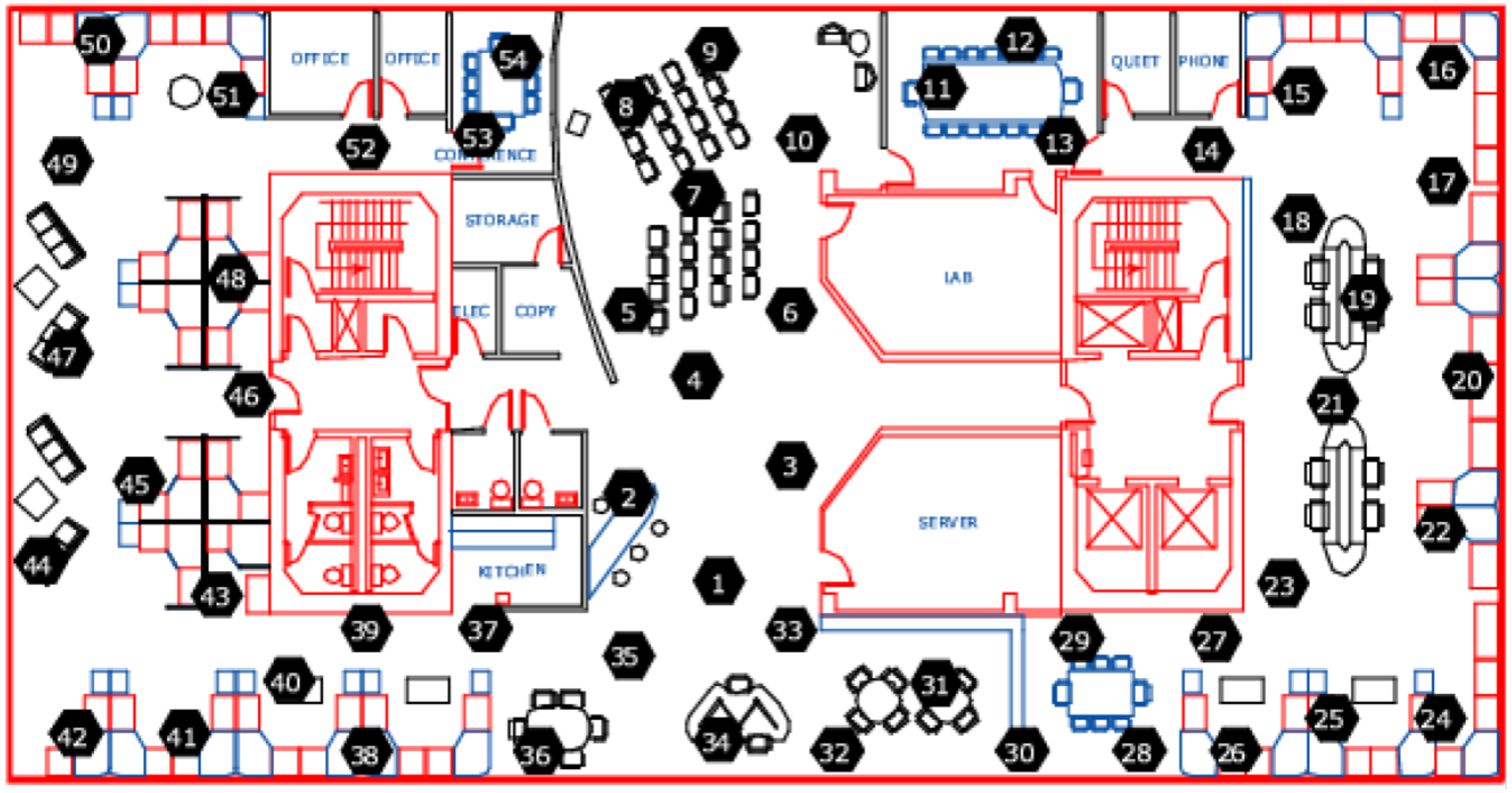


How many components?

- Check the distribution of eigen-values
- Take enough eigenvectors to cover 80-90% of the variance

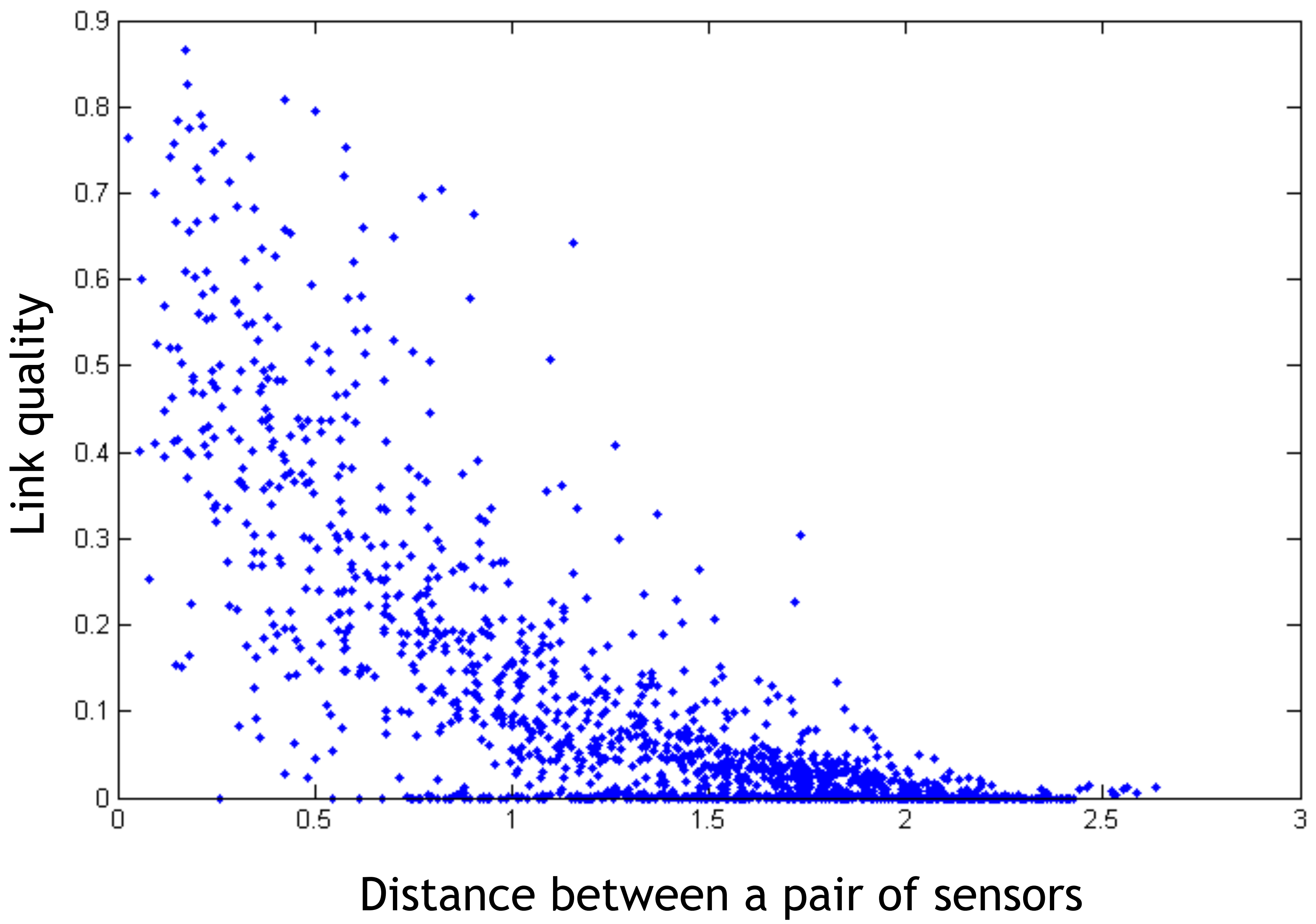


Example Sensor networks



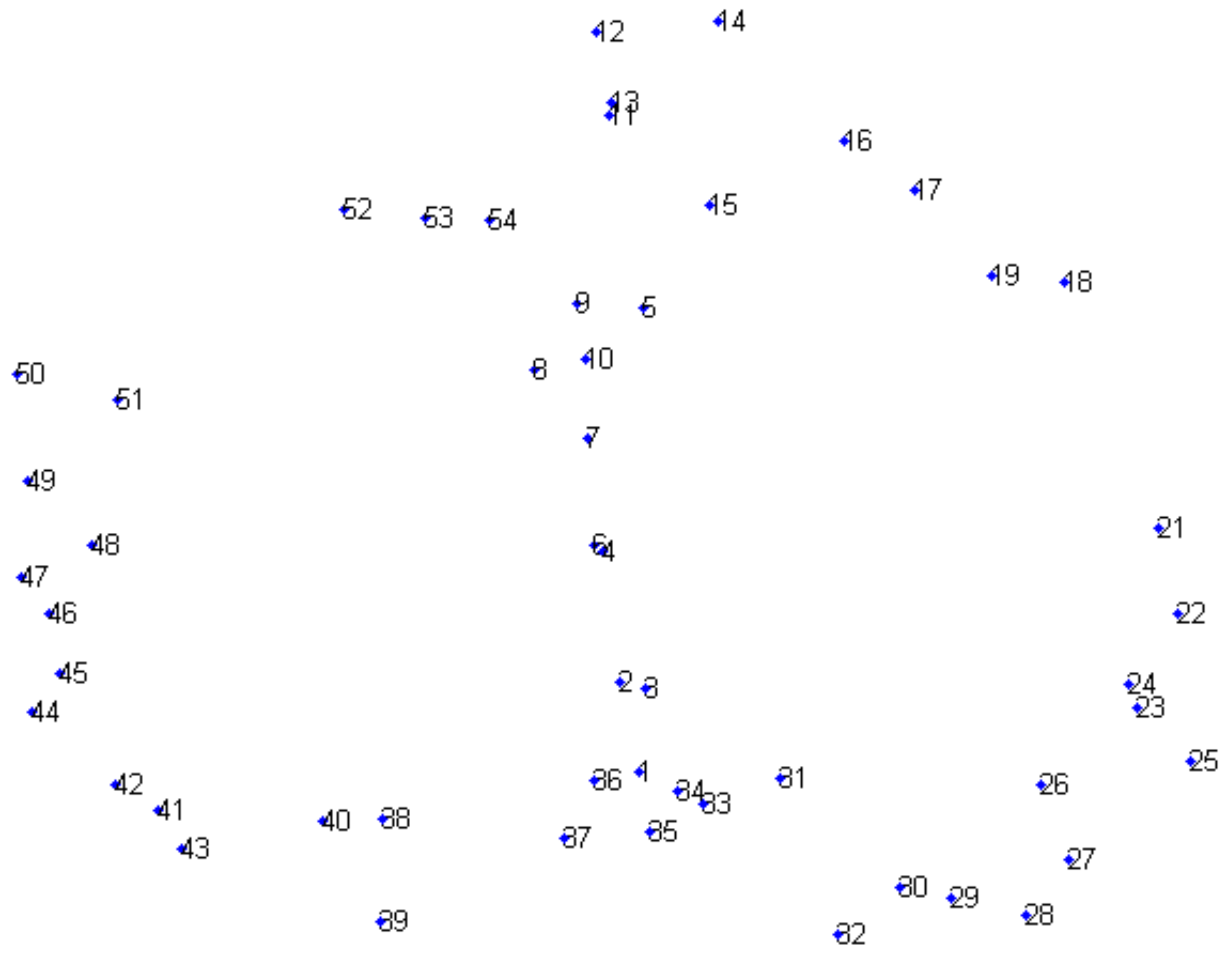
Sensors in Intel Berkeley Lab

Pairwise link quality vs. distance



PCA in action

- Given a 54x54 matrix of pairwise link qualities
- Do PCA
- Project down to 2 principal dimensions
- PCA discovered the map of the lab



Problems and limitations of PCA

- What if very large dimensional data?
  - e.g., Images ($d \geq 10^4$)
- Problem:
  - Covariance matrix Σ is size (d^2)
  - $d=10_4 \rightarrow |\Sigma| = 10^8$

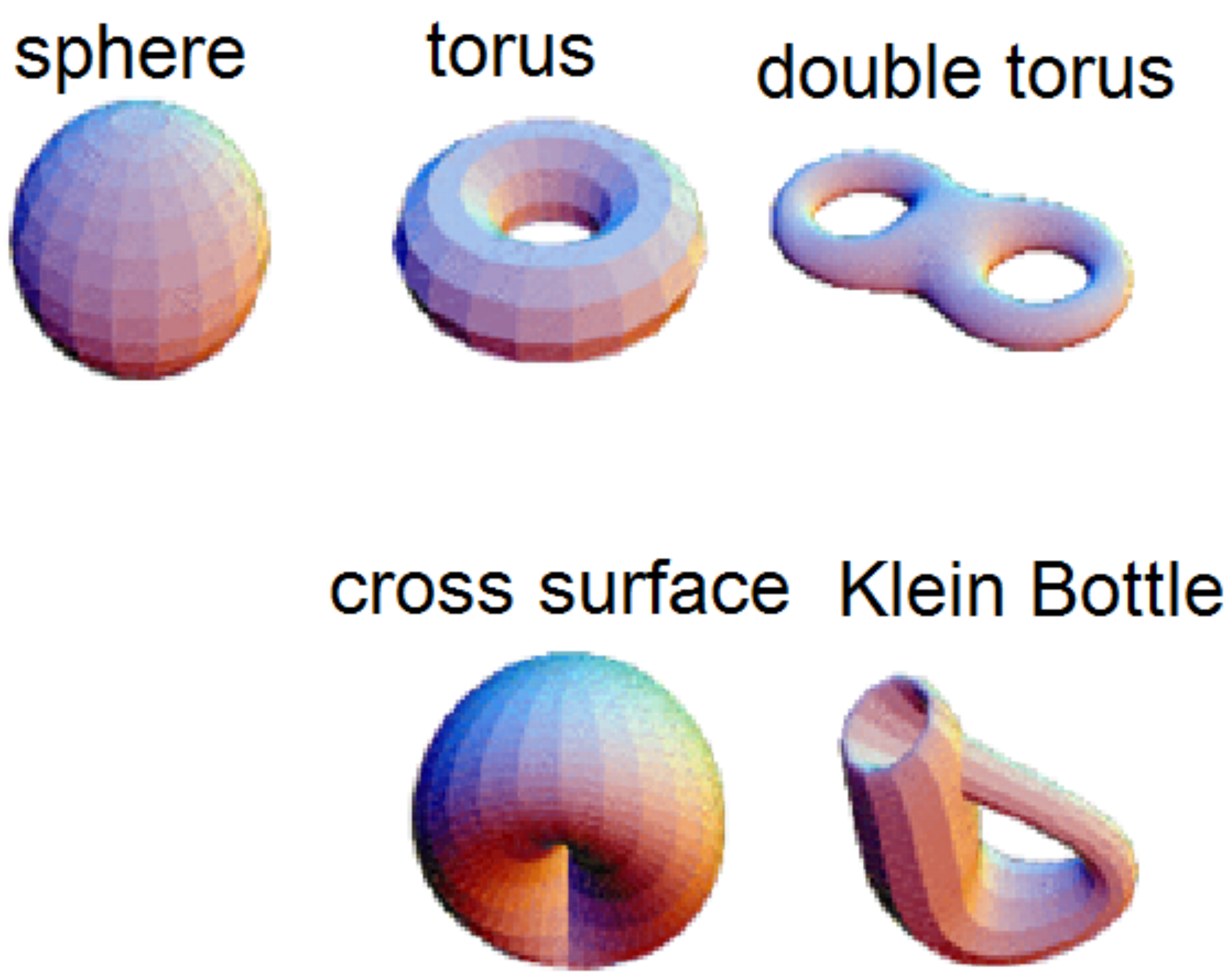
- Singular Value Decomposition (SVD)!
  - efficient algorithms available
  - some implementations find just top N eigenvectors

Laplacian Eigenmaps

Manifold

- A manifold is a **topological space** which is **locally Euclidean.** In general, any object which is nearly "flat" on small scales is a manifold.

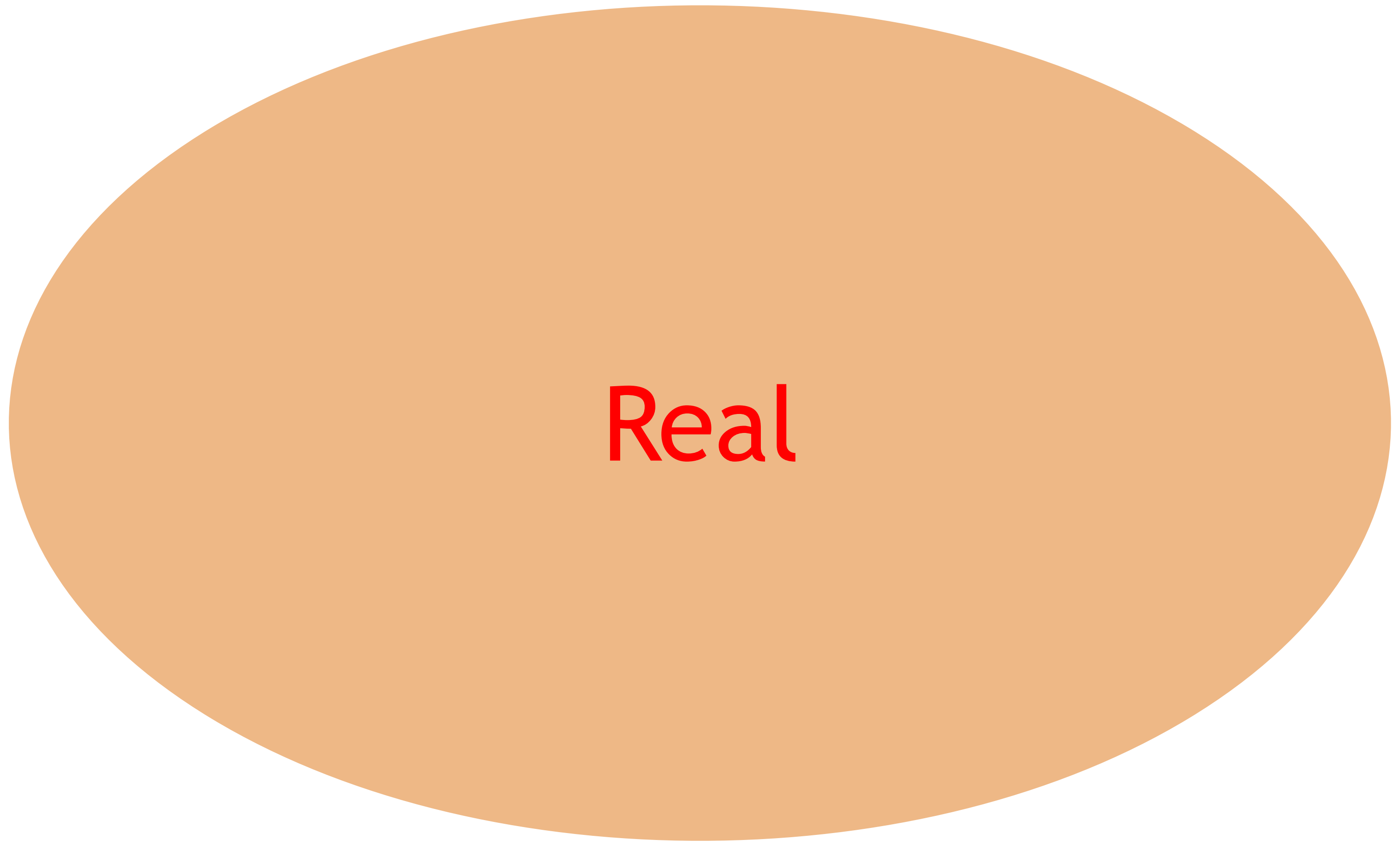
Examples of 1-D manifolds include a line, a circle, and two separate circles.



Embedding

An embedding is a **representation of a topological object,** manifold, graph, field, etc. in a certain space in such a way that its connectivity or algebraic properties are preserved.

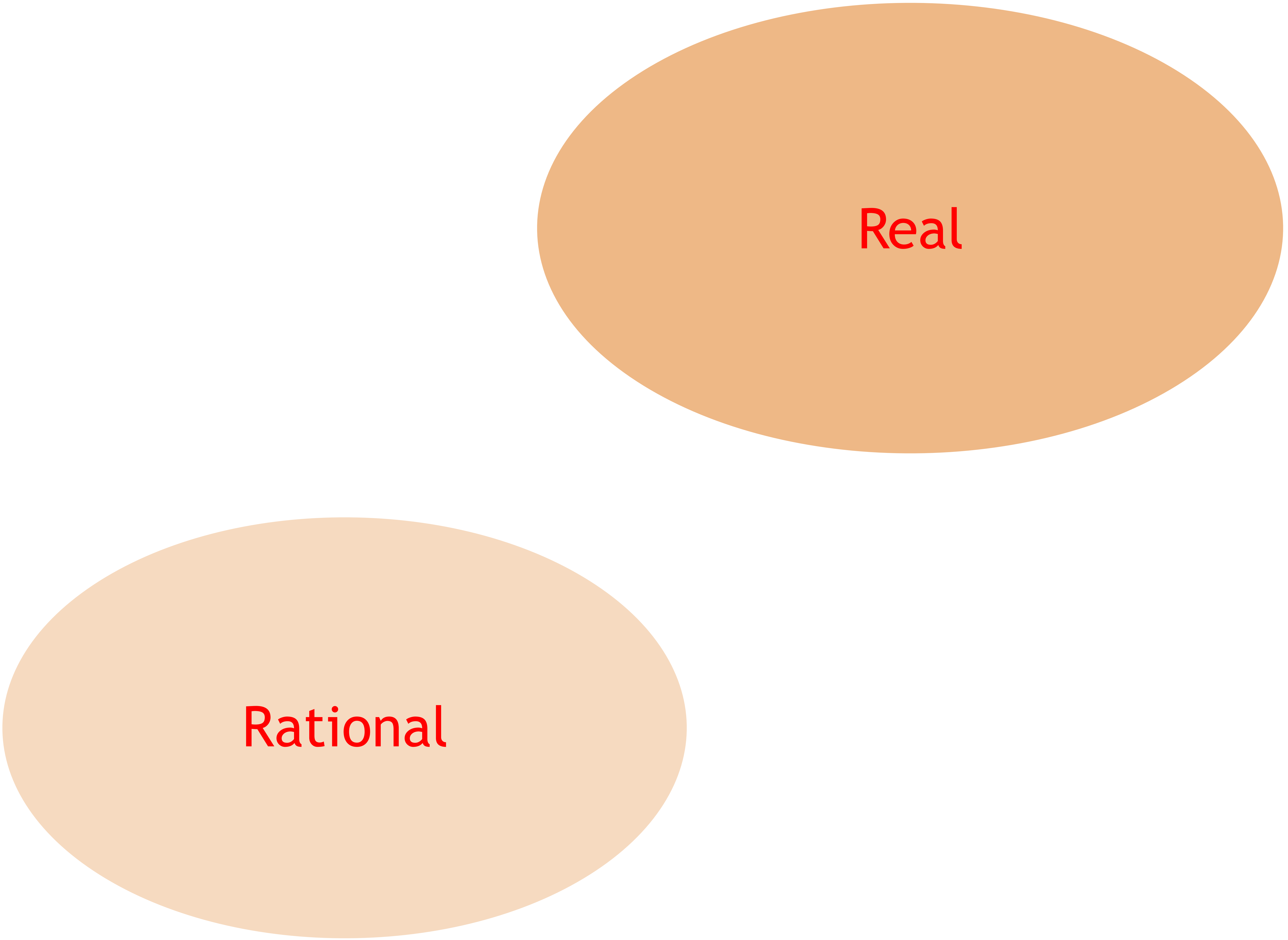
Examples:



Embedding

An embedding is a **representation of a topological object,** manifold, graph, field, etc. in a certain space in such a way that its connectivity or algebraic properties are preserved.

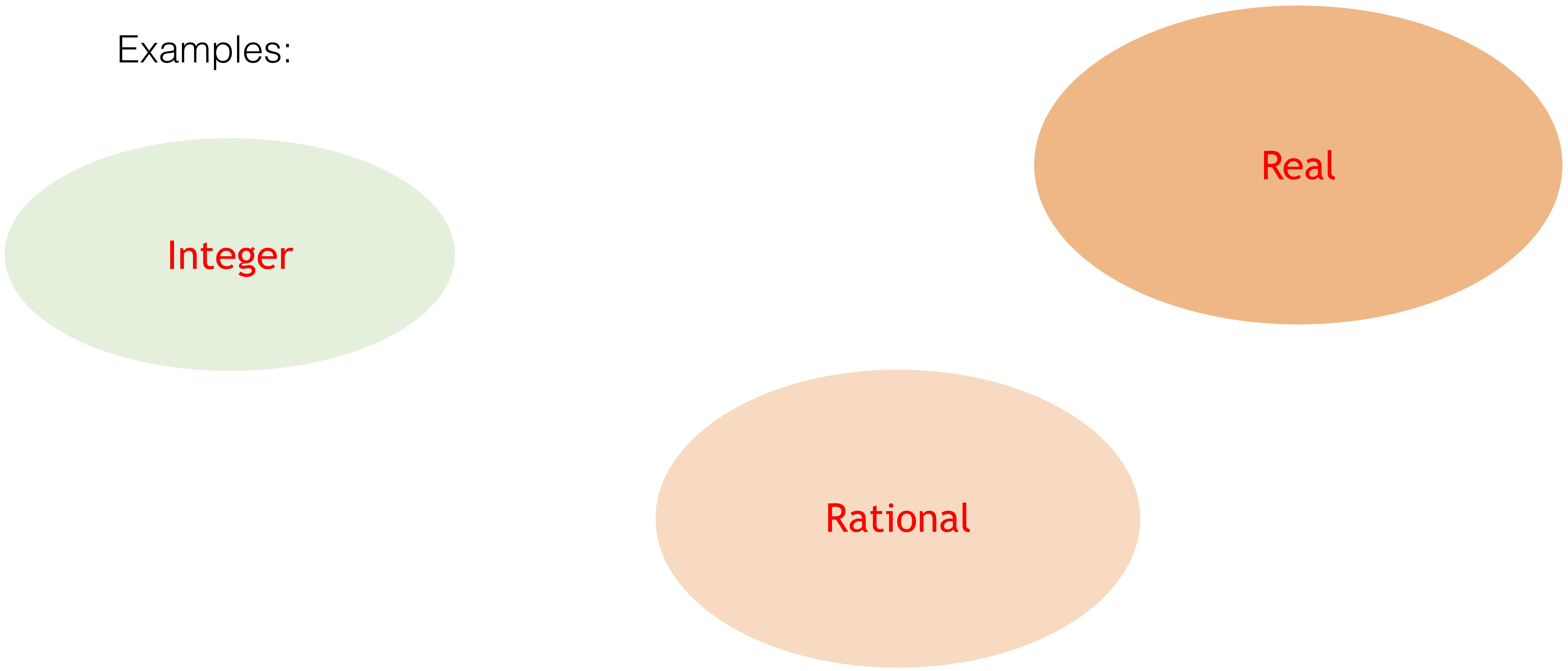
Examples:



Embedding

An embedding is a **representation of a topological object,** manifold, graph, field, etc. in a certain space in such a way that its connectivity or algebraic properties are preserved.

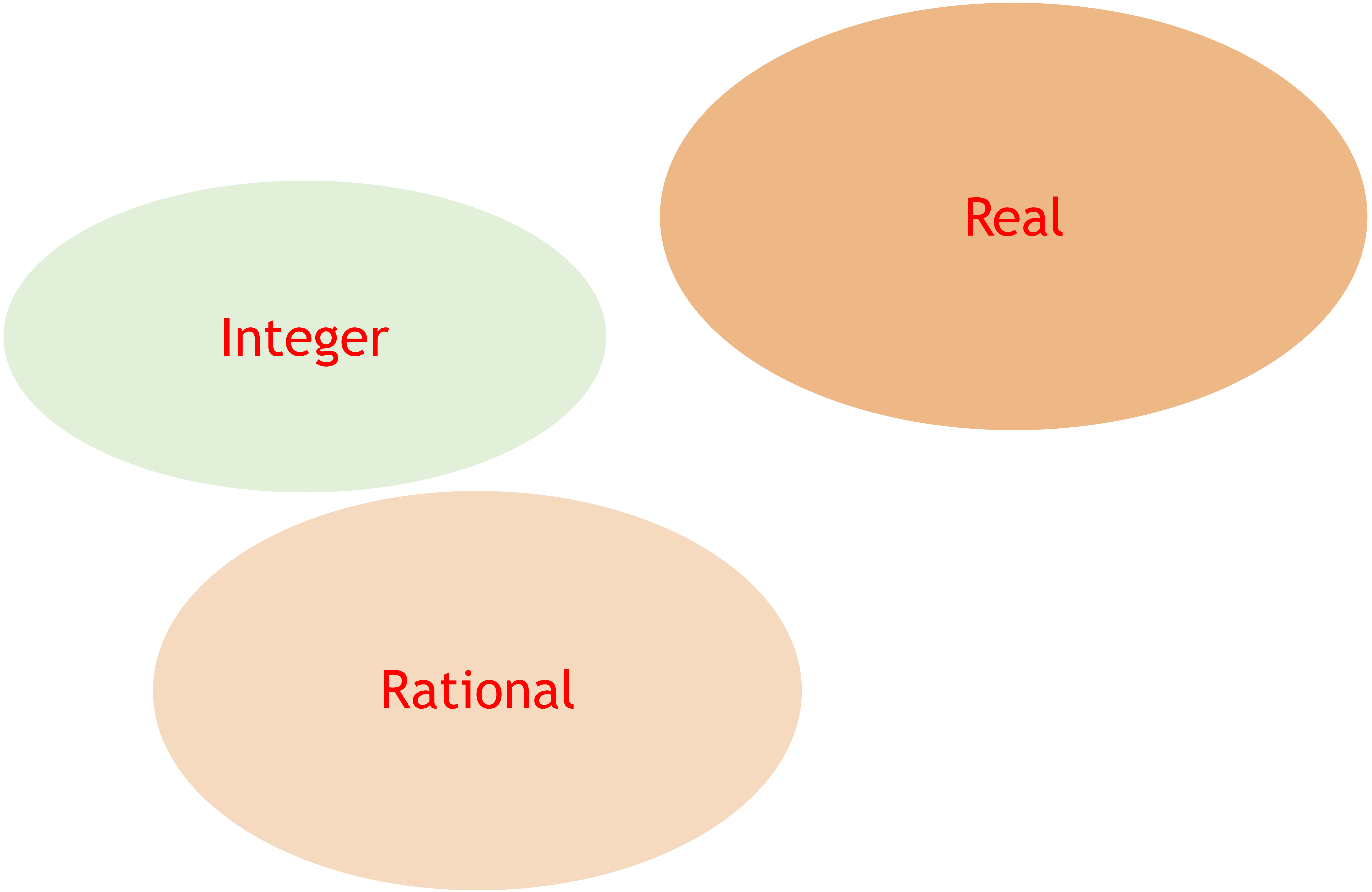
Examples:



Embedding

An embedding is a **representation of a topological object,** manifold, graph, field, etc. in a certain space in such a way that its connectivity or algebraic properties are preserved.

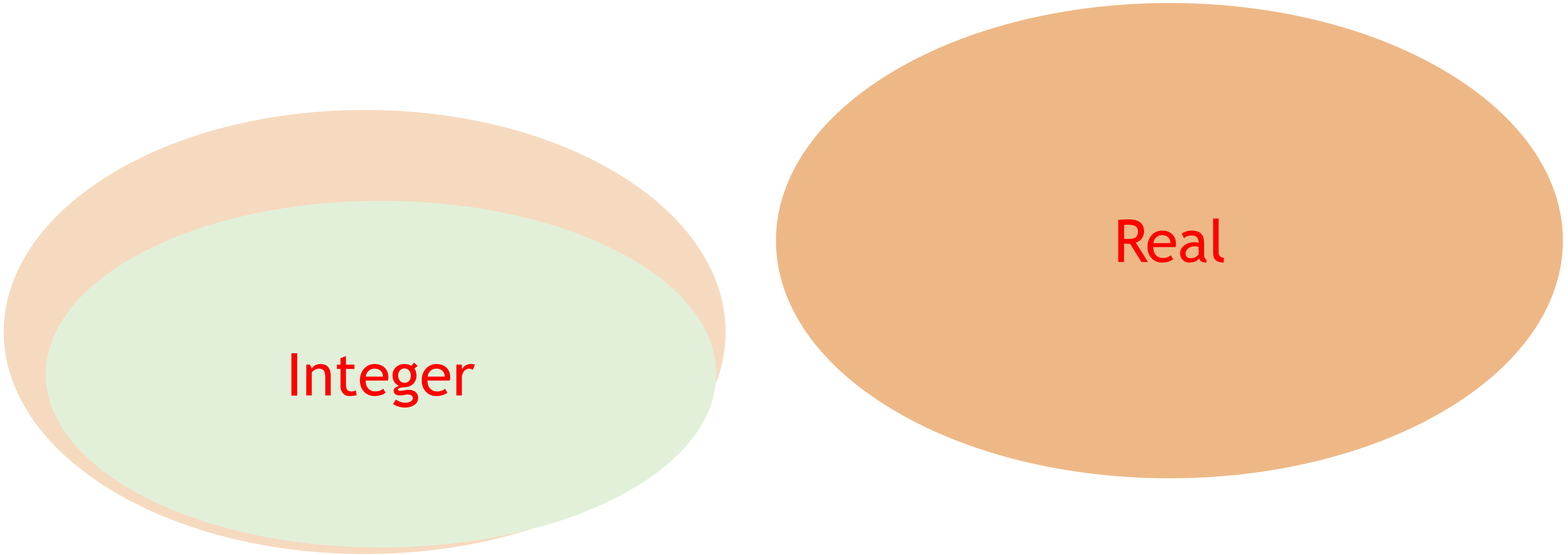
Examples:



Embedding

An embedding is a **representation of a topological object,** manifold, graph, field, etc. in a certain space in such a way that its connectivity or algebraic properties are preserved.

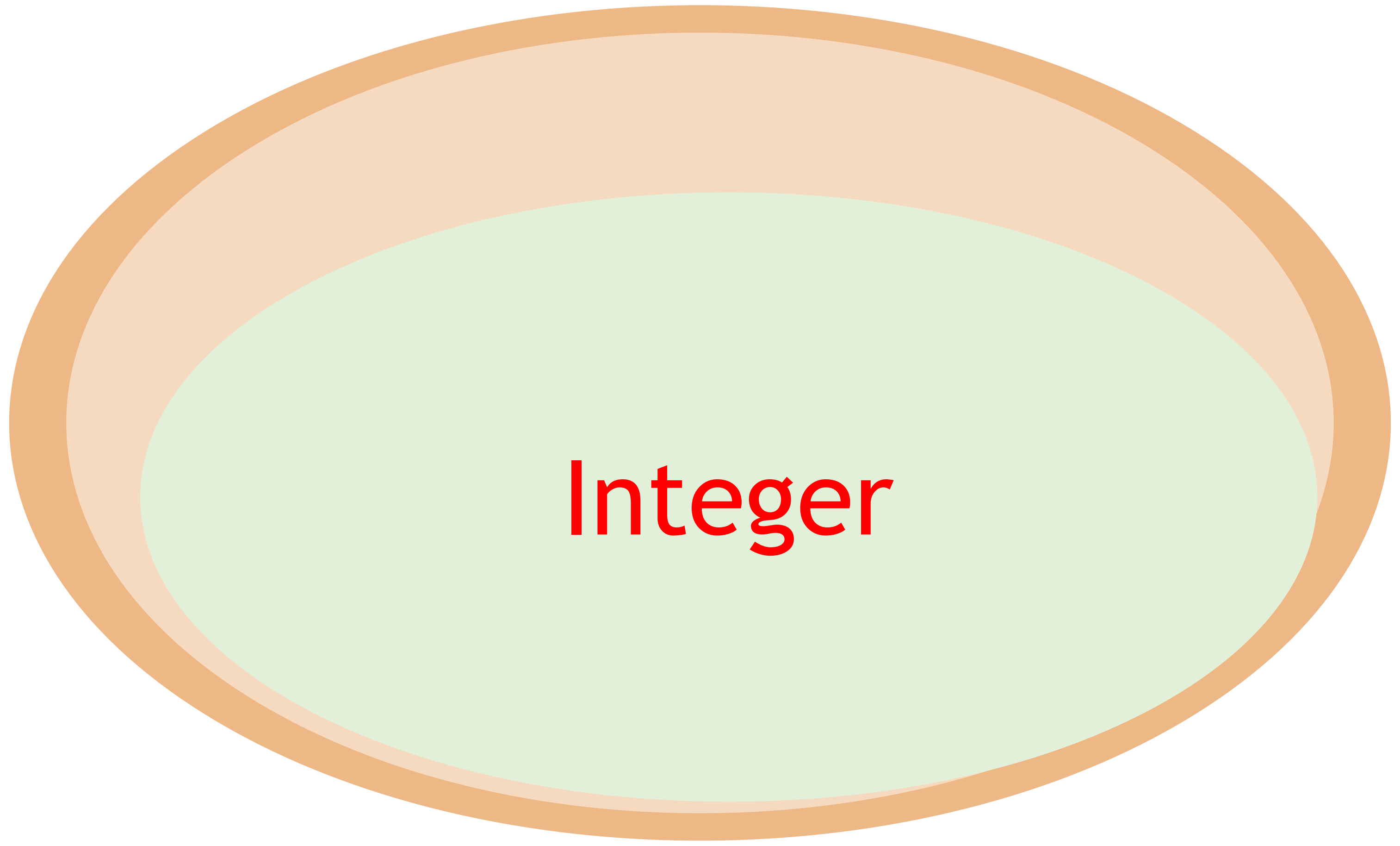
Examples:



Embedding

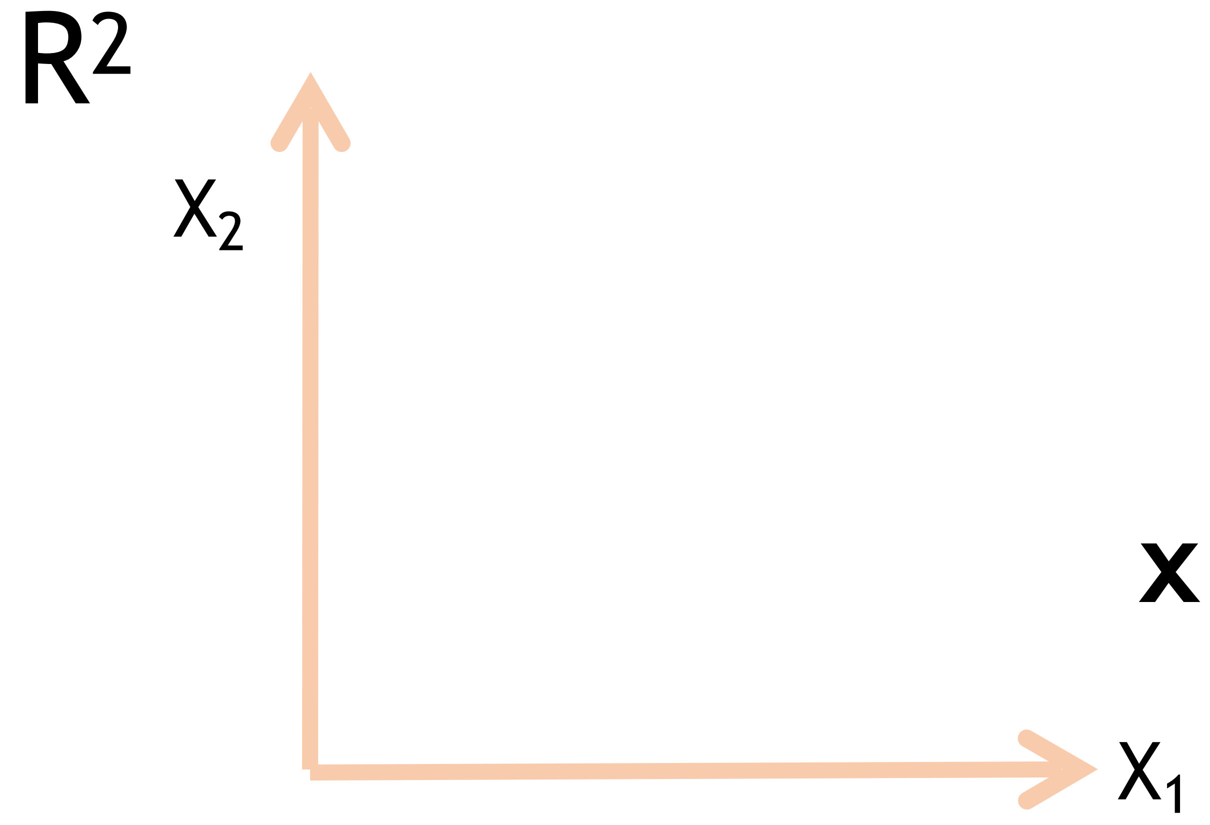
An embedding is a **representation of a topological object,** manifold, graph, field, etc. in a certain space in such a way that its connectivity or algebraic properties are preserved.

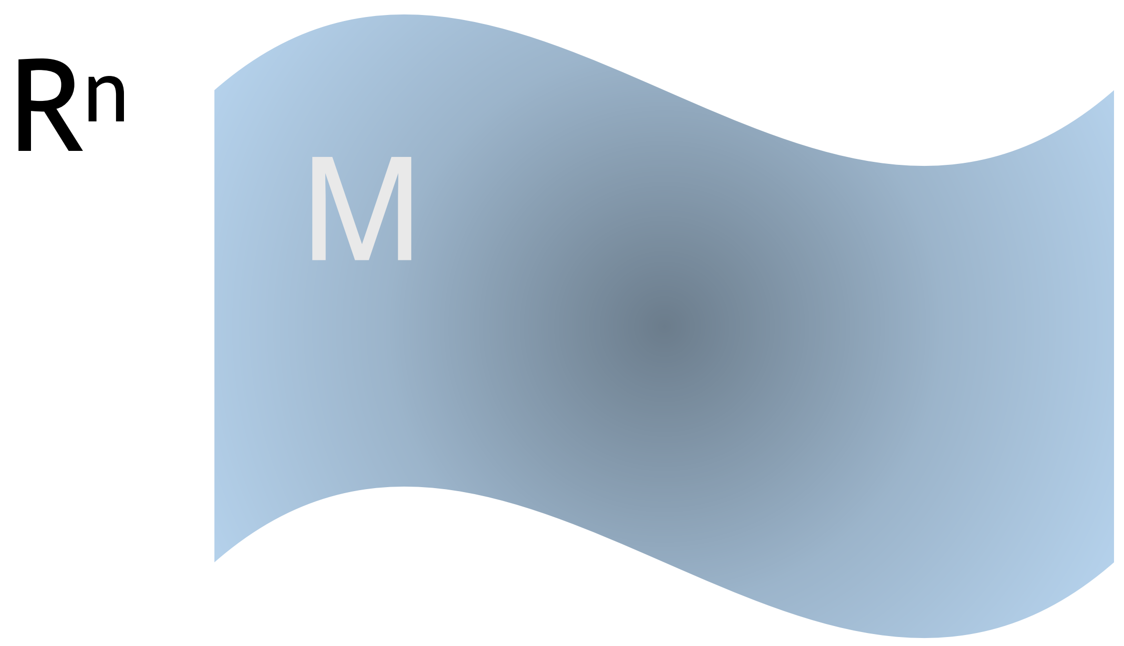
Examples:



Manifold and Dimensionality Reduction

- Manifold: generalized “subspace” in $\mathbf{R}^n$
- Points in a local region on a manifold can be indexed by a subset of $\mathbf{R}^k$ ($k<<n$)

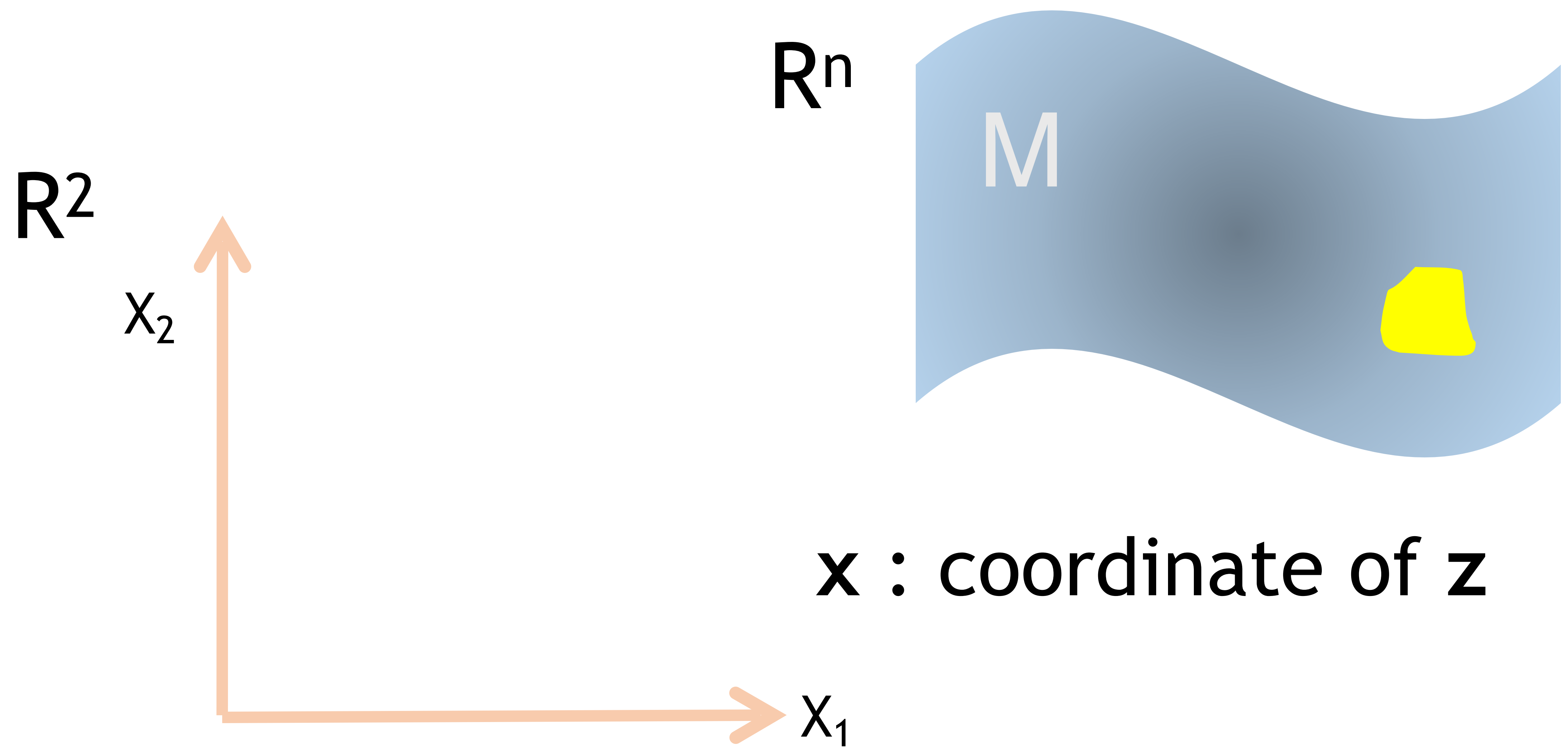




x : coordinate of **z**

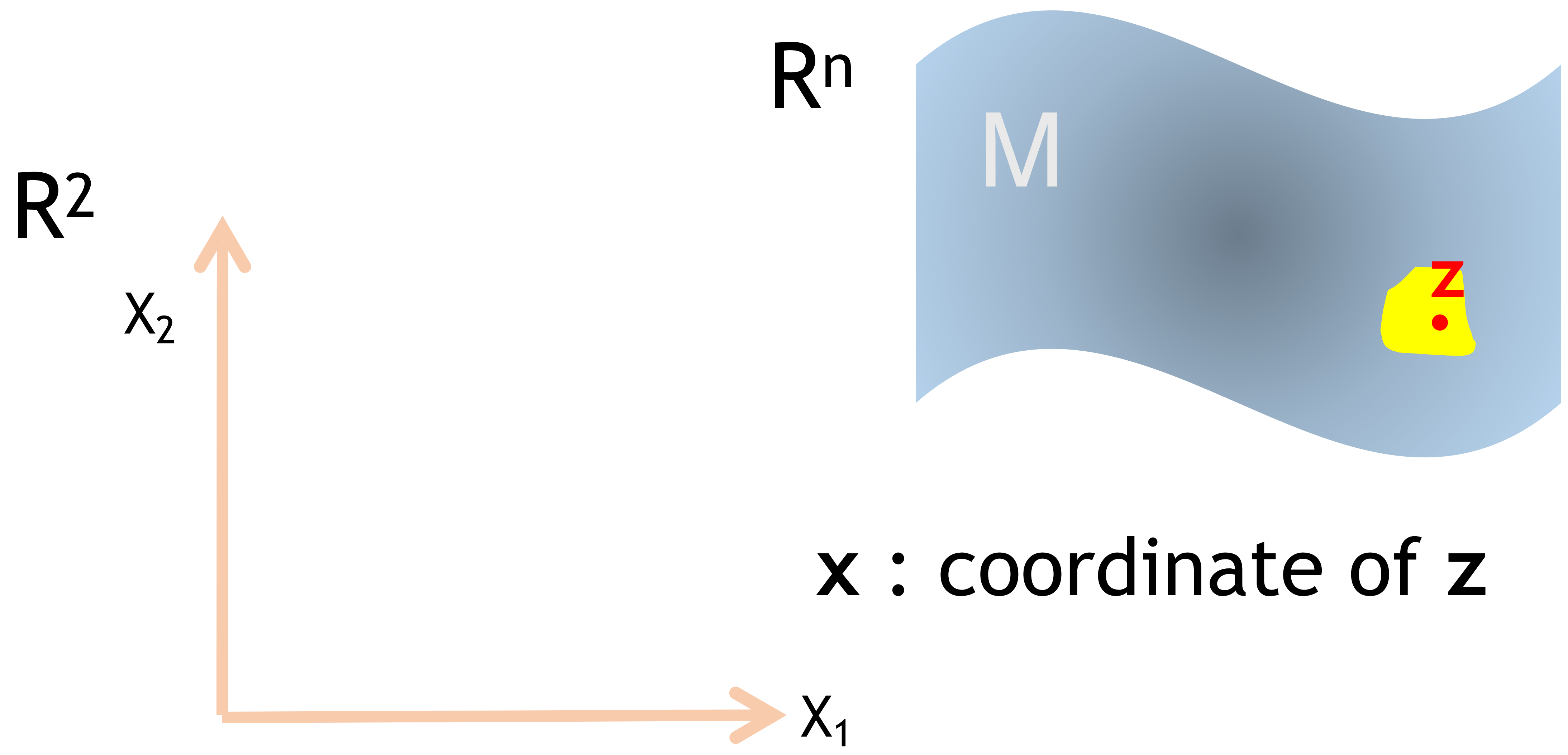
Manifold and Dimensionality Reduction

- Manifold: generalized "subspace" in $\mathbf{R}^n$
- Points in a local region on a manifold can be indexed by a subset of $\mathbf{R}^k$ ($k<<n$)



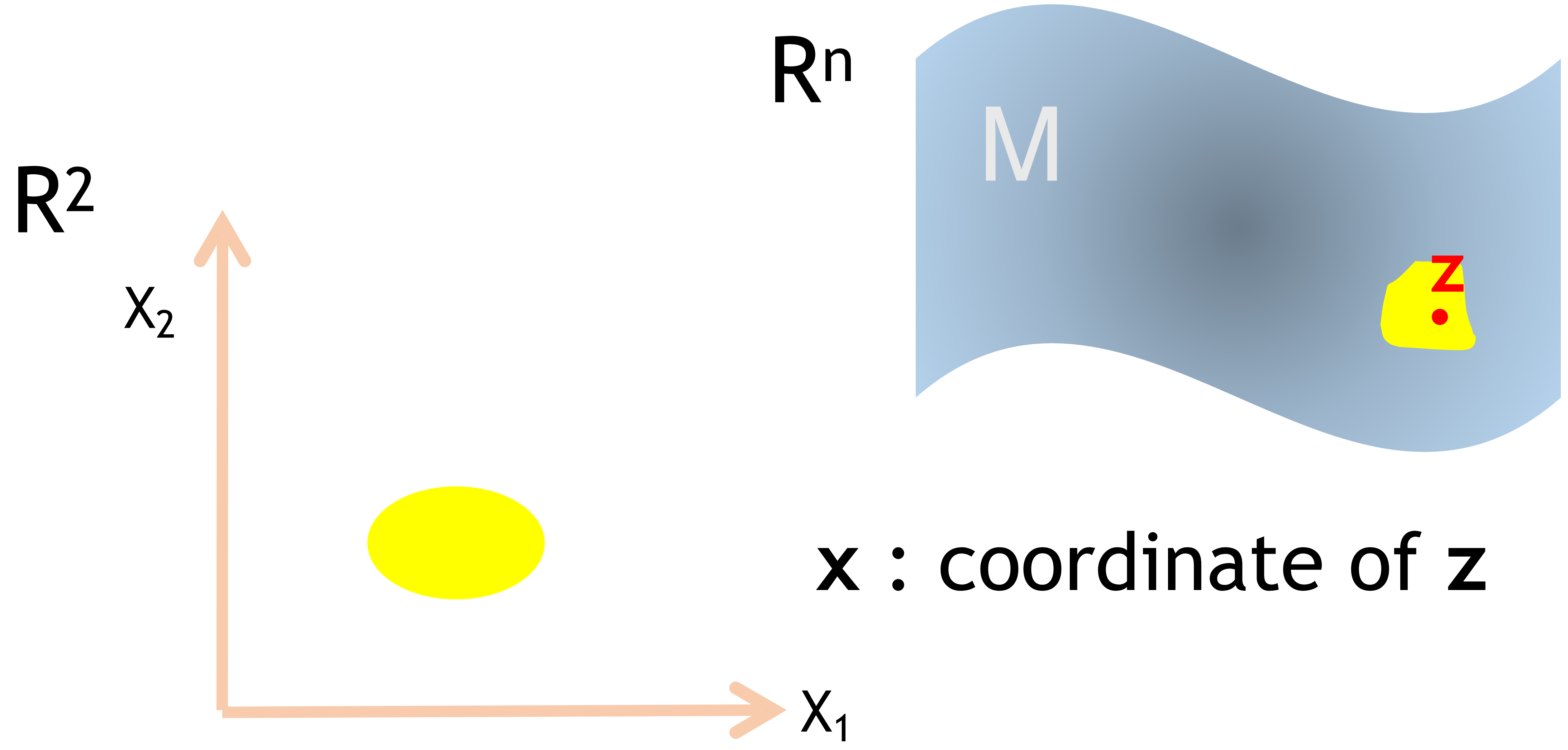
Manifold and Dimensionality Reduction

- Manifold: generalized “subspace” in $\mathbf{R}^n$
- Points in a local region on a manifold can be indexed by a subset of $\mathbf{R}^k$ (k<<n)



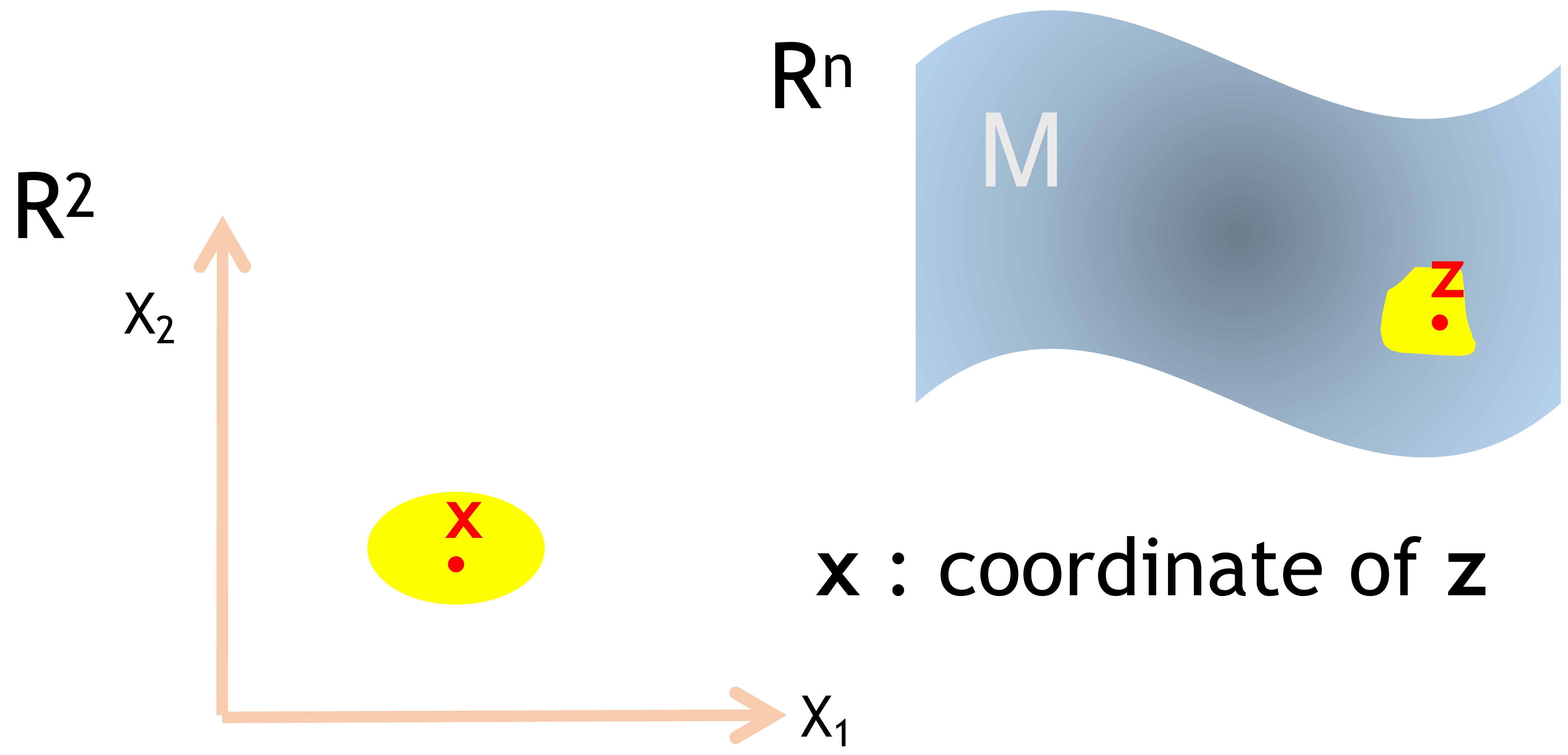
Manifold and Dimensionality Reduction

- Manifold: generalized "subspace" in $\mathbf{R}^n$
- Points in a local region on a manifold can be indexed by a subset of $\mathbf{R}^k$ ($k<<n$)



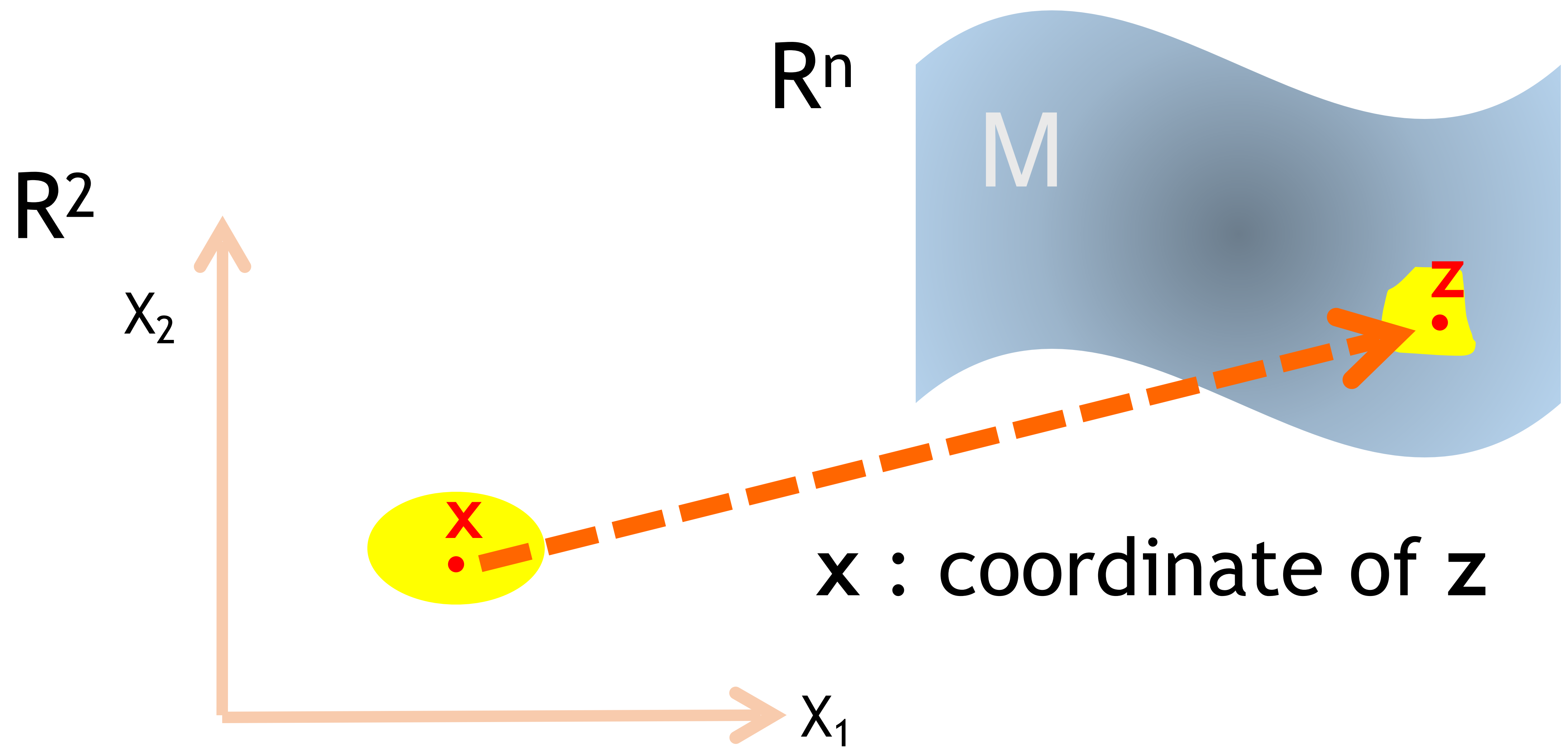
Manifold and Dimensionality Reduction

- Manifold: generalized "subspace" in $\mathbf{R}^n$
- Points in a local region on a manifold can be indexed by a subset of $\mathbf{R}^k$ ($k<<n$)



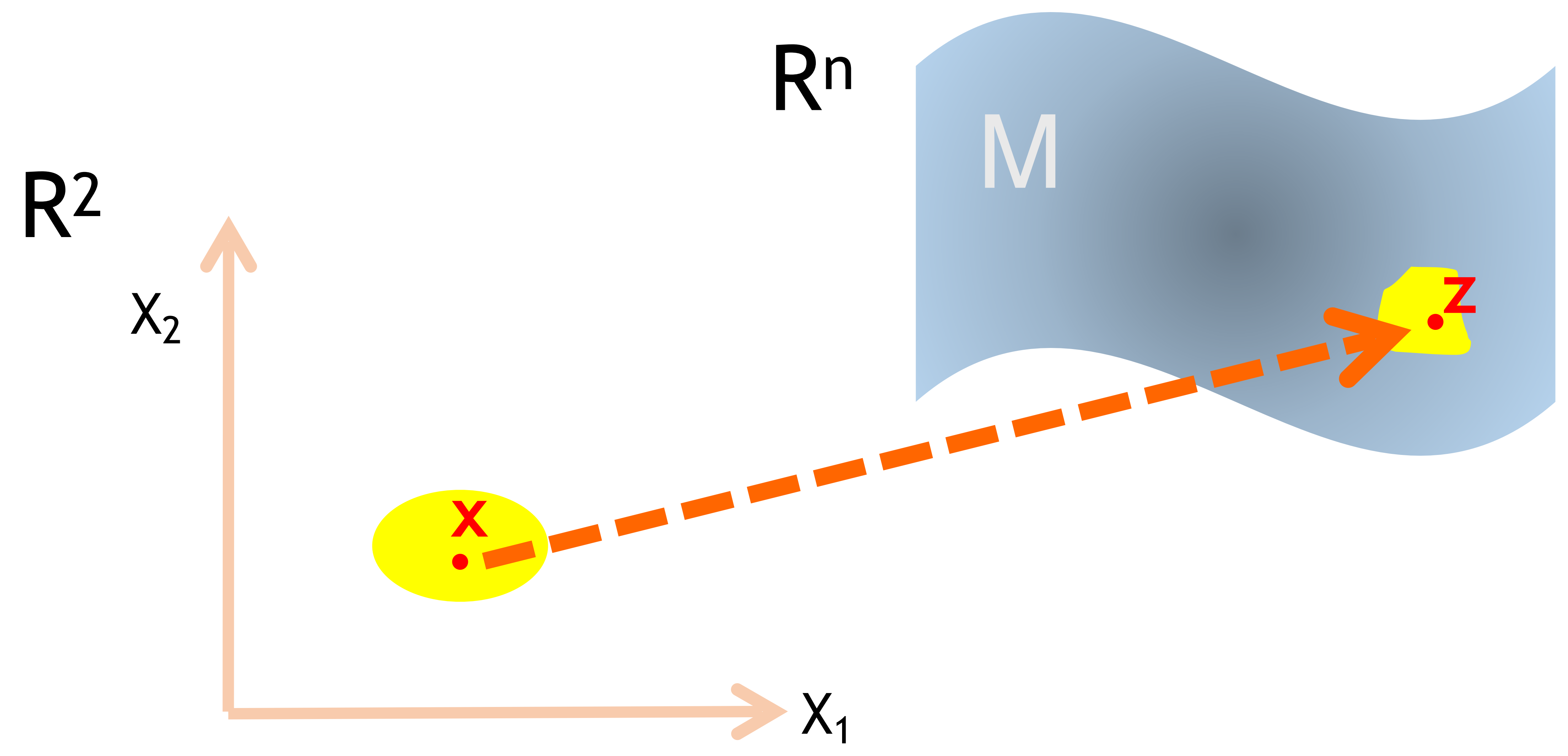
Manifold and Dimensionality Reduction

- Manifold: generalized "subspace" in $\mathbf{R}^n$
- Points in a local region on a manifold can be indexed by a subset of $\mathbf{R}^k$ (k<<n)



Manifold and Dimensionality Reduction

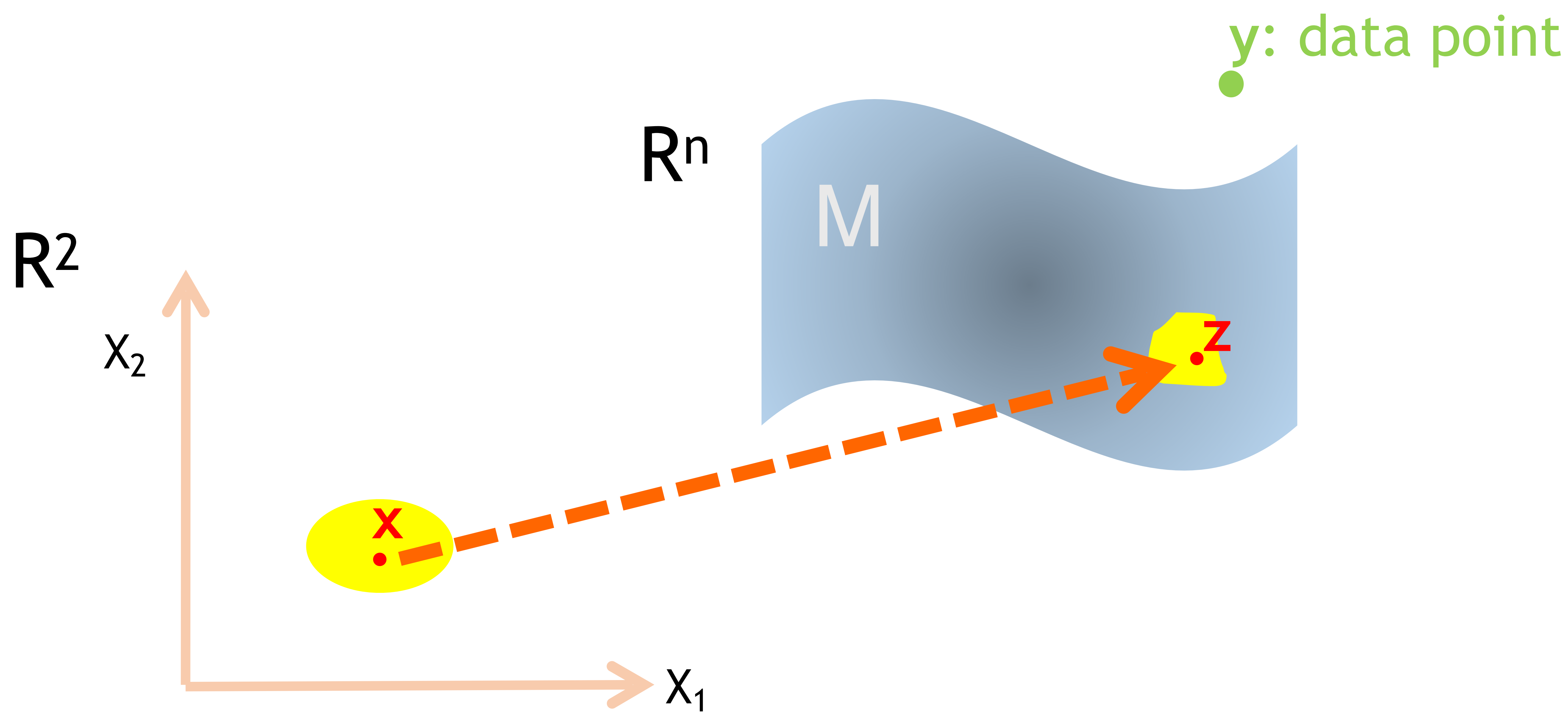
- If there is a global indexing scheme for **M that maps a data point y on M**



x : coordinate of **z**

Manifold and Dimensionality Reduction

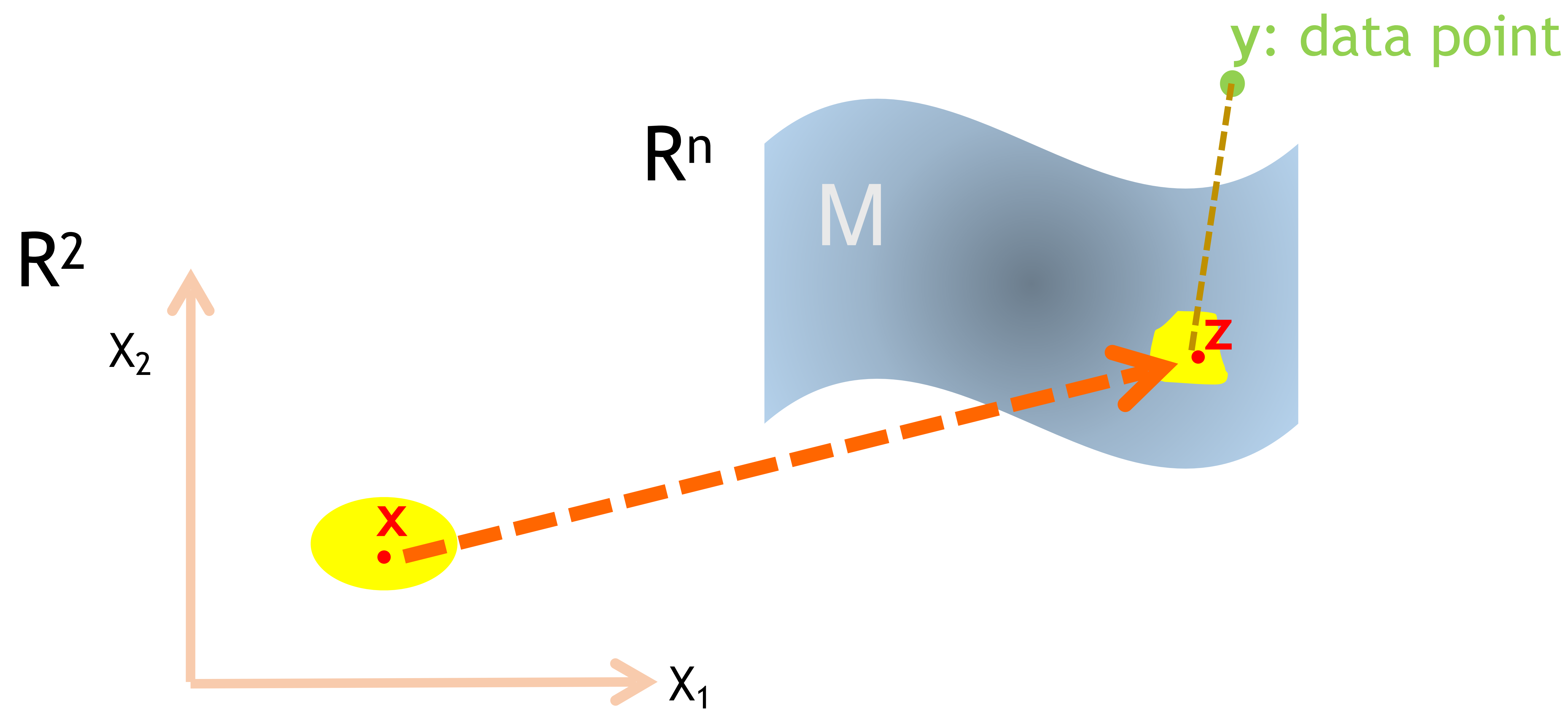
- If there is a global indexing scheme for **M that maps a data point y on M**



x : coordinate of **z**

Manifold and Dimensionality Reduction

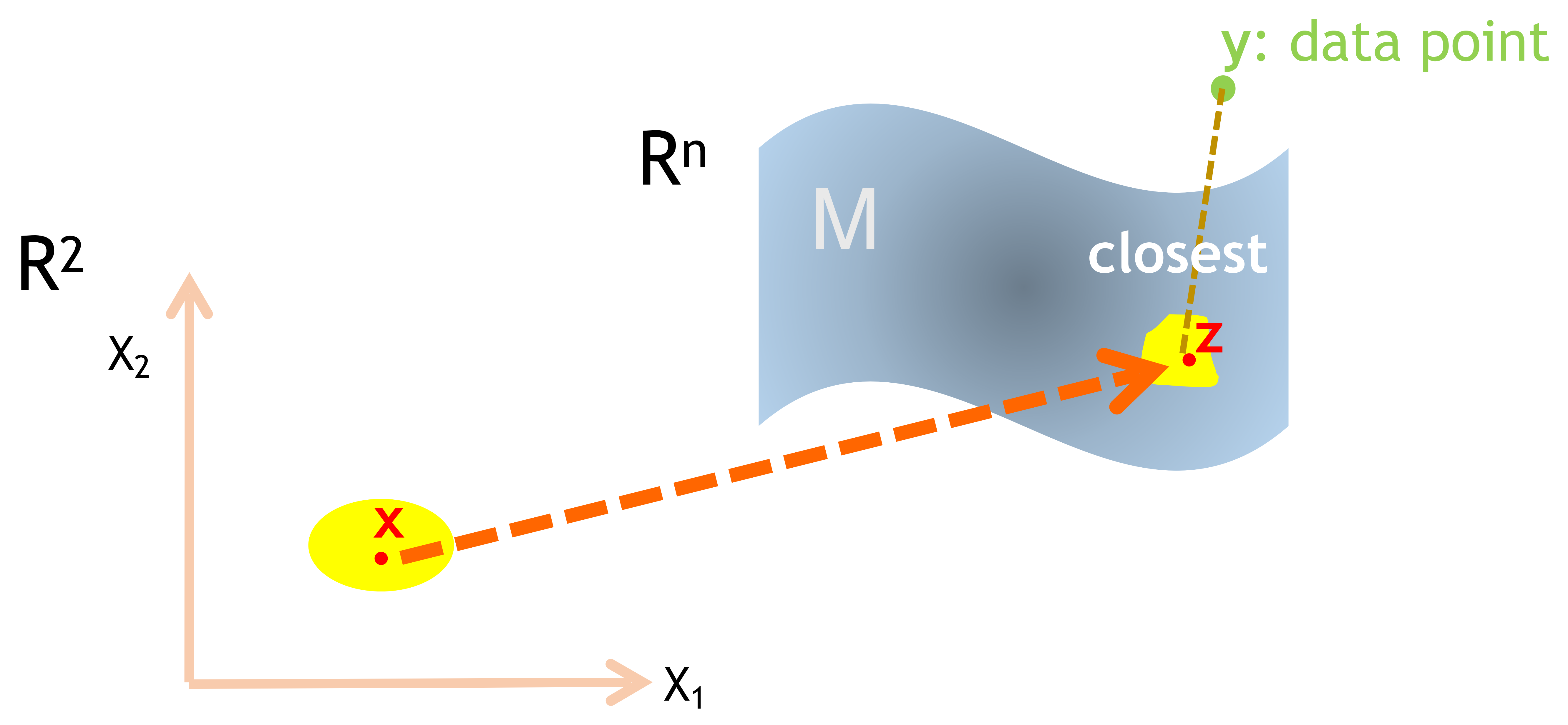
- If there is a global indexing scheme for **M that maps a data point y on M**



x : coordinate of **z**

Manifold and Dimensionality Reduction

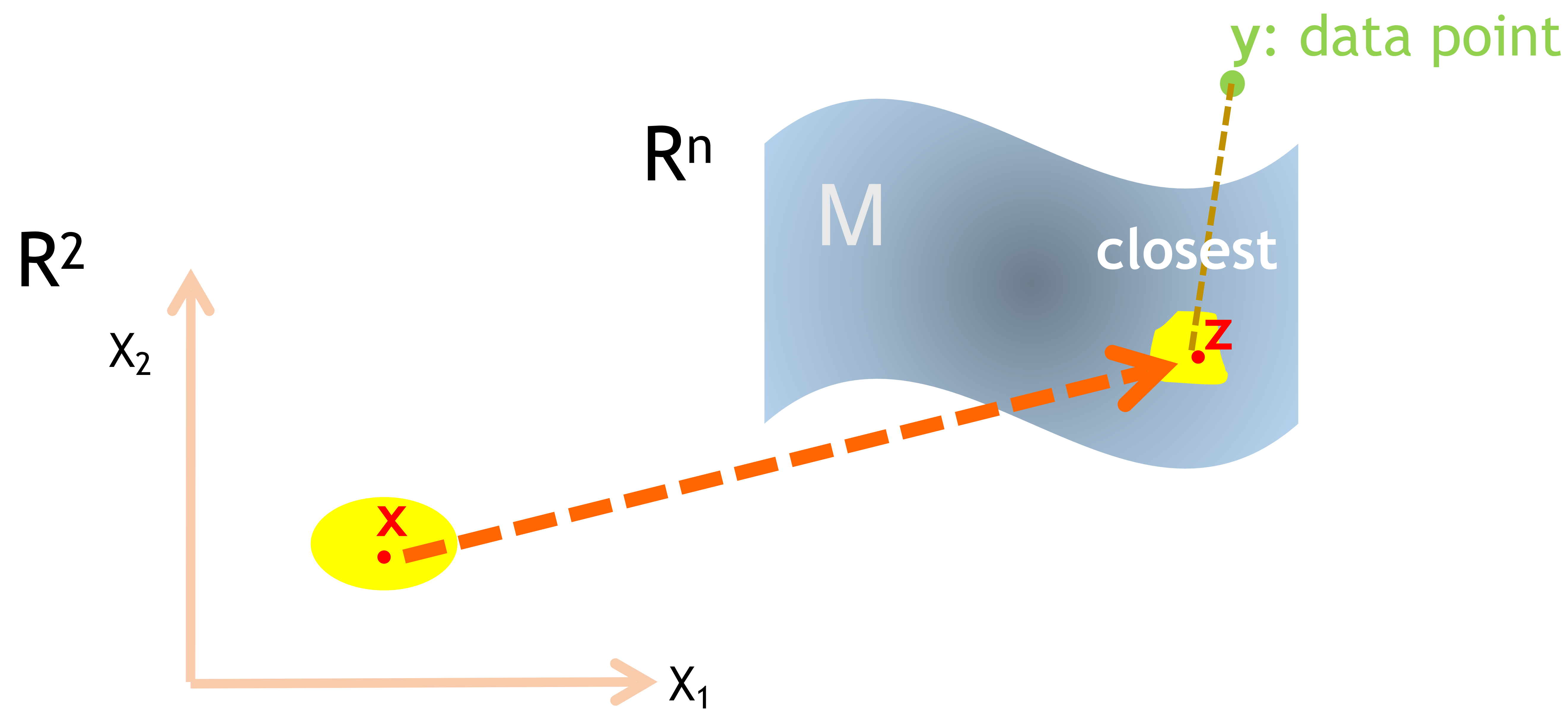
- If there is a global indexing scheme for **M that maps a data point y on M**



x : coordinate of **z**

Manifold and Dimensionality Reduction

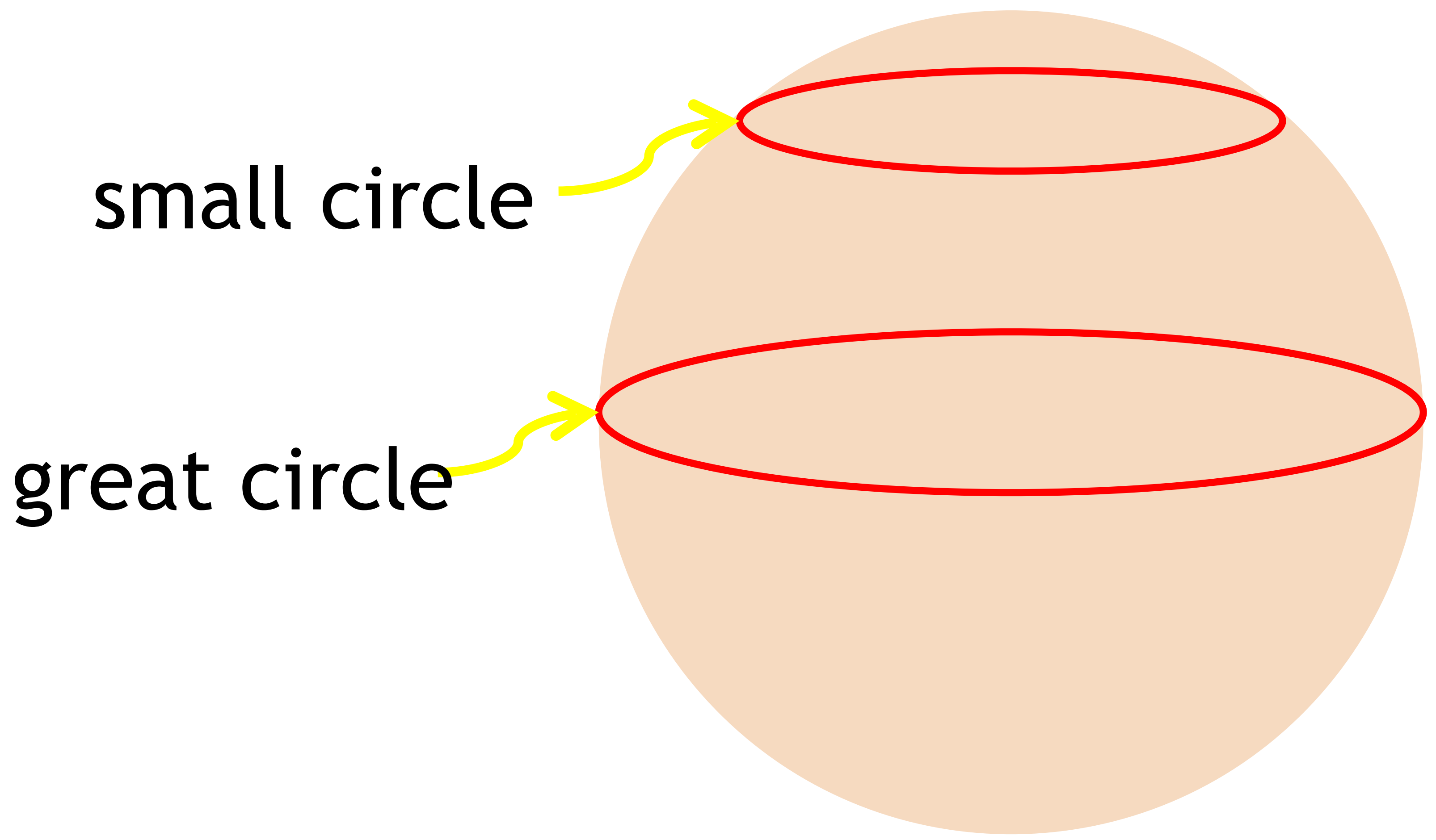
- If there is a global indexing scheme for **M that maps a data point y on M**



x : coordinate of **z** →reduced dimension representation of **y**

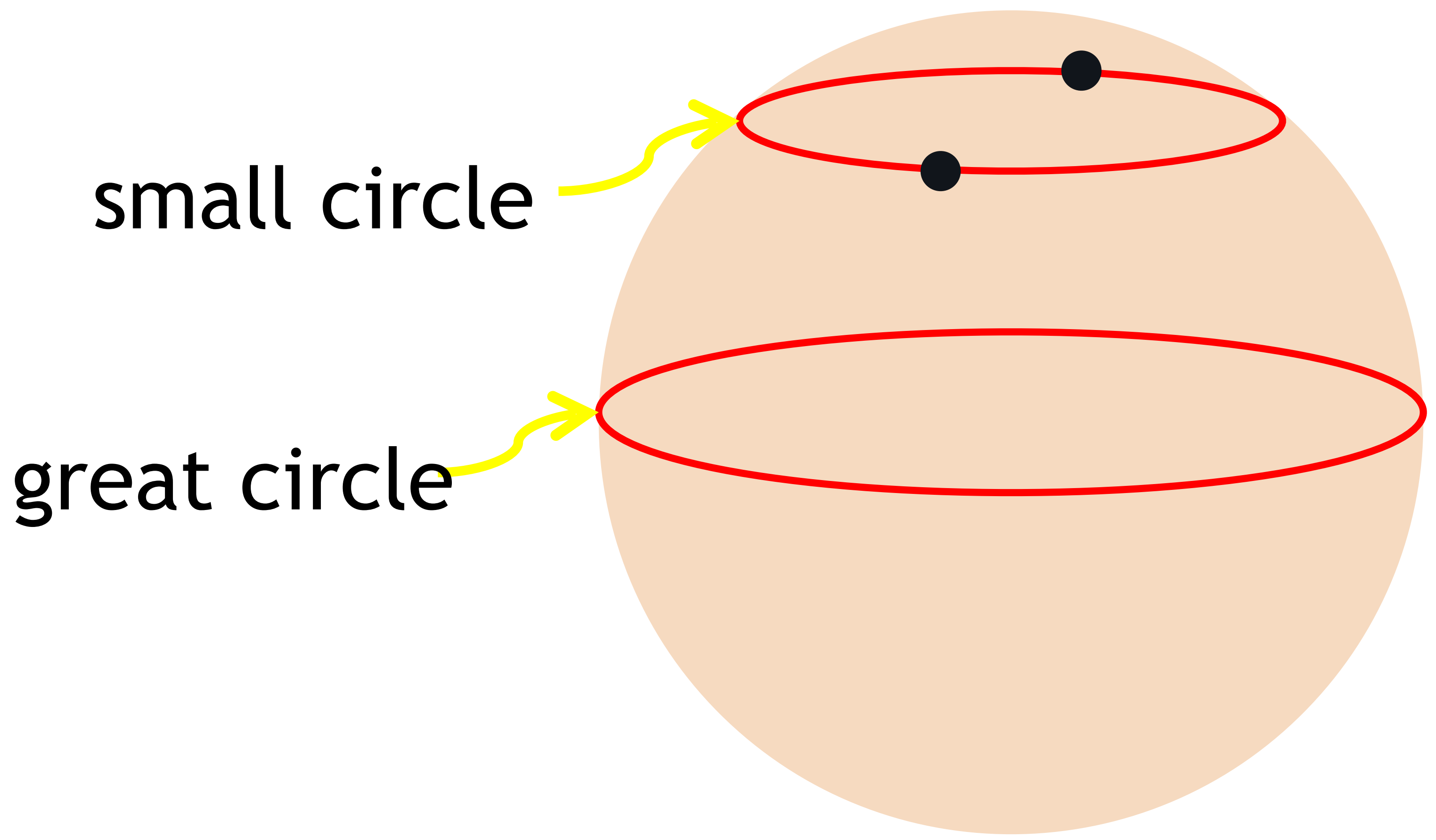
Geodesic Distance

- **Geodesic**: the shortest curve on a manifold that connects two points on the manifold
  - Example: on a sphere, geodesics are great circles
- **Geodesic distance**: length of the geodesic



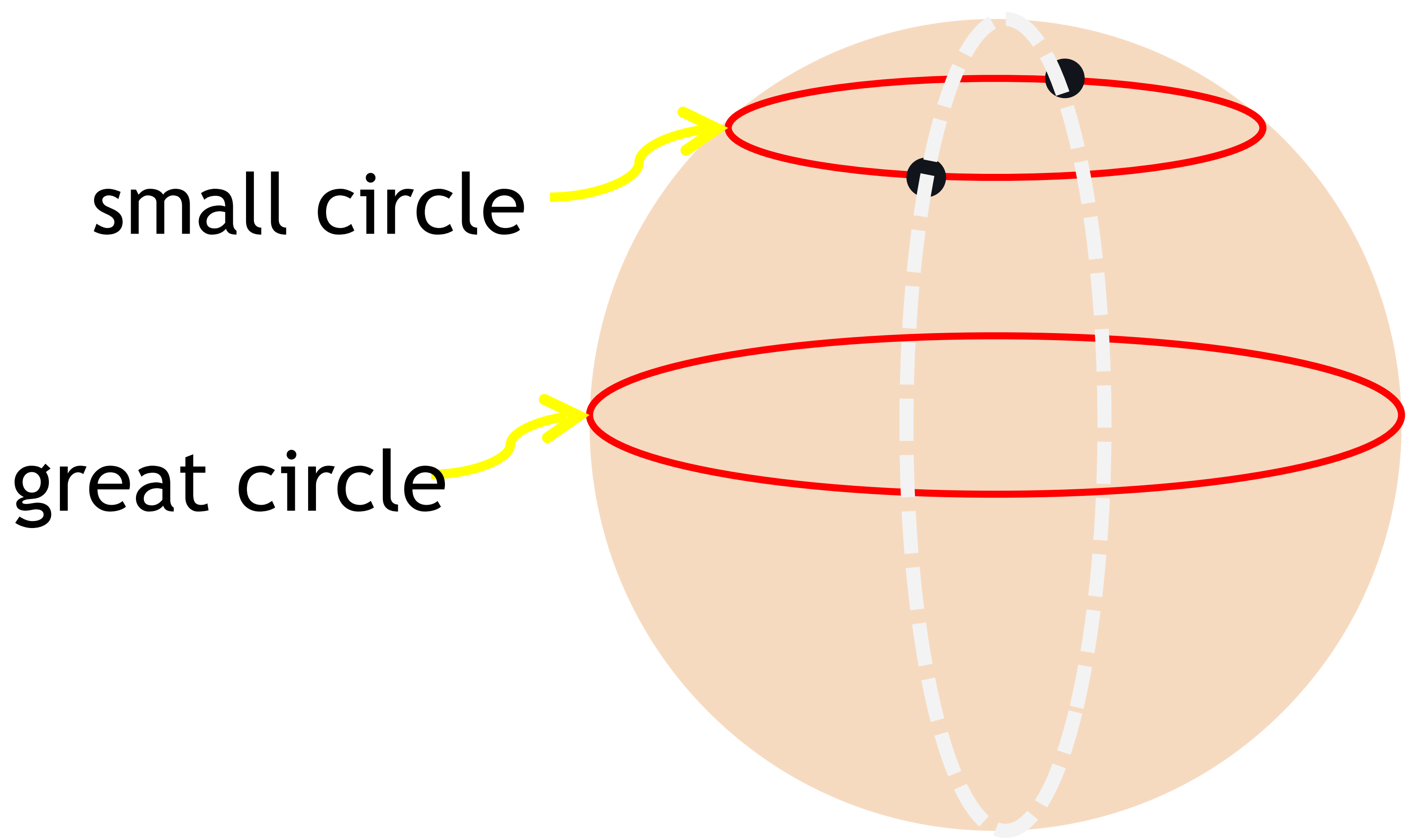
Geodesic Distance

- **Geodesic**: the shortest curve on a manifold that connects two points on the manifold
  - Example: on a sphere, geodesics are great circles
- **Geodesic distance**: length of the geodesic



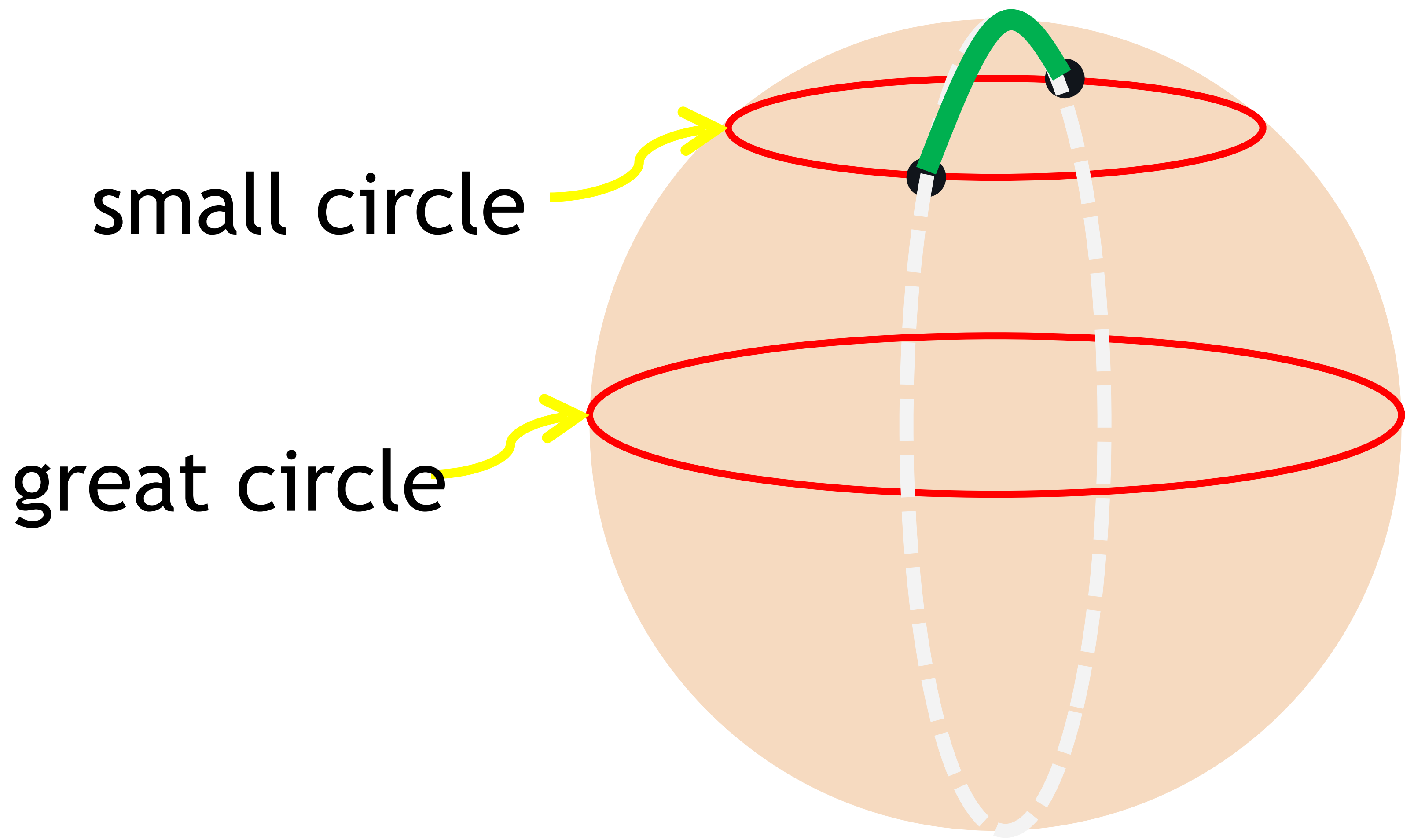
Geodesic Distance

- **Geodesic**: the shortest curve on a manifold that connects two points on the manifold
  - Example: on a sphere, geodesics are great circles
- **Geodesic distance**: length of the geodesic



Geodesic Distance

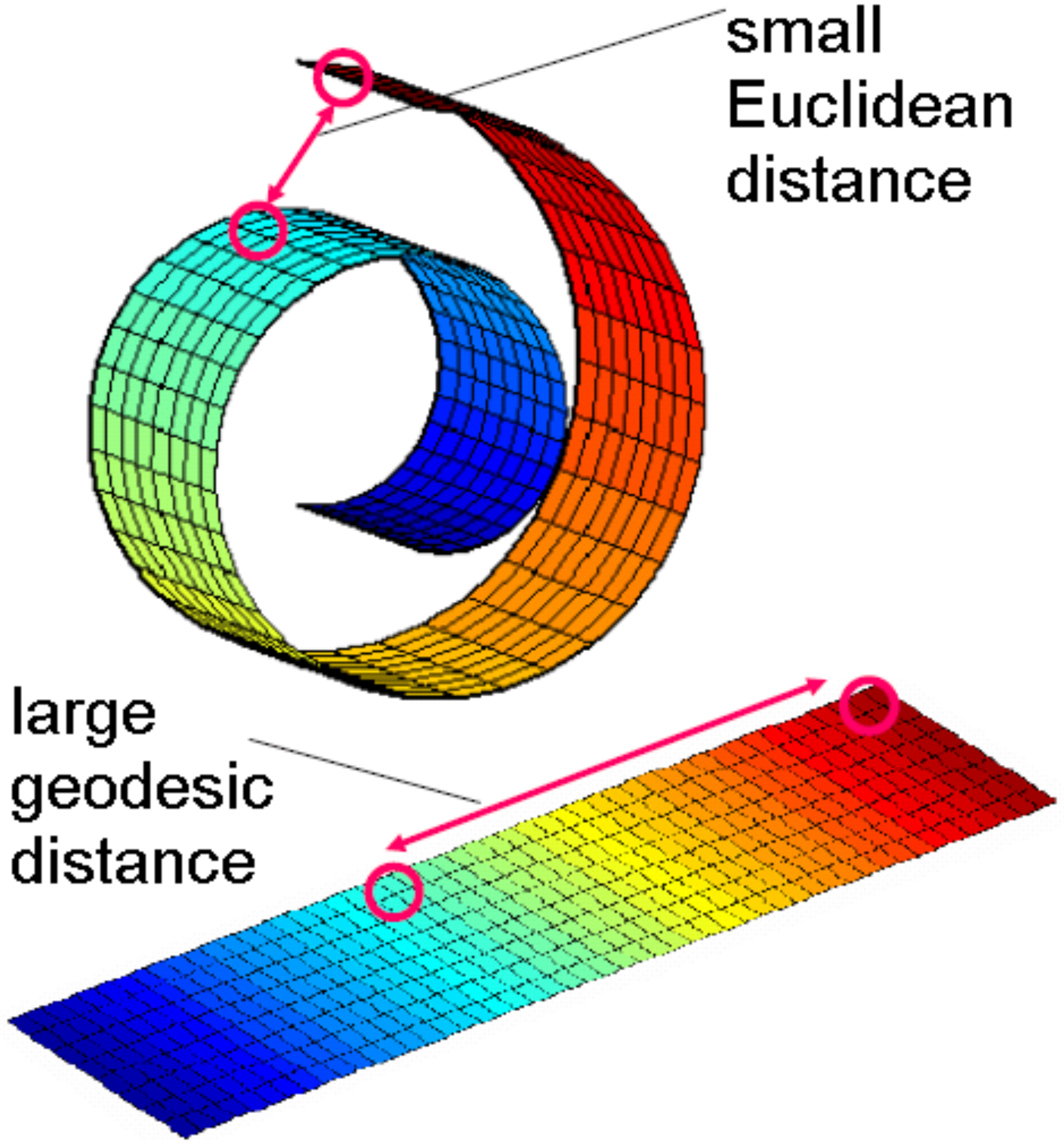
- **Geodesic**: the shortest curve on a manifold that connects two points on the manifold
  - Example: on a sphere, geodesics are great circles
- **Geodesic distance**: length of the geodesic



Geodesic Distance

- Euclidean distance **may not be** a good measure between two points on a manifold

- **Length of geodesic** is more appropriate



LLE and Laplacian Eigenmap

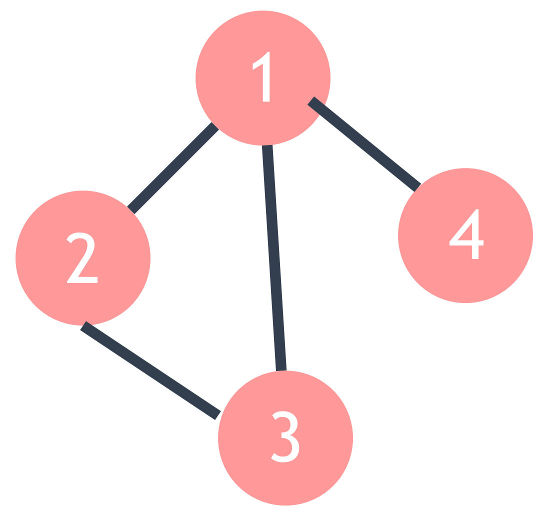
- The graph-based algorithms have **3 basic steps:**

1. Find K nearest neighbors.
2. Estimate local properties of manifold by looking at neighborhoods found in Step 1.
3. Find a global embedding that preserves the properties found in Step 2.

Lapalcian of a Graph

- Let G(V,E) be a undirected graph without graph loops. The Un-normalized Laplacian of the graph is

$$L=D-A$$

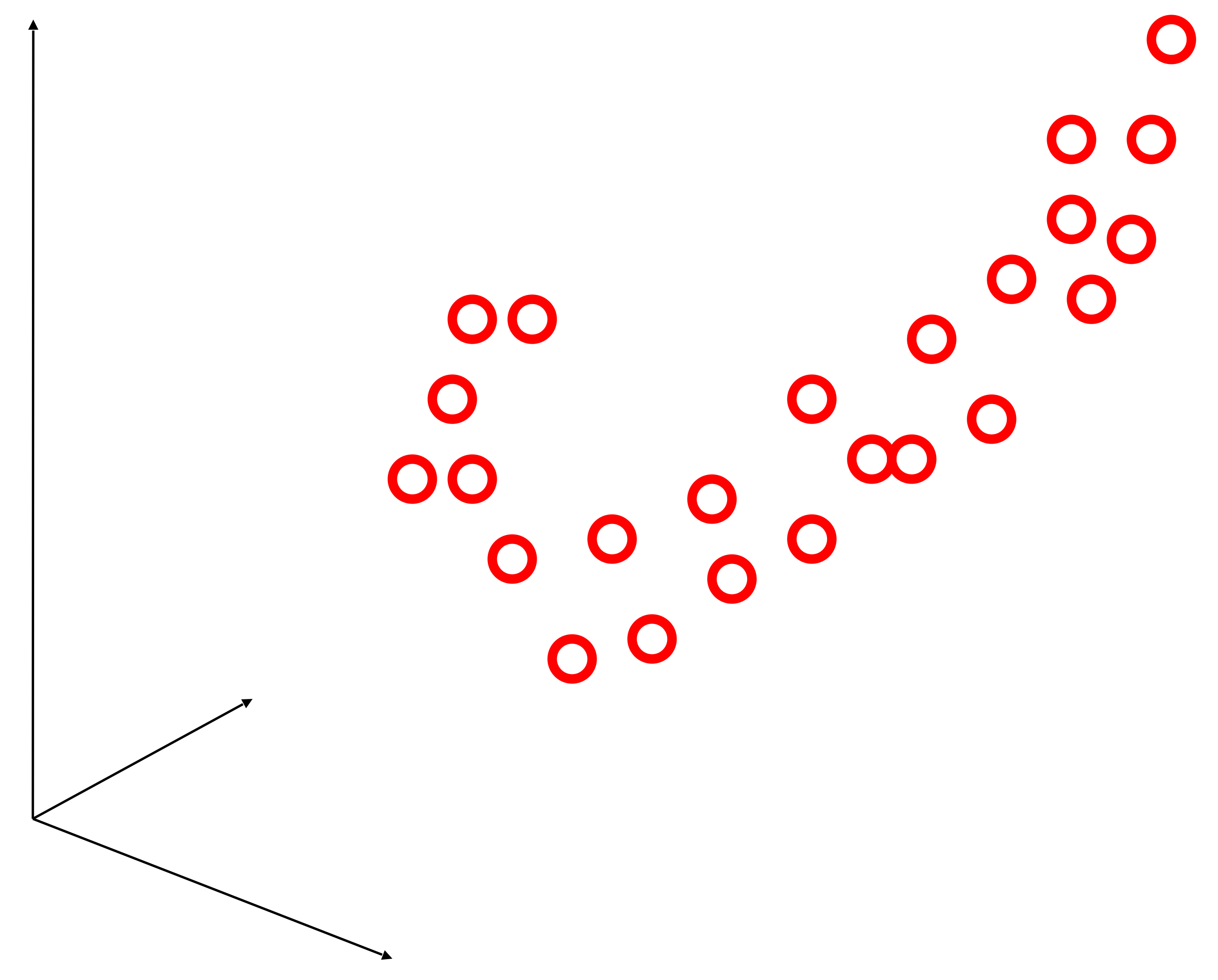


$$L=\begin{bmatrix}3 & -1 & -1 & -1\\ -1 & 2 & -1 & 0\\ -1 & -1 & 2 & 0\\ -1 & 0 & 0 & 1\end{bmatrix}=\underbrace{\begin{bmatrix}3 & 0 & 0 & 0\\ 0 & 2 & 0 & 0\\ 0 & 0 & 2 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}}_{D}-\underbrace{\begin{bmatrix}0 & 1 & 1 & 1\\ 1 & 0 & 1 & 0\\ 1 & 1 & 0 & 0\\ 1 & 0 & 0 & 0\end{bmatrix}}_{A}$$

Laplacian Eigenmap

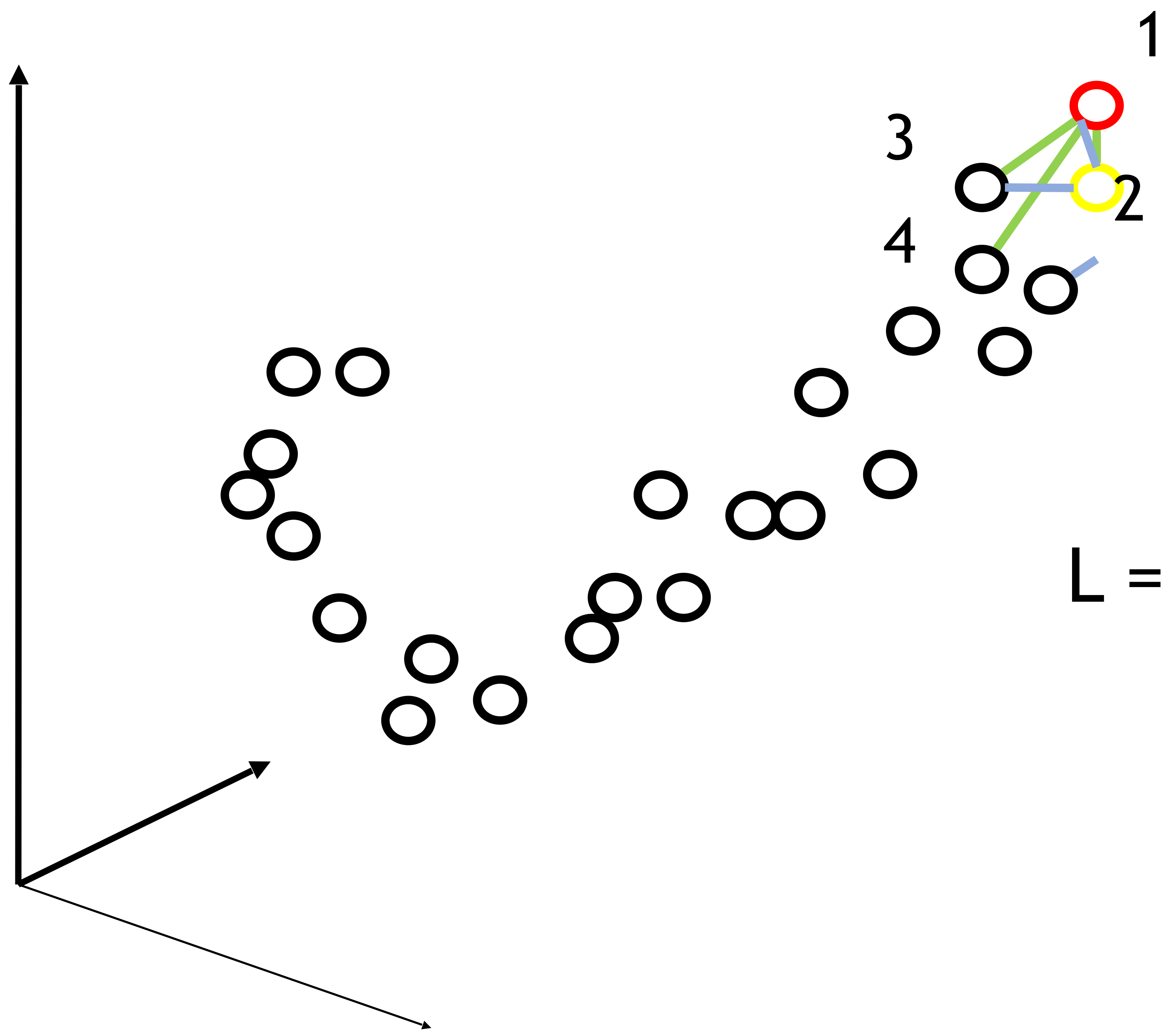
- Consider that X is a set o points in M , and M is a manifold embedded in R^n.

- Find $\underline{y}_1,.., \underline{y}_n$ in R^m such that $\underline{y}_i$ represents $\underline{x}_i$(m<<n)



Laplacian Eigenmap

- Construct the adjacency graph to approximate the manifold



$$L = \begin{bmatrix} 3 & -1 & & \\ -1 & 3 & & \\ -1 & -1 & & \\ -1 & 0 & & \\ 0 & -1 & & \\ \vdots & 0 & & \\ & \vdots & & \end{bmatrix} \qquad L=D-A$$

Laplacian Eigenmap

There are two variations for W (weight matrix)

- **simple-minded** (1 if connected, 0 o.w.)

- **heat kernel** (t is real)

$$A_{ij} = e^{-\frac{\|x_i - x_j\|^2}{t}}$$

Laplacian Eigenmap N-Dimensional case

- Now we consider the more general problem of embedding the graph into **m-dimensional** Euclidean space

- Let Y be such a nxm map

$$Y=\begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1m} \\ y_{21} & y_{22} & \cdots & y_{2m} \\ \vdots & \vdots & \cdots & y_{nm} \end{bmatrix}$$

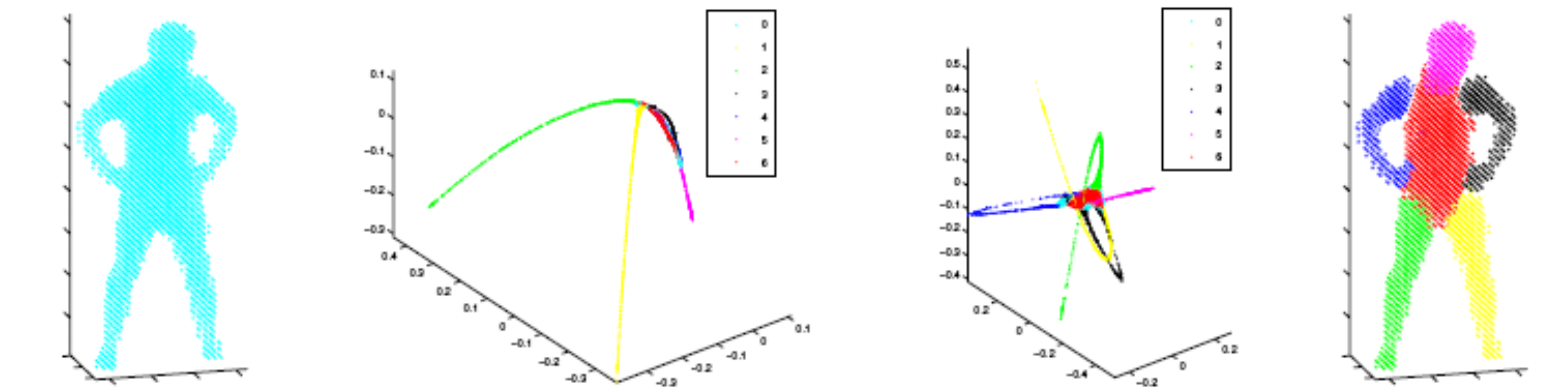
- N-dimensional **dirichlet** energy

$$argmin_Y \; trace(Y'LY)$$
$$with$$
$$Y'Y = I$$

Solutions are the first m eigenvectors

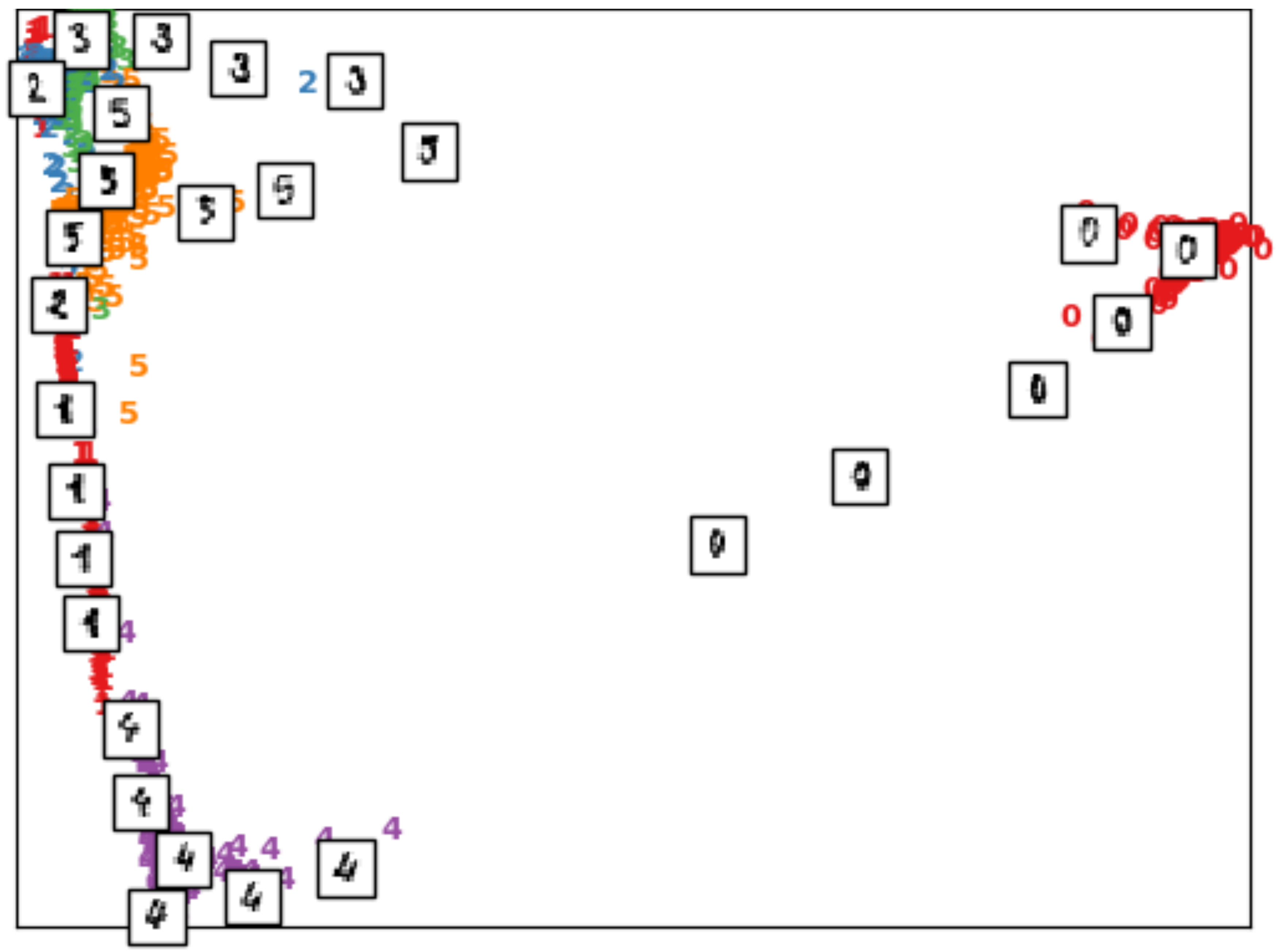
Applications

- We can apply manifold learning to pattern recognition (face, handwriting etc)
- Recently, ISOMAP and Laplacian eigenmap are used to initialize the human body model.



(a) Normal space (b) Eig. 1-3 (c) Eig. 4-6 (d) Segmented

Handwritten digit visualization



Considerations

- PROS:

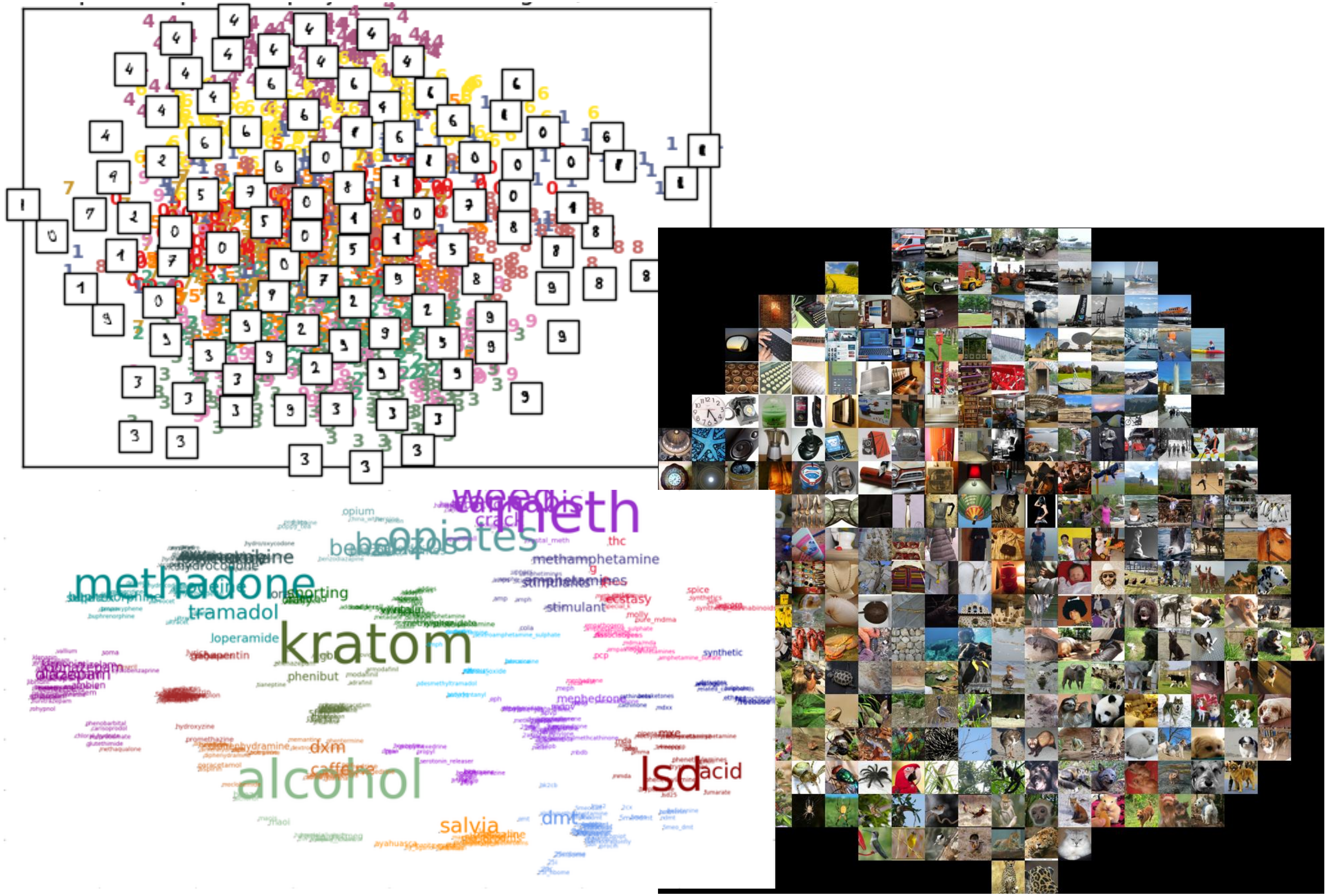
Laplacian eigenmap provides a computationally efficient approach to non-linear dimensionality reduction that has locality preserving properties

BUT

Laplacian Eigenmap attempts to approximate or preserve neighborhood information

If you need GLOBAL consistency? -> Look at the ISOMAP method

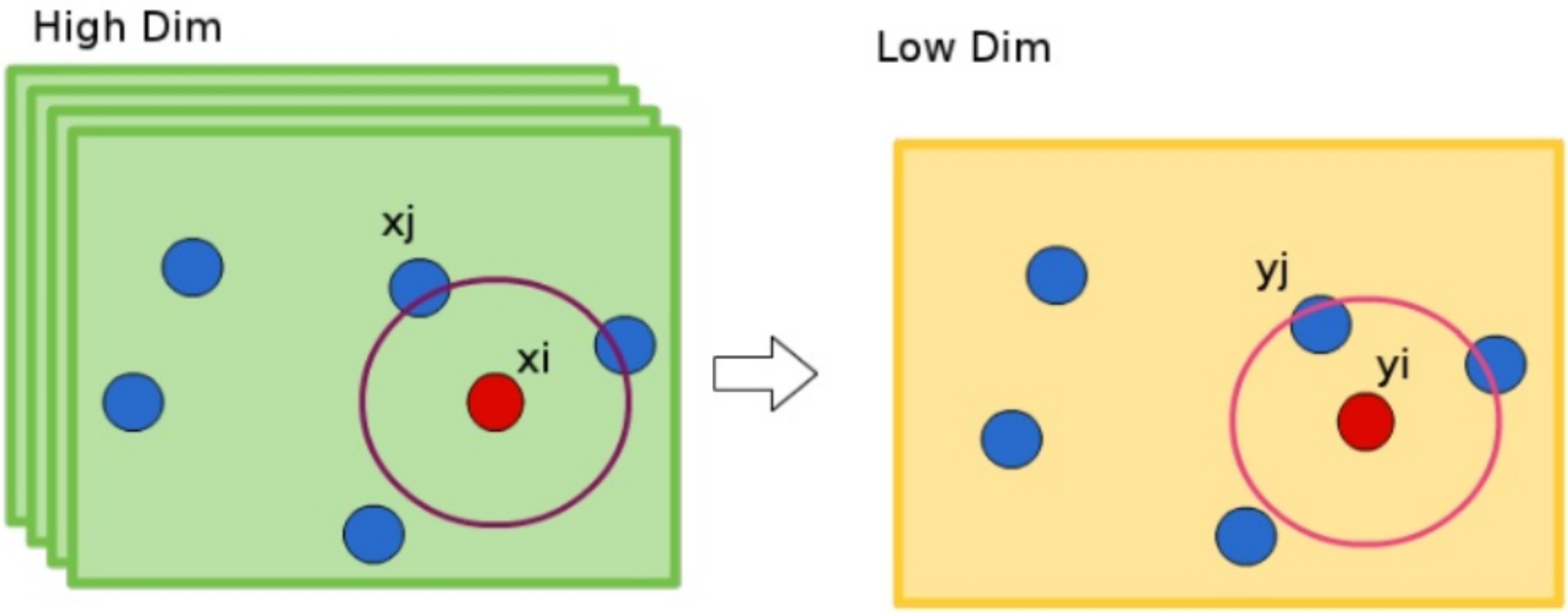
T-SNE



meth
opiates
crack
opium
thc
methamphetamine
amphetamines
stimulant
ecstasy
spice
synthetic
methadone
tramadol
loperamide
kratom
gabapentin
phenibut
diazepam
mephedrone
dxm
alcohol
lsd
acid
salvia
dmt

T - SNE

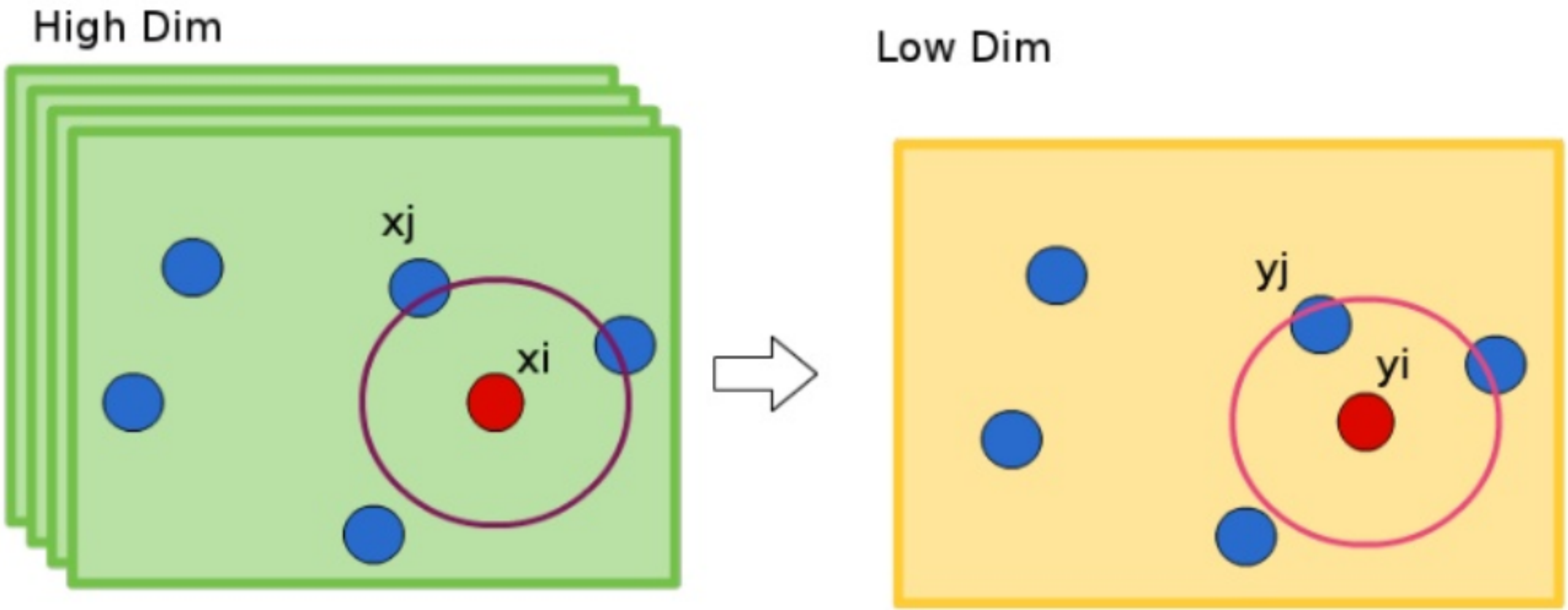
- Map point from High Dimensional space (x)to low dimensional space (y) preserving points distributions
- density distribution around single points are preserved by the objective



$$p_{j|i} = \frac{exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k\neq i} exp(-||x_i - x_k||^2/2\sigma_i^2)}$$

T - SNE

- Map point from High Dimensional space (x)to low dimensional space (y) preserving points distributions
- density distribution around single points are preserved by the objective



$$p_{j|i} = \frac{exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-||x_i - x_k||^2/2\sigma_i^2)}$$

Stochastic neighbour embedding

- Similarity of datapoint is converted to probabilities

$$p_{j|i} = \frac{exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-||x_i - x_k||^2/2\sigma_i^2)}$$

- Similarity in the low dimensional space y

$$q_{j|i} = \frac{exp(-||y_i - y_j||^2)}{\sum_{k \neq i} exp(-||y_i - y_k||^2)}$$

- OBJECTIVE:

Make the two distributions be as close as possible

- Minimize the Kullback Liebler Divergence

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} log \frac{p_{j|i}}{q_{j|i}}$$

Gradient descent solution

- Solve the problem pointwise by taking gradient of C w.r.t. points i=1…n

$$\frac{\partial C}{\partial y_i} = \sum_{j \neq i} (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(y_i - y_j)$$

1. Start with random y_i with i=1…n with n number of points
2. Move y_i using gradient step update
3. DO it for all points in Y using momentum

$$Y^{(t)} = Y^{(t-1)} + \eta \frac{\partial C}{\partial y_i} + \beta(t)(Y^{(t-1)} - Y^{(t-2)})$$

Gradient proof here:

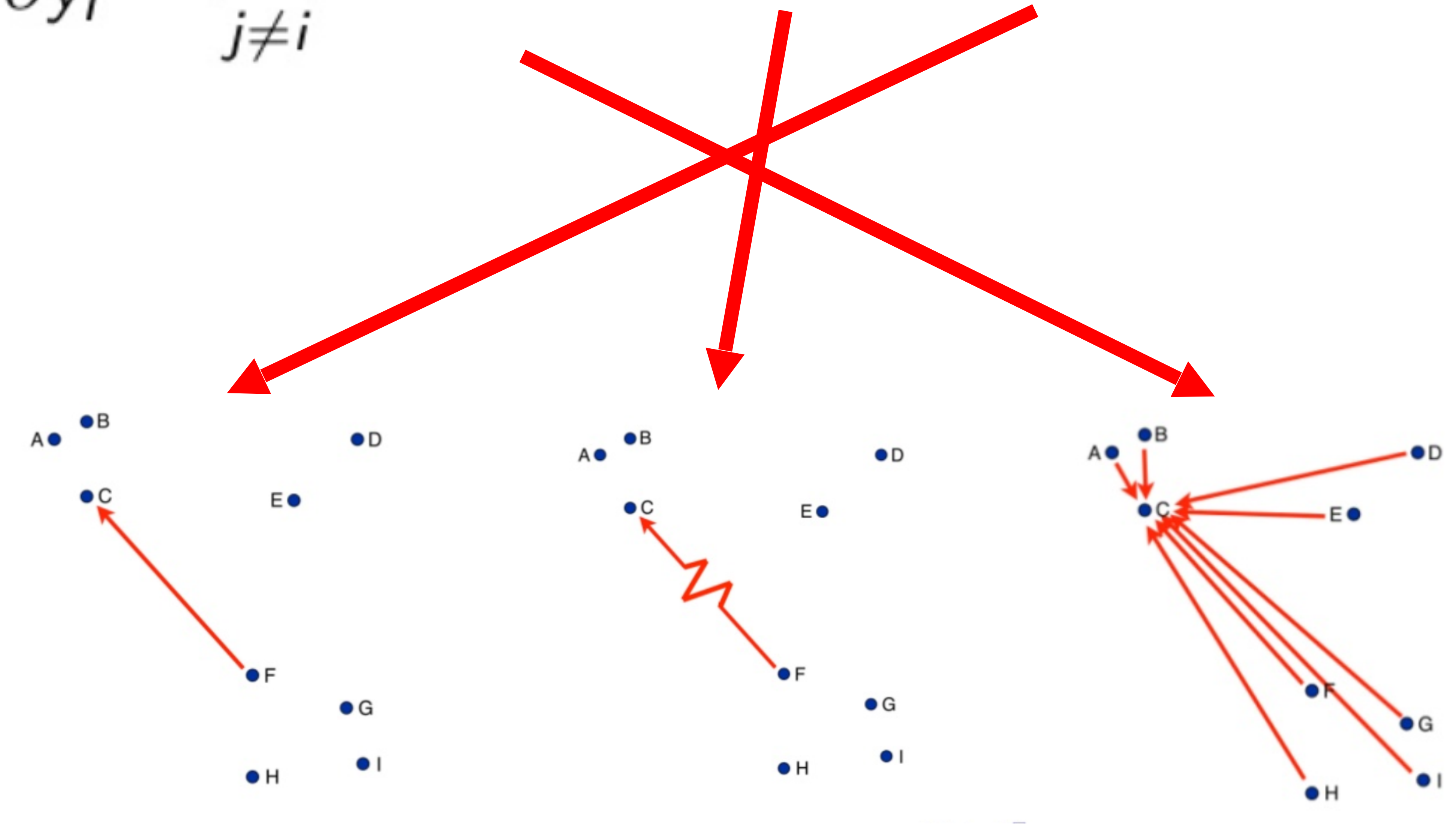
L.J.P. van der Maaten and G.E. Hinton. Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research 9

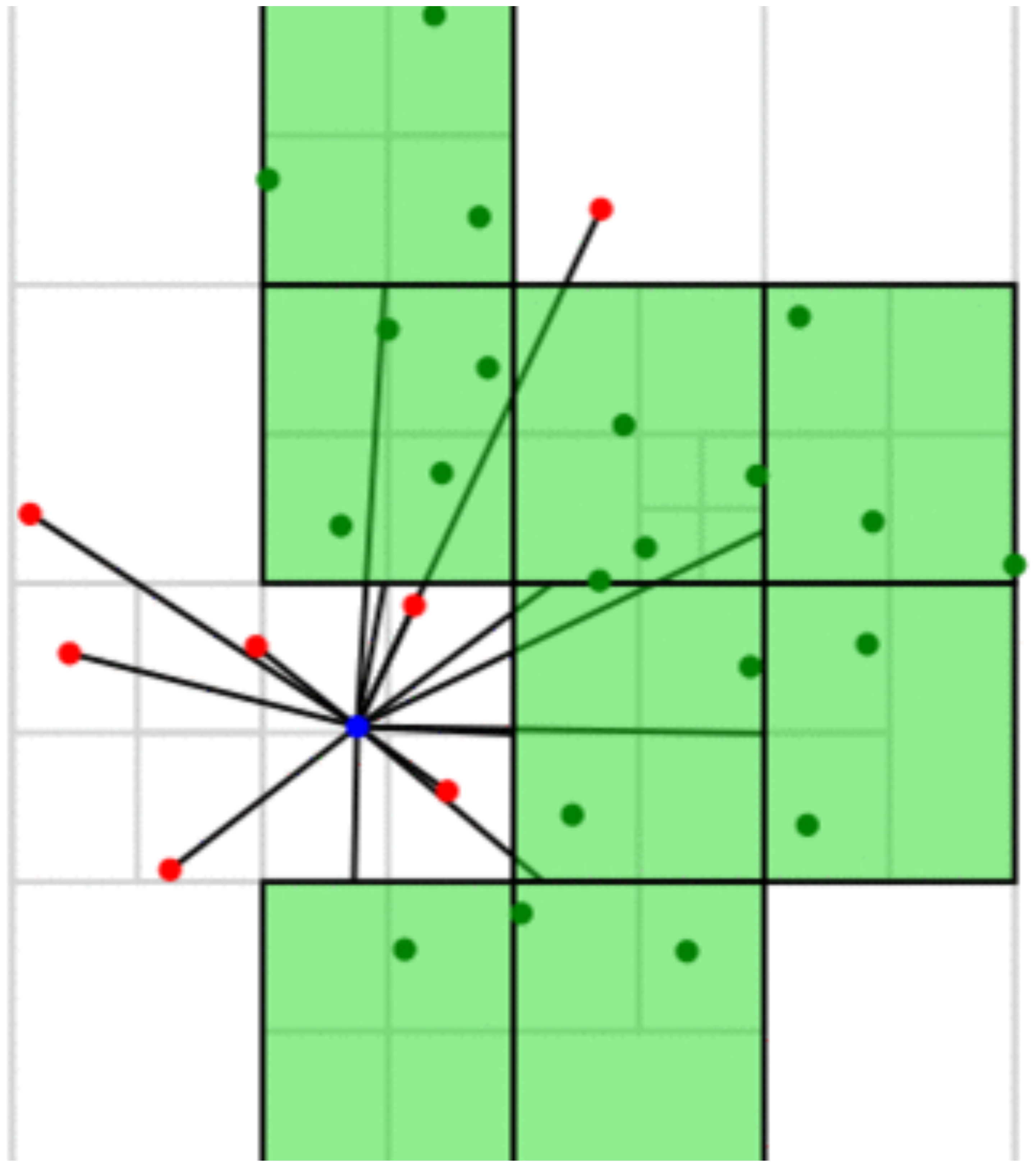
https://lvdmaaten.github.io/publications/papers/JMLR_2008.pdf

Gradient Interpretation

- Similar to N body problem

$$\frac{\partial C}{\partial y_i} = \sum_{j \neq i} (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(y_i - y_j)$$

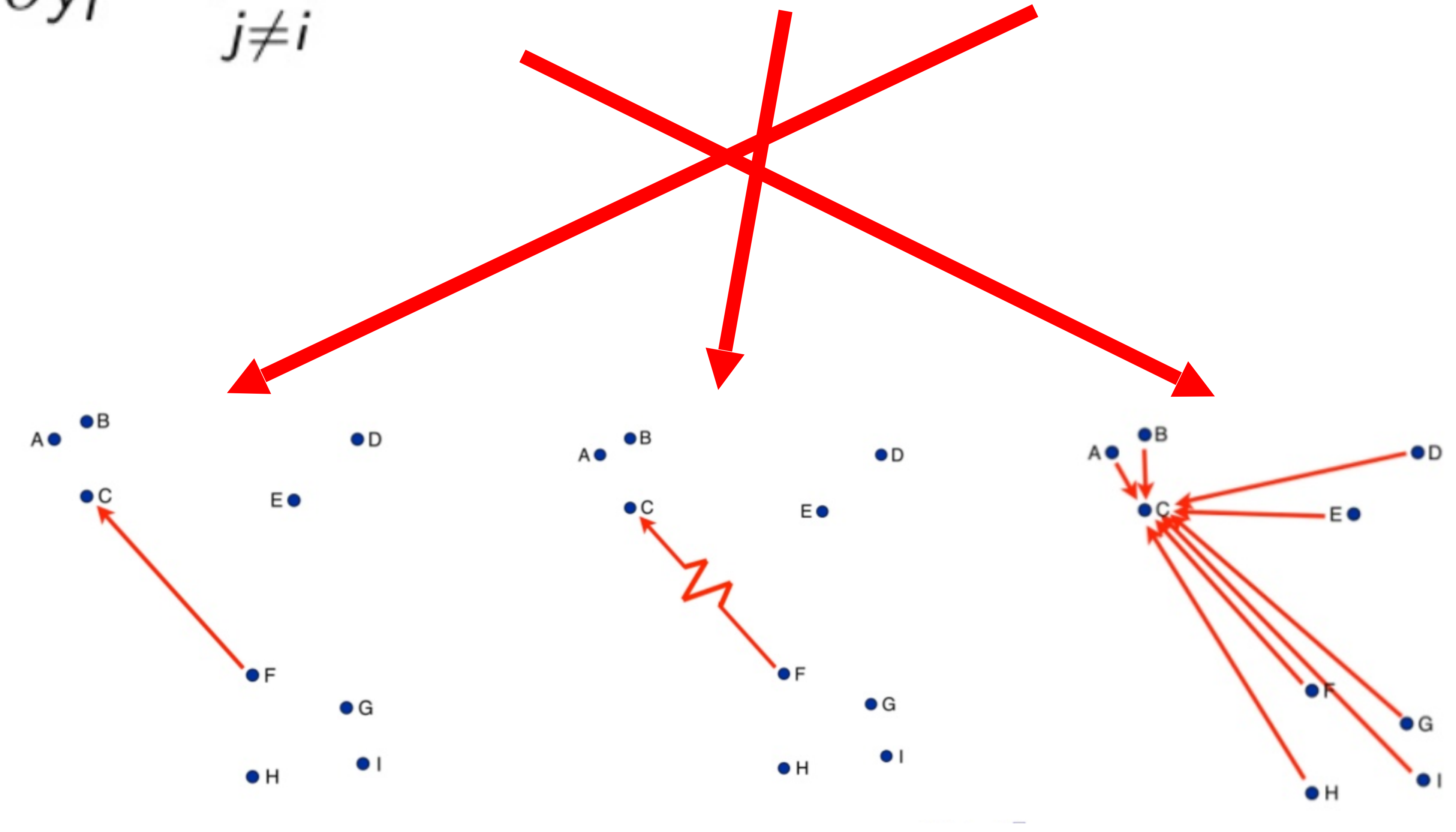


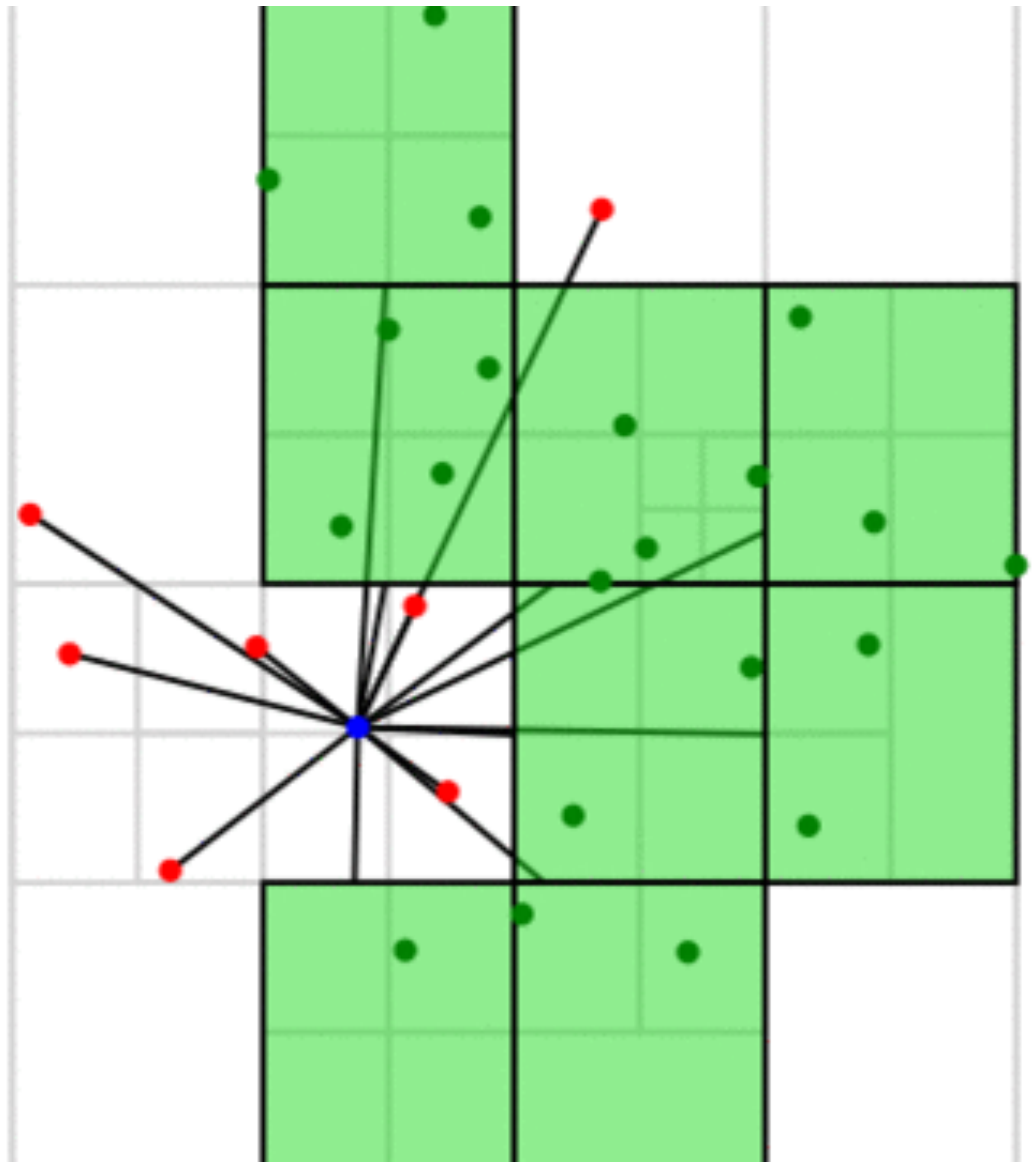


Gradient Interpretation

- Similar to N body problem

$$\frac{\partial C}{\partial y_i} = \sum_{j \neq i} (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(y_i - y_j)$$





Example Netflix movies

- More examples can be found here https://lvdmaaten.github.io/tsne/

T-sne code and additional resources

- T-SNE is the most popular embedding visualization method now.
- It is in most of the ML packages
- Inside SCIKIT LEARN
- Code and implementation for different languages here https://lvdmaaten.github.io/tsne/
- Sigma is crucial a good example on how sigma affect mapping https://distill.pub/2016/misread-tsne/
- Different TSNE variants : Symmetric, BH , Random Tree based