12. More on Dim. Reduction

EECE454 Introduction to Machine Learning Systems

2023 Fall, Jaeho Lee

Recap: PCA as a variance maximization

- **PCA.** Projecting the data to a plane spanned by principal components

eigenvectors for largest eigenvalues
of the data covariance matrix

  - Derived as a solution of variance maximization.

$$\max_{\mathsf{U}} \mathrm{Var}\Big(\{\pi_{\mathsf{U}}(\mathbf{x}_i)\}_{i=1}^{n}\Big)$$

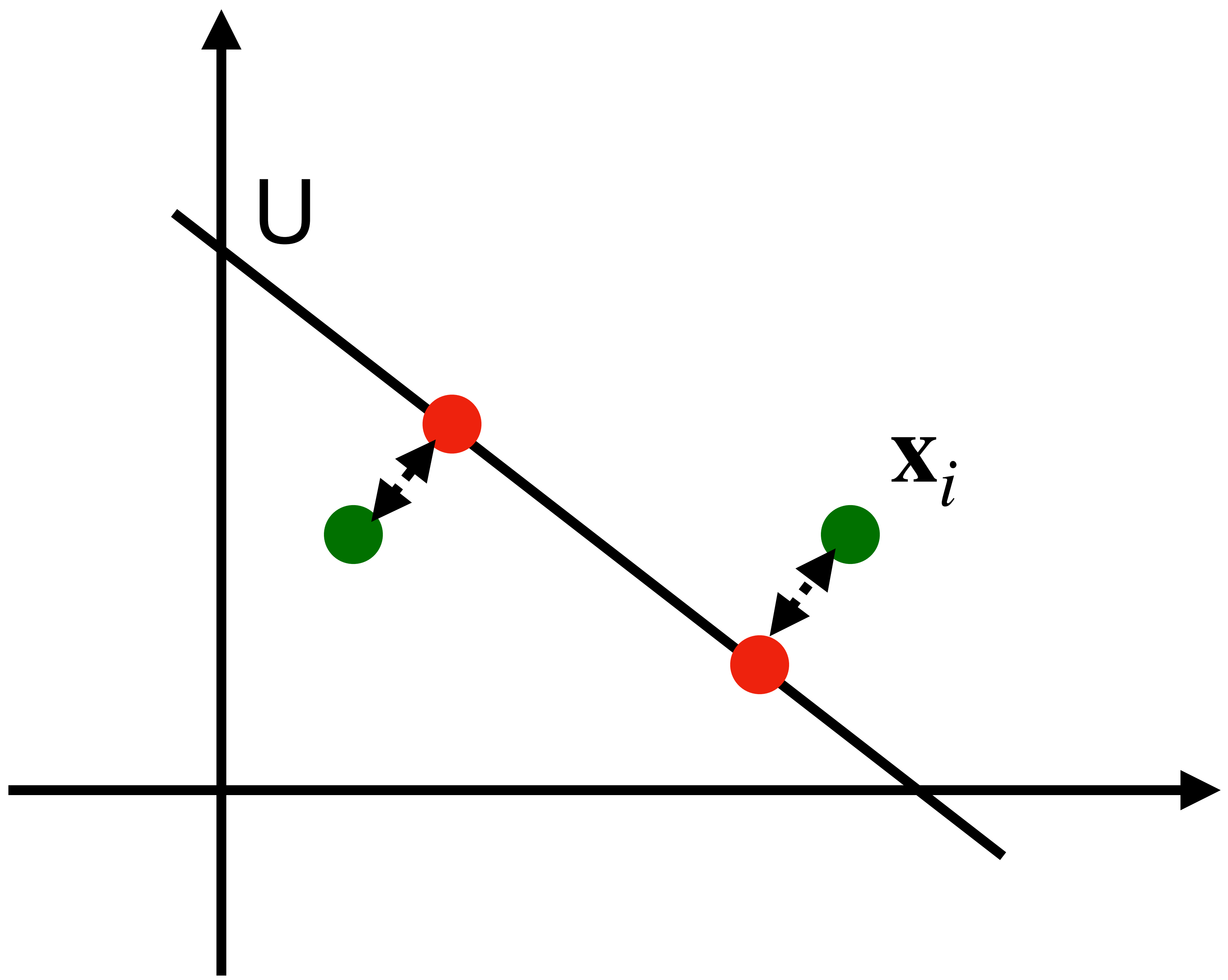
projection of $\mathbf{x}_i$ on the
affine subspace U

PCA as Distortion Minimization

Distortion Minimization

- Here's a perspective:

“If the projected point is close to the original point,
maybe it did not lose too much of original information.”



Distortion Minimization

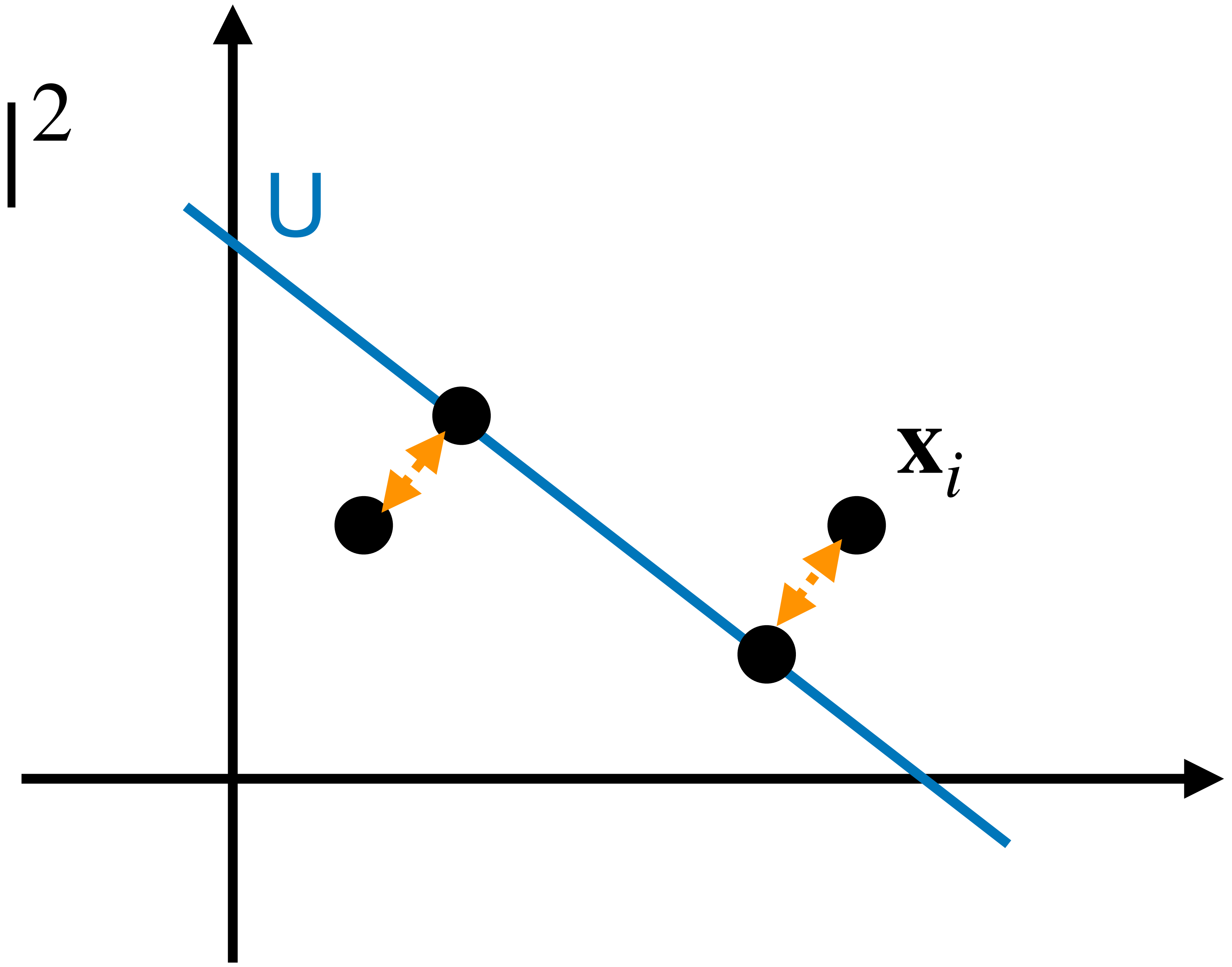
- Suppose that we try to find an affine subspace

$$\mathsf{U} = \{a_1\mathbf{u}_1 + \cdots + a_k\mathbf{u}_k + \mathbf{b} \ : \ a_i \in \mathbb{R}\}$$

such that the **mean of squared distortion** of each datum is minimized:

$$\min_{\mathsf{U}} \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}_i - \pi_{\mathsf{U}}(\mathbf{x}_i)\|^2$$

(distortion ≈ reconstruction error)



Distortion Minimization

- Suppose that we try to find an affine subspace

$$\mathsf{U} = \{a_1\mathbf{u}_1 + \cdots + a_k\mathbf{u}_k + \mathbf{b} \ : \ a_i \in \mathbb{R}\}$$

such that the **mean of squared distortion** of each datum is minimized:

$$\min_{\mathsf{U}} \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{x}_i - \pi_{\mathsf{U}}(\mathbf{x}_i)\|^2$$

- Using the definition of projection from last class, this is:

$$\min_{\mathbf{U},\mathbf{b}} \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{U}\mathbf{x}_i - \mathbf{b}\|^2$$

$$\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i - \mathbf{U}\mathbf{x}_i - \mathbf{b}\|^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(\|\mathbf{x}_i\|^2 + \|\mathbf{b}\|^2 - \mathbf{x}_i^\top \mathbf{U}\mathbf{x}_i - 2\mathbf{b}^\top \mathbf{x}_i + 2\mathbf{b}^\top \mathbf{U}\mathbf{x}_i\right)$$

$$= \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i\|^2 + \|\mathbf{b}\|^2 - \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^\top \mathbf{U}\mathbf{x}_i - 2\mathbf{b}^\top \bar{\mathbf{x}} + 2\mathbf{b}^\top \mathbf{U}\bar{\mathbf{x}}$$

- That is, we are solving

$$\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i\|^2 + \min_{\mathbf{U},\mathbf{b}}\left(\|\mathbf{b}\|^2 - \frac{1}{n}\sum \mathbf{x}_i^\top \mathbf{U}\mathbf{x}_i - 2\mathbf{b}^\top \bar{\mathbf{x}} + 2\mathbf{b}^\top \mathbf{U}\bar{\mathbf{x}}\right)$$

Optimizing b

$$\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i\|^2 + \min_{\mathbf{U},\mathbf{b}}\left(\|\mathbf{b}\|^2 - \frac{1}{n}\sum \mathbf{x}_i^\top \mathbf{U}\mathbf{x}_i - 2\mathbf{b}^\top\bar{\mathbf{x}} + 2\mathbf{b}^\top\mathbf{U}\bar{\mathbf{x}}\right)$$

- Minimizing with respect to $\mathbf{b}$, we get:

$$\mathbf{b} = \bar{\mathbf{x}} - \mathbf{U}\bar{\mathbf{x}}$$

- Plug in to get:

$$\left(\underbrace{\frac{1}{n}\sum\|\mathbf{x}_i\|^2 - \bar{\mathbf{x}}^\top\bar{\mathbf{x}}}_{=\mathrm{Var}\left(\{\mathbf{x}_i\}_{i=1}^{n}\right)}\right) + \min_{\mathbf{U}}\left(\underbrace{\bar{\mathbf{x}}^\top\mathbf{U}\bar{\mathbf{x}} - \frac{1}{n}\sum\mathbf{x}_i^\top\mathbf{U}\mathbf{x}_i}_{=-\sum_{j=1}^{k}\mathbf{u}_j^\top\mathbf{S}\mathbf{u}_j}\right)$$

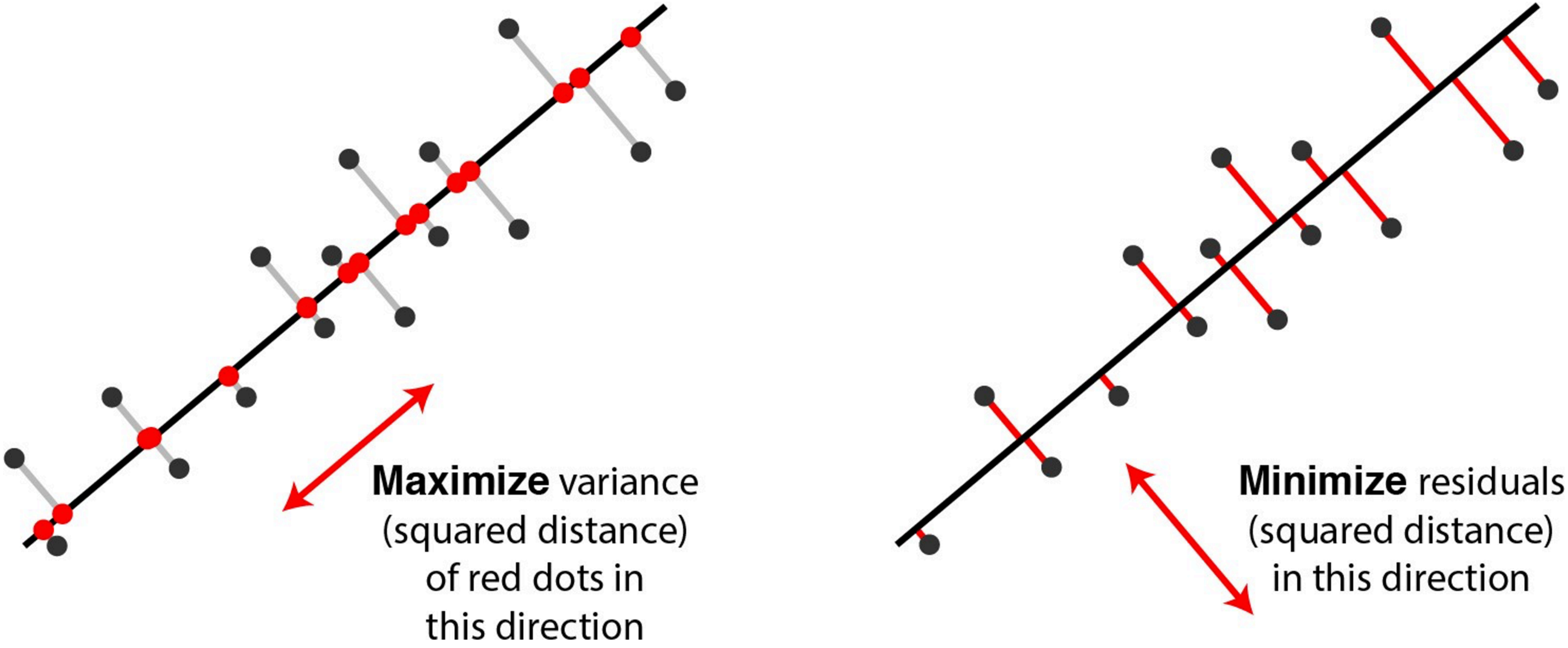
The equivalence

- Summing up, we have

$$\min_{\mathsf{U}} \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{x}_i - \pi_{\mathsf{U}}(\mathbf{x}_i)\|^2 = \mathrm{Var}(\{\mathbf{x}_i\}) - \max_{\mathbf{U}} \left(\sum_{j=1}^{k} \mathbf{u}_j \mathbf{S} \mathbf{u}_j \right)$$

exactly what we solved for
variance maximization problem

- **Difference.** The bias $\mathbf{b}$ is well-characterized in this case.



Maximize variance
(squared distance)
of red dots in
this direction
Minimize residuals
(squared distance)
in this direction

PCA in a nutshell

PCA as the best linear compression

- We project the data to a k-**dimensional affine subspace** in $\mathbb{R}^d$.
  - A datum $\mathbf{x} \in \mathbb{R}^d$ is projected to a k-dimensional **code**

$$\mathbf{z} = (a_1, \ldots, a_k), \qquad \text{where} \quad \mathbf{x} = \sum_{i=1}^{k} a_i \mathbf{e}_i$$

for some bases $\mathbf{e}_1, \ldots, \mathbf{e}_k$ of the subspace.

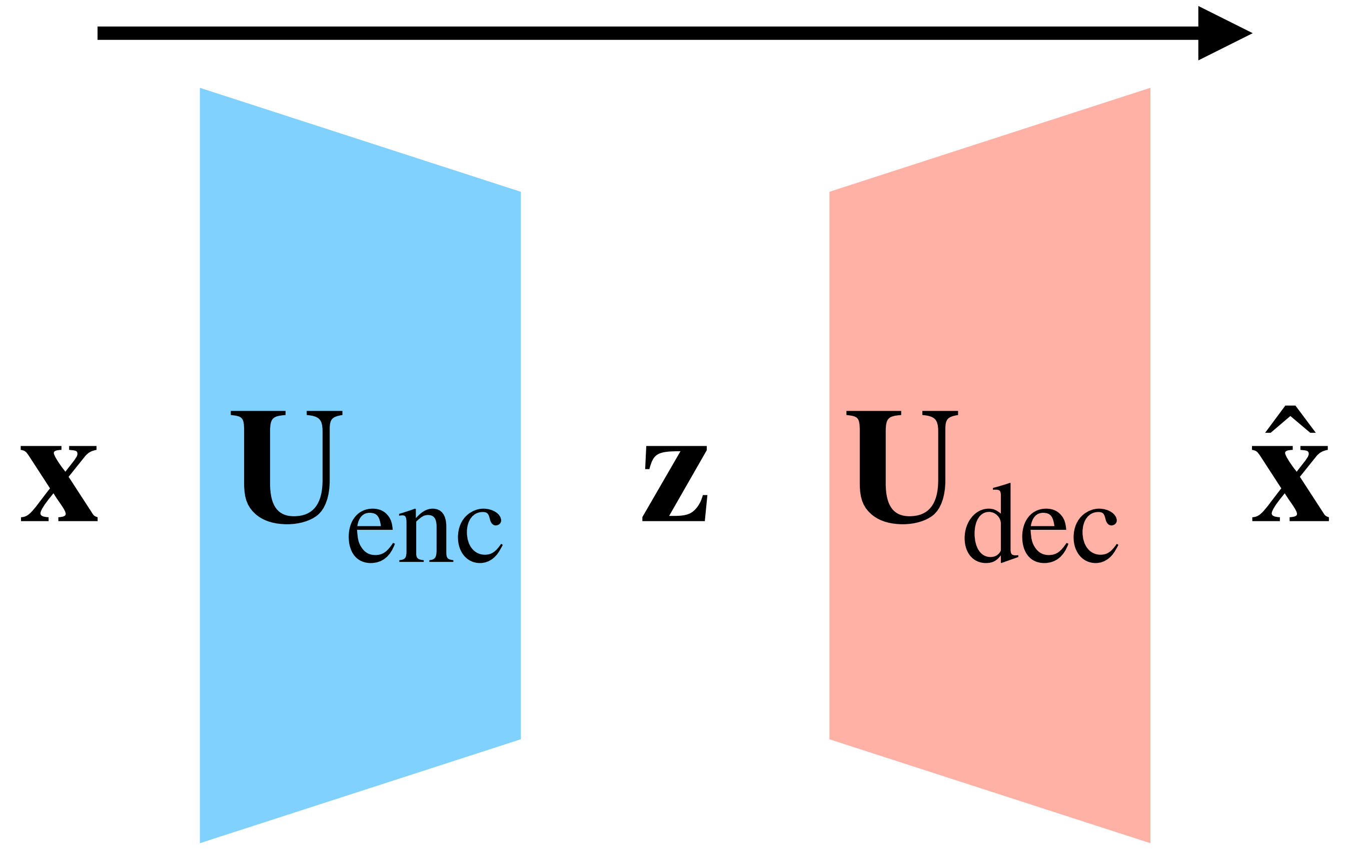
PCA as the best linear compression

- This linear **encoding** can be written as

$$\mathbf{z} = \mathbf{U}_{\text{enc}}\mathbf{x}, \qquad \text{where} \quad \mathbf{U}_{\text{enc}} = \begin{bmatrix} \leftarrow & \mathbf{e}_1^\top & \rightarrow \\ & \cdots & \\ \leftarrow & \mathbf{e}_k^\top & \rightarrow \end{bmatrix} \in \mathbb{R}^{k \times d}$$

- One can **decode** back the data using some linear matrix $\mathbf{U}_{\text{dec}}$:

$$\hat{\mathbf{x}} = \mathbf{U}_{\text{dec}}\mathbf{z}$$

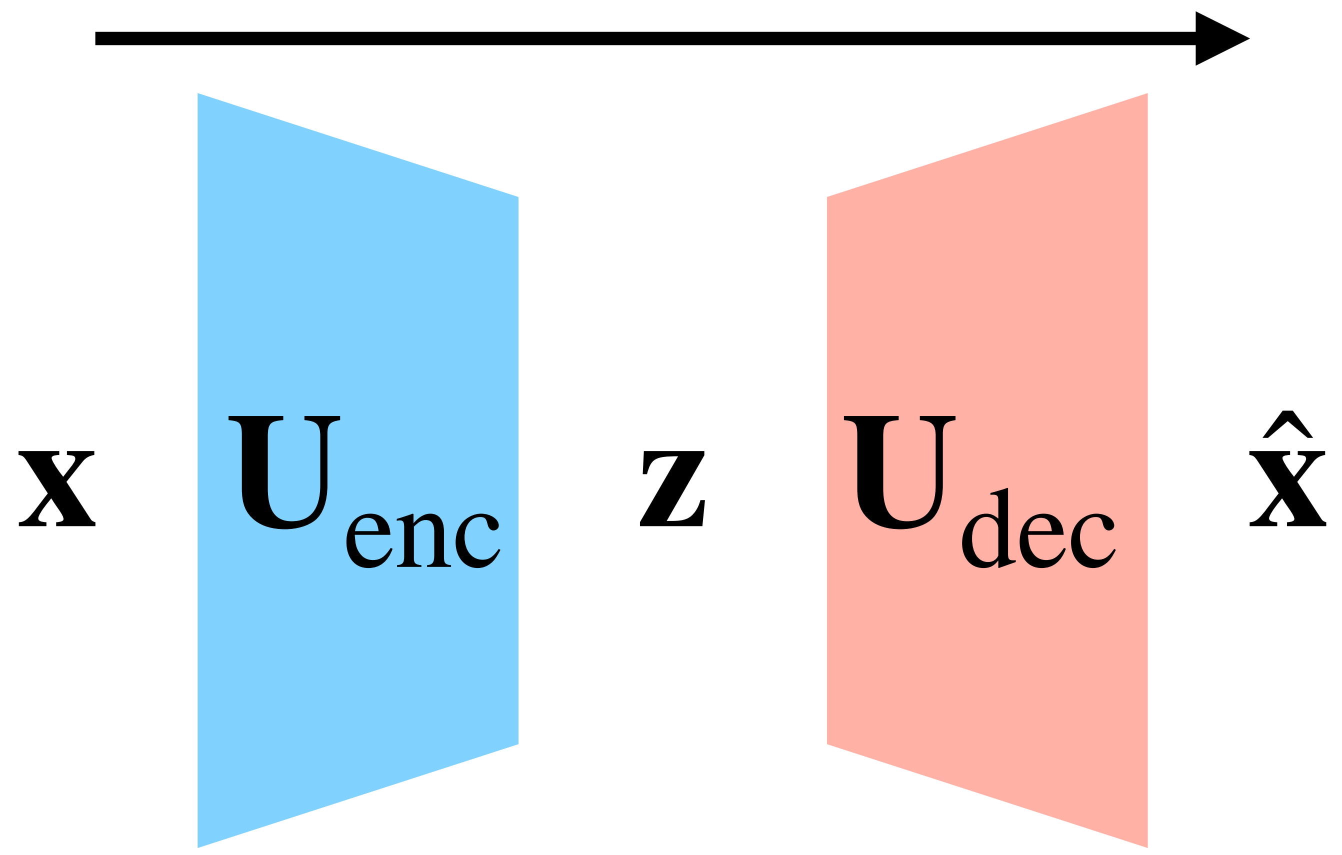


PCA as the best linear compression

- PCA solves the **reconstruction error minimization** problem

$$\min_{\mathbf{U}_{\mathrm{enc}},\mathbf{U}_{\mathrm{dec}}} \frac{1}{n}\sum_{i=1}^{n} \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2^2$$

  - Our mathematical derivations say that it is optimal to:
    - **Encode**. Use the top-k principal components $\mathbf{e}_1, \ldots, \mathbf{e}_k \in \mathbb{R}^d$ of data covariance matrix to construct $\mathbf{U}_{\mathrm{enc}}$
    - **Decode.** Use $\mathbf{U}_{\mathrm{dec}} = \mathbf{U}_{\mathrm{enc}}^\top$



Applications of PCA

Face Recognition

- An ancient example: Eigenface (1991).
  - We can identify important characteristics of faces.

    ⇒ Can be used for rapid recognition, tracking, and reconstruction

Original Dataset

Eigenvectors

Image Compression

- Image Compression
  - Divide each image to 12x12 pixel patches.
  - Save only low-dimensional values

Each patch is represented as

$$a_1\mathbf{u}_1 + \cdots + a_k\mathbf{u}_k$$

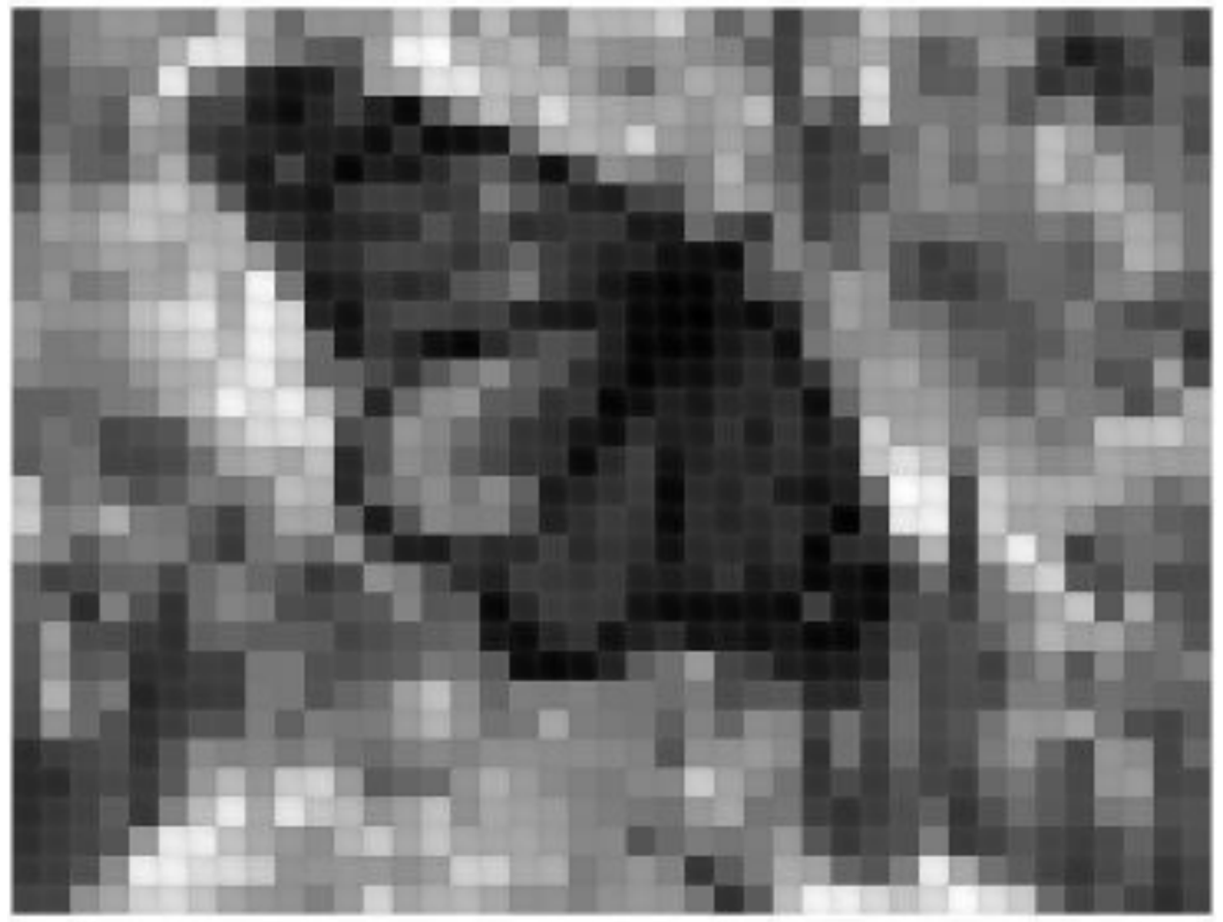
Save for each patch = $(a_1, \cdots, a_k)$

Common codebook = $(\mathbf{u}_1, \cdots, \mathbf{u}_k)$

144-dimension (full)

60-dimension

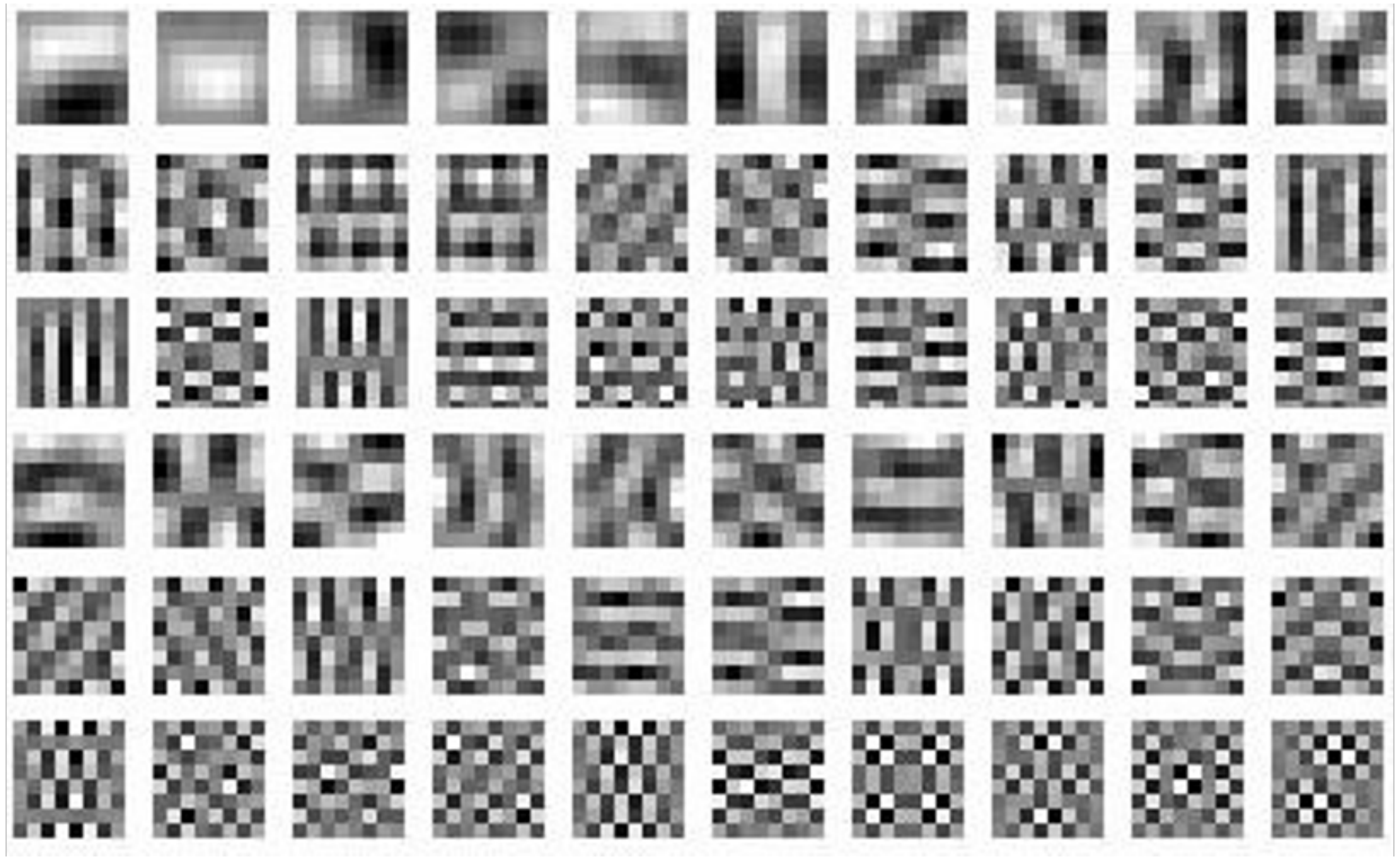
6-dimension



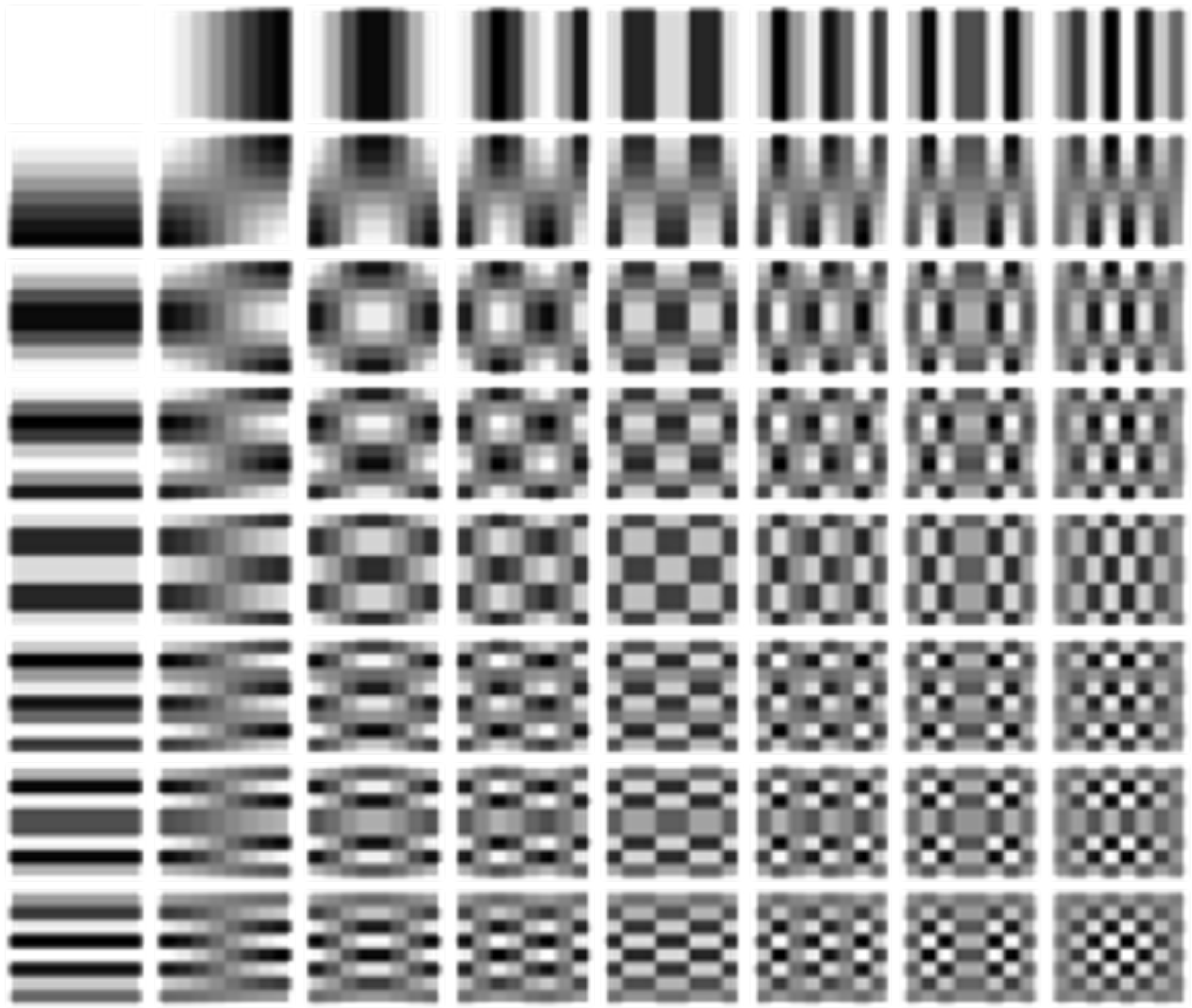
1-dimension

Image Compression

- The eigenvectors look similar to discrete cosine transforms (DCTs), which are used in JPEG



Eigenvectors



Discrete Cosine Bases

Noise Filtering

- Noises often contribute small to principal components, and thus can be removed by PCA

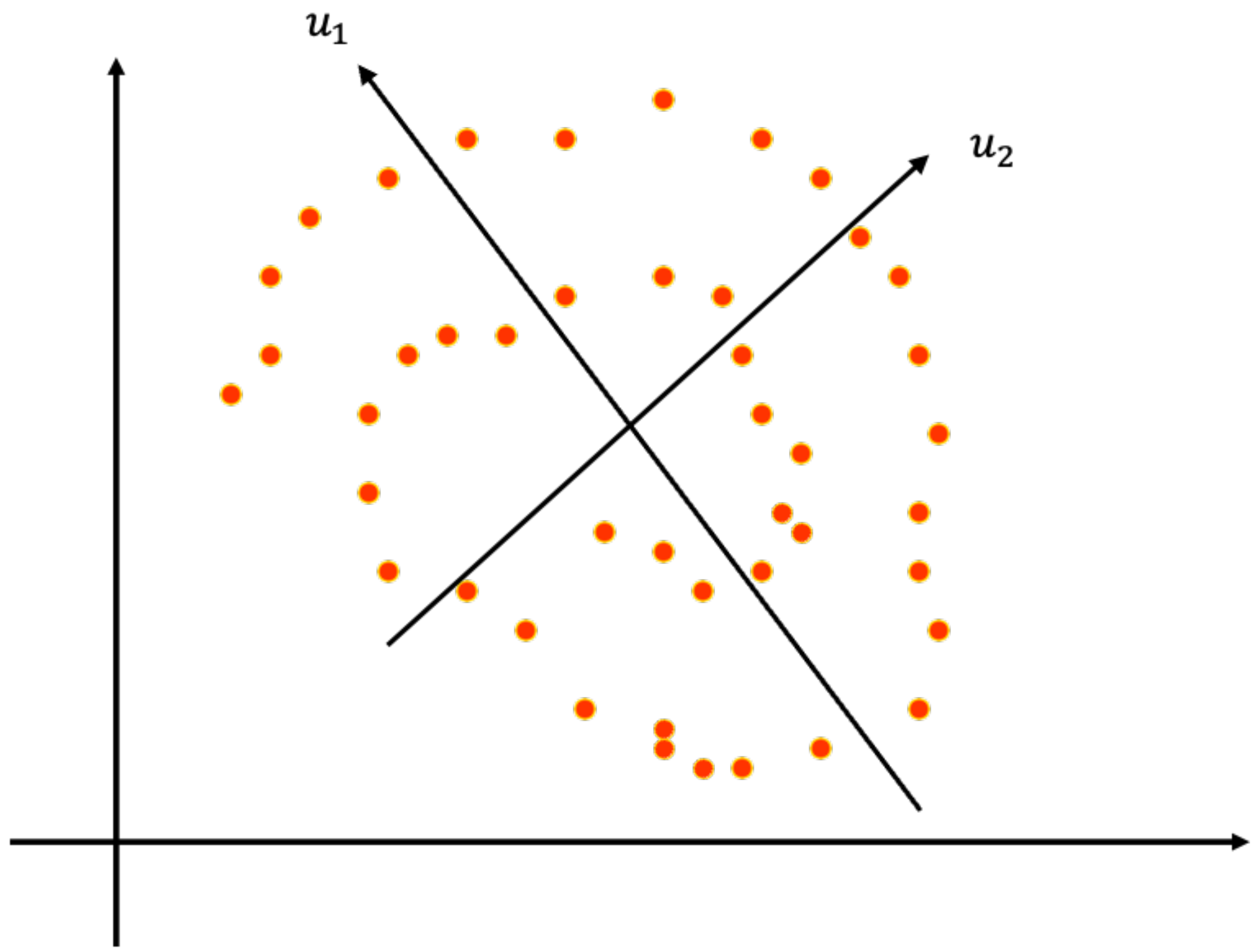
Noisy Image

15-dimension

Limitations of PCA

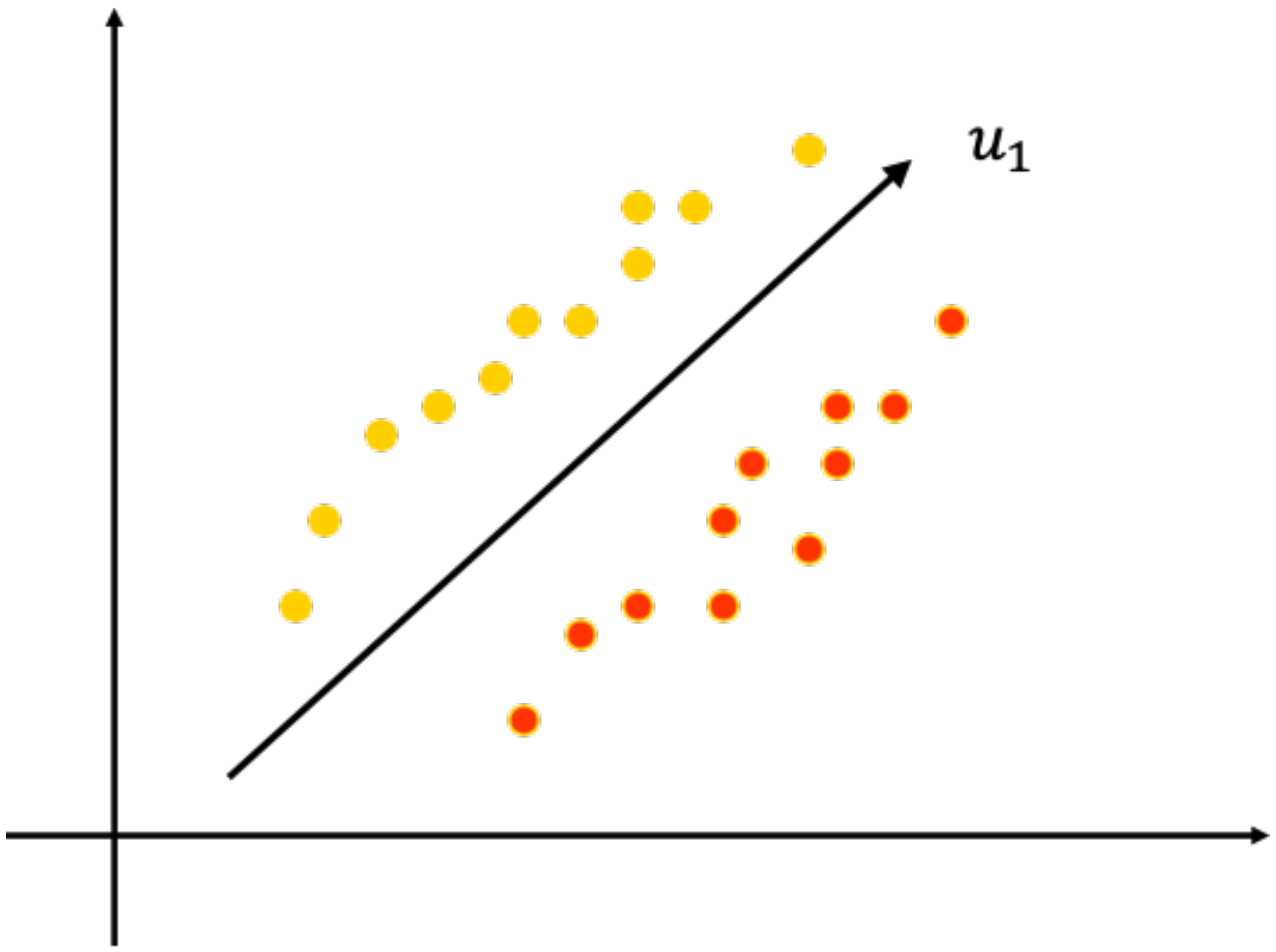
Failure Modes of PCA

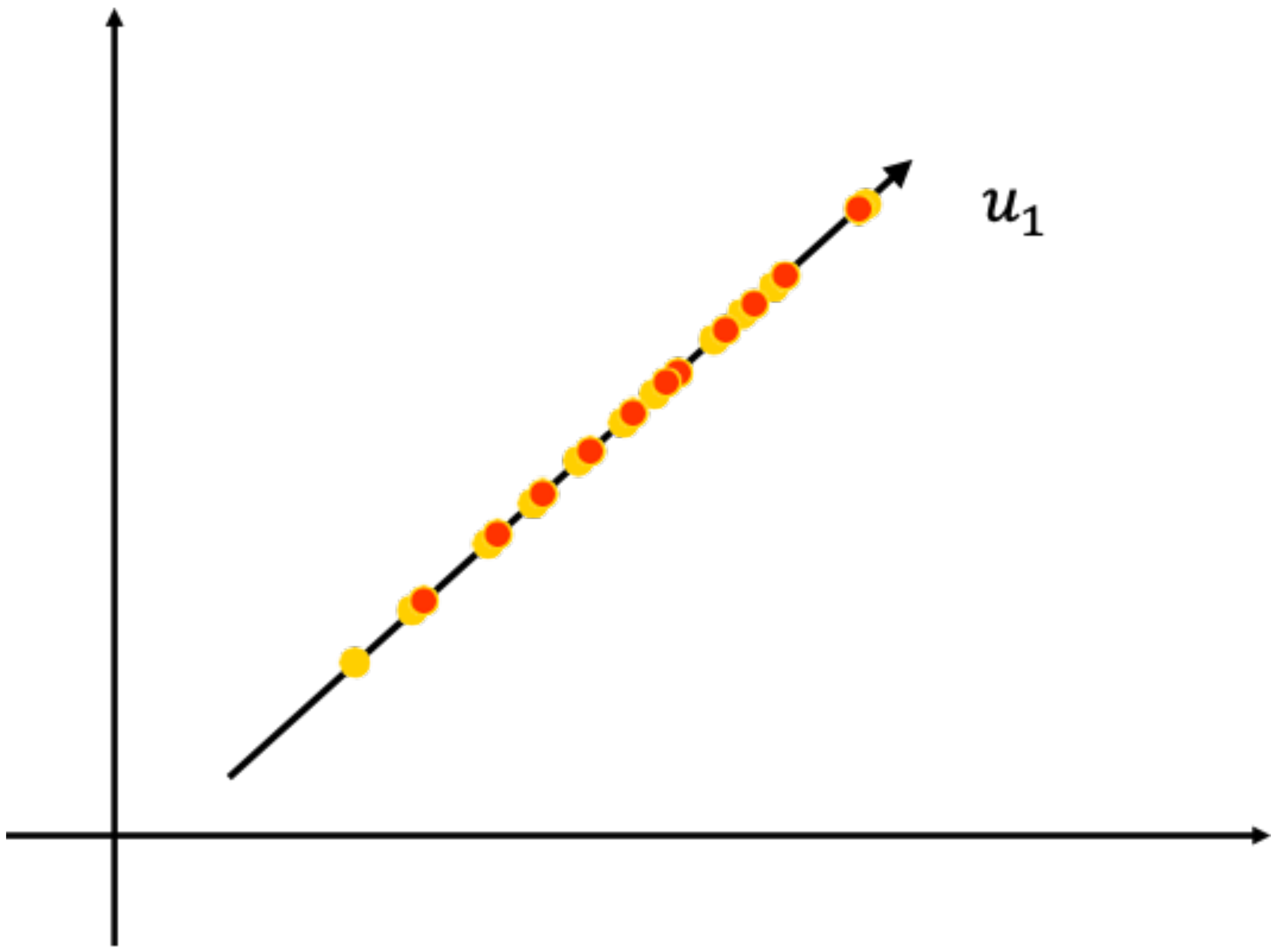
- Difficult to capture non-linear datasets



Failure Modes of PCA

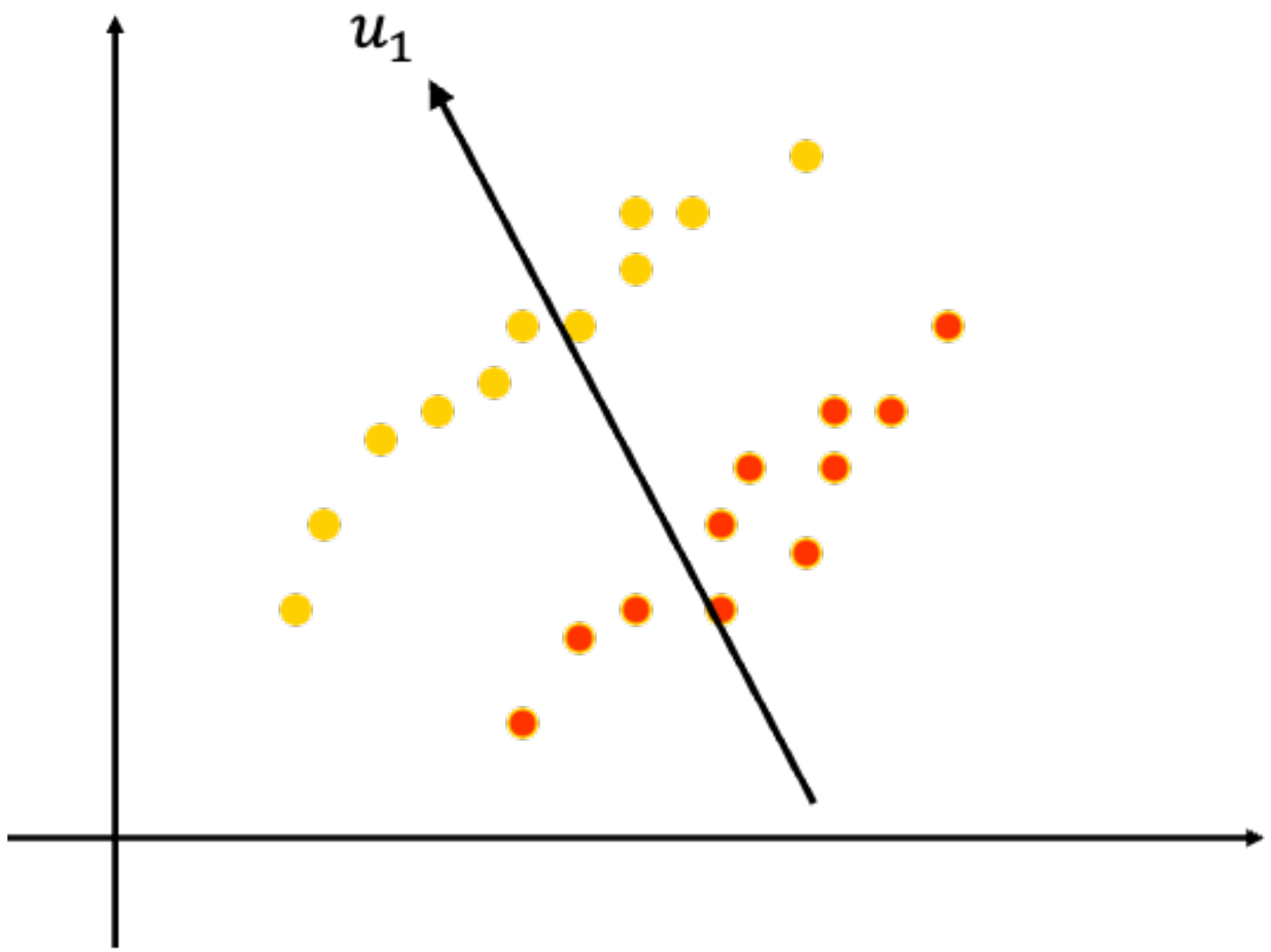
- PCA does not account for class labels





Failure Modes of PCA

- PCA does not account for class labels
    - If it could account for...

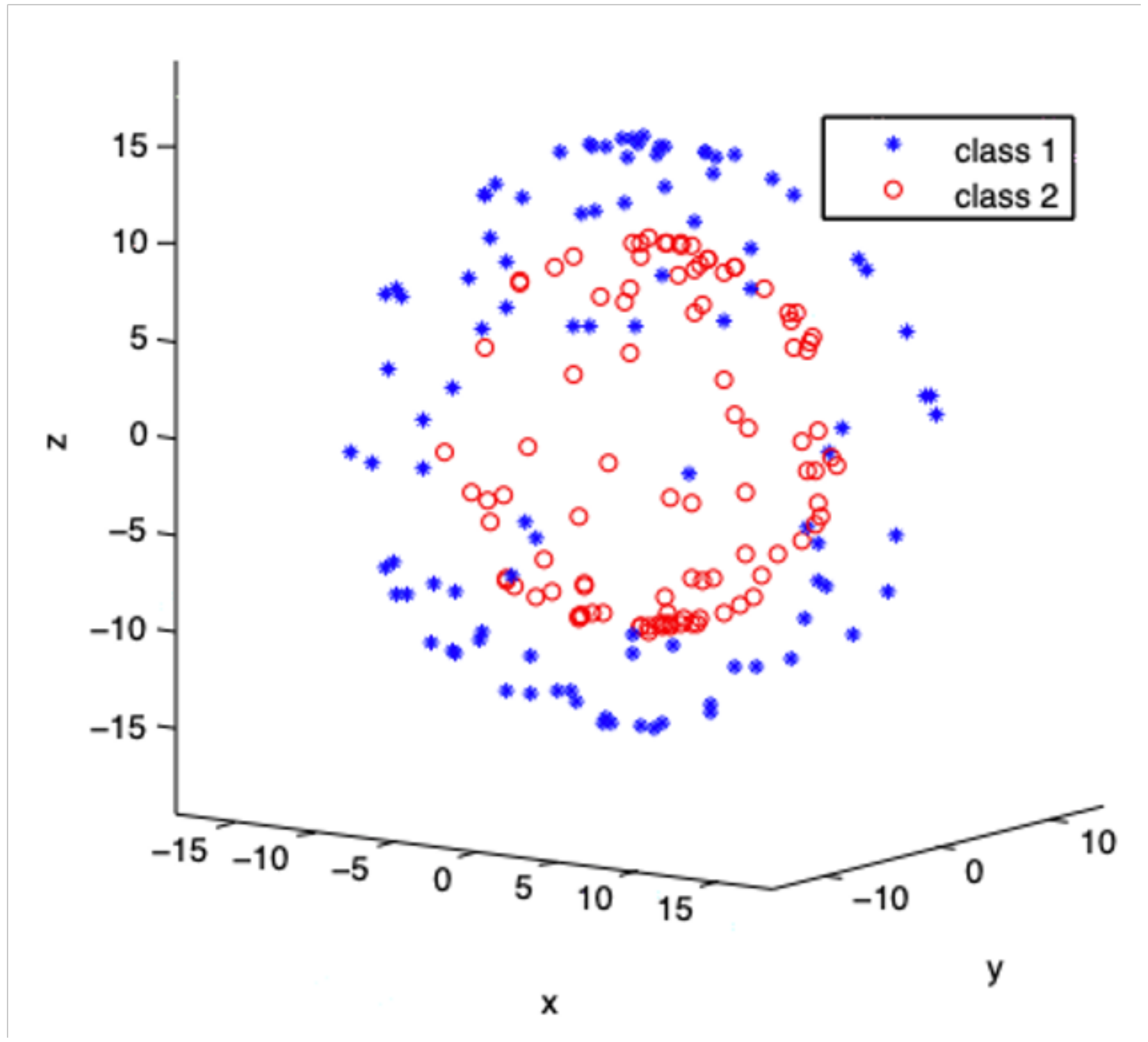


Advanced Methods

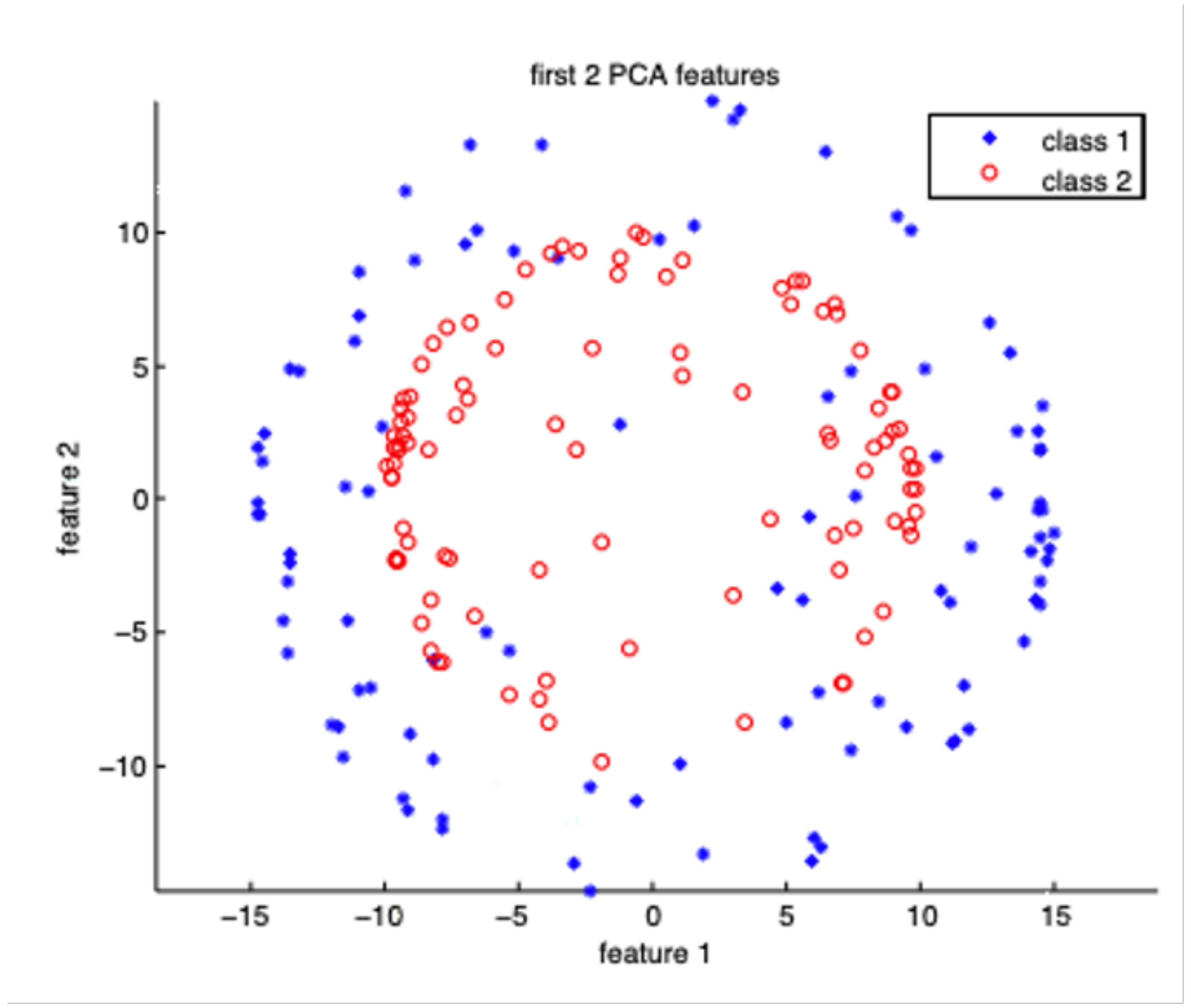
Kernel PCA

- **Idea.** Perform PCA for $\Phi(\mathbf{x})$, not $\mathbf{x}$

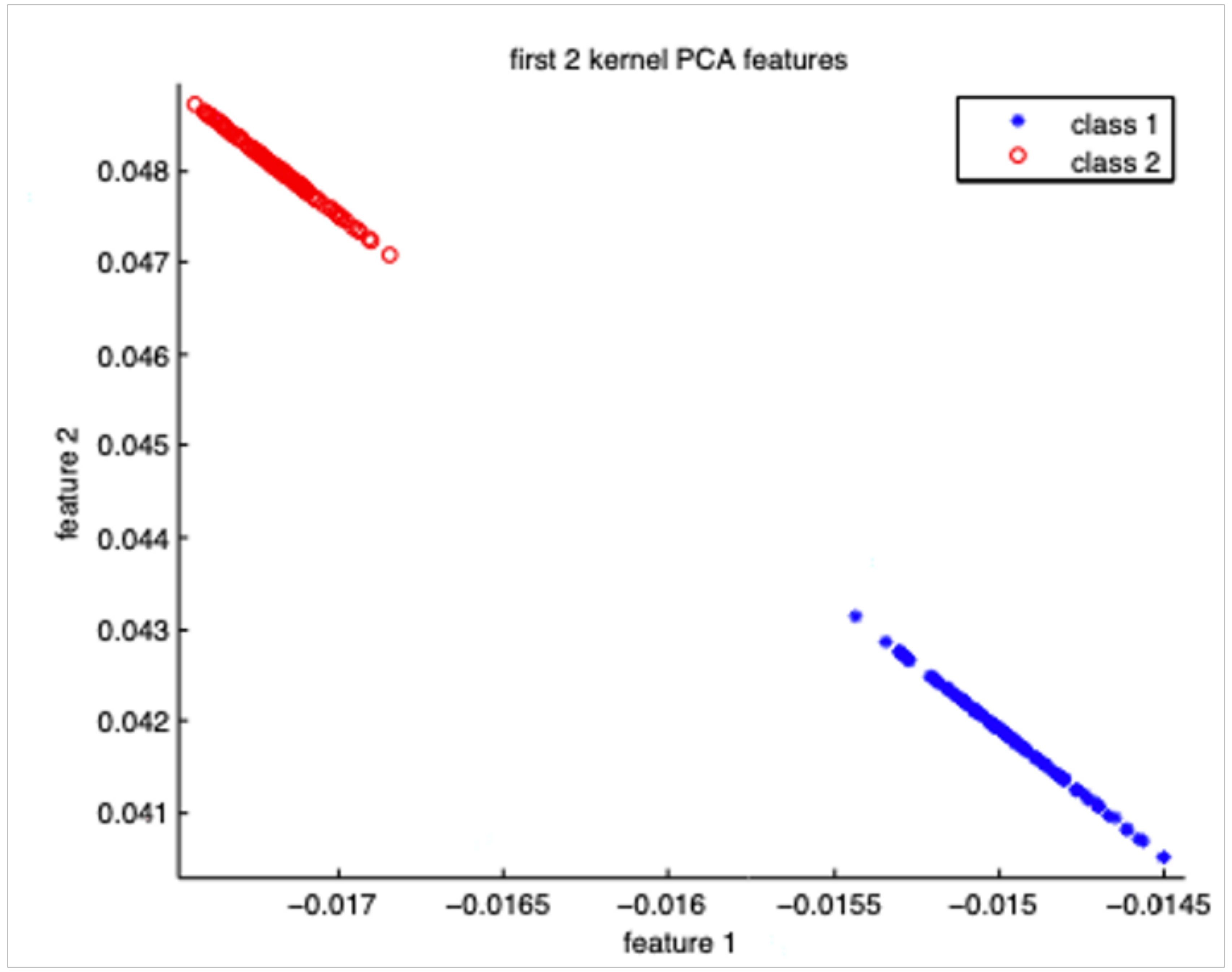
  (requires careful hyperparameter tuning & validation)



Spherical Data



No Kernel



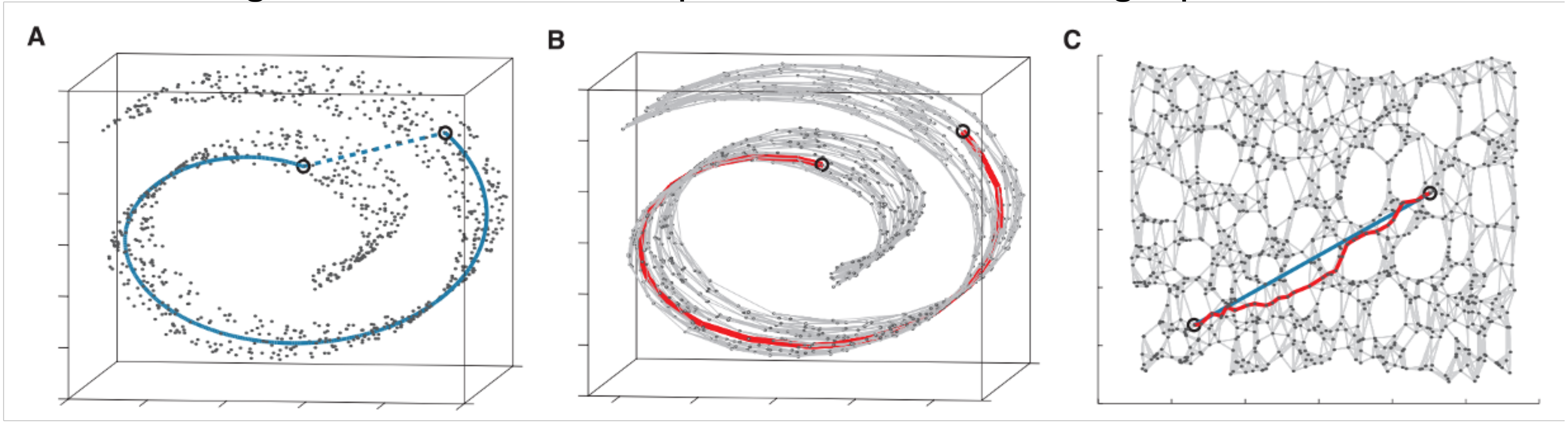
Gaussian Kernel ($\sigma = 20$)

Isomap (2000)

- Embed each data to low-dimensional space so that

  distance on the manifold = distance on the embedded space

- **Idea.** Build a graph of points by connecting each point to k-nearest neighbors $\Rightarrow$ Measure pairwise distance as graph distance.



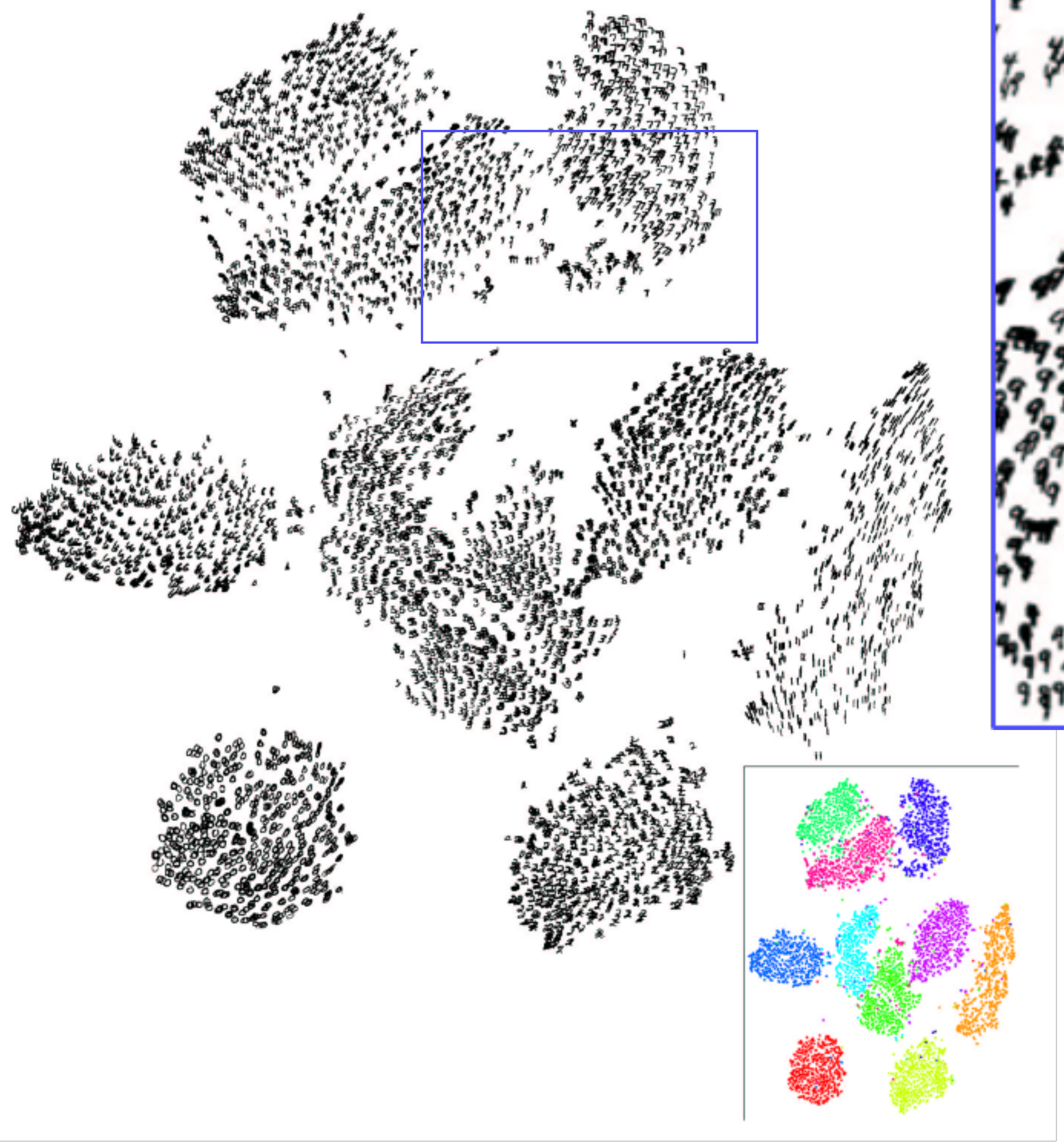
t-SNE

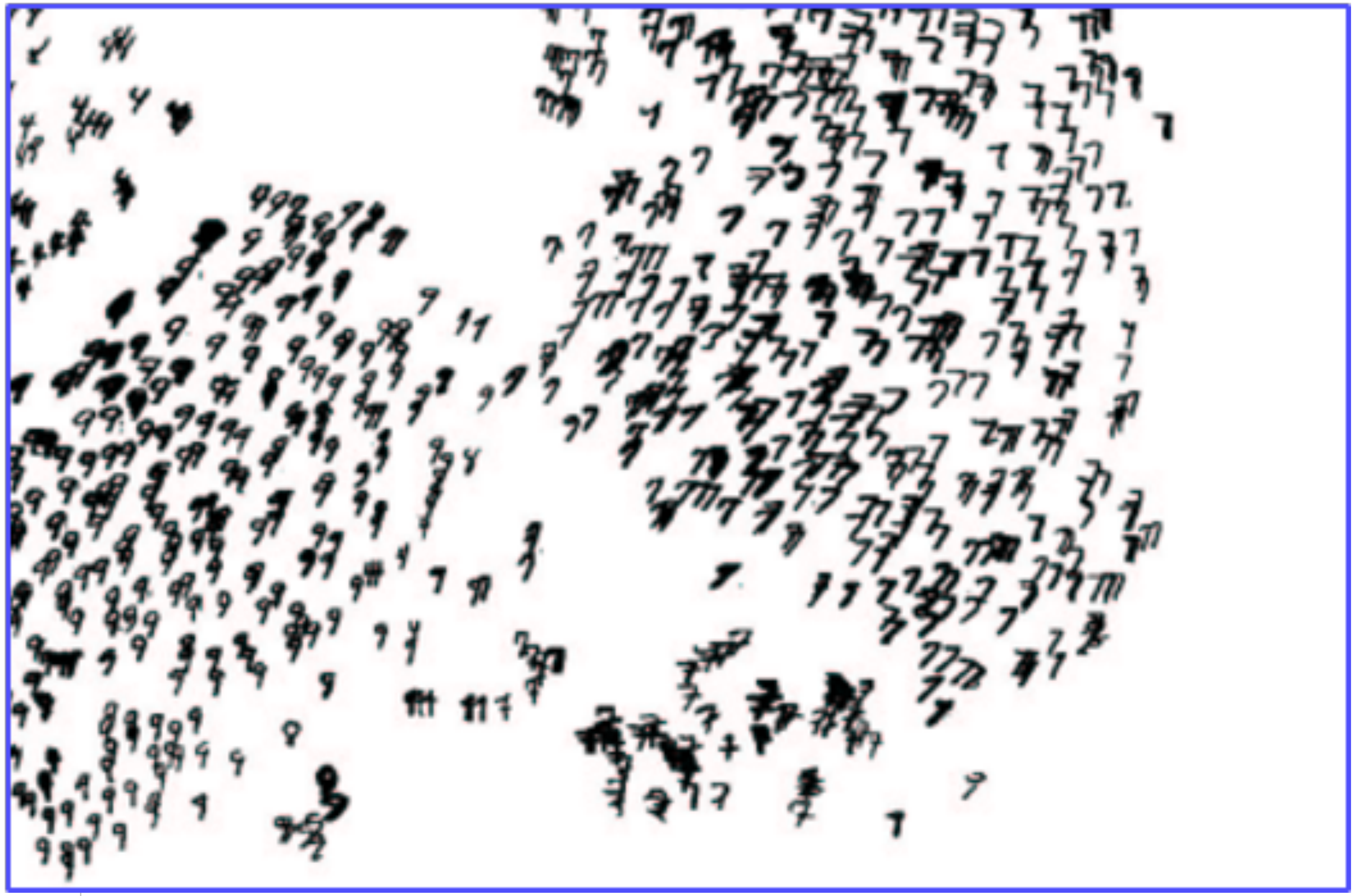
- Similar to Isomap, we preserve some distance.
  - **Difference.** Encode neighbor info. as a probability distribution.

$$p_i(j) = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma^2)}$$

Then, we find the low-dim embedding such that

$$\mathrm{dist}(p_i, p_j) \approx \mathrm{dist}(\mathbf{z}_i, \mathbf{z}_j)$$





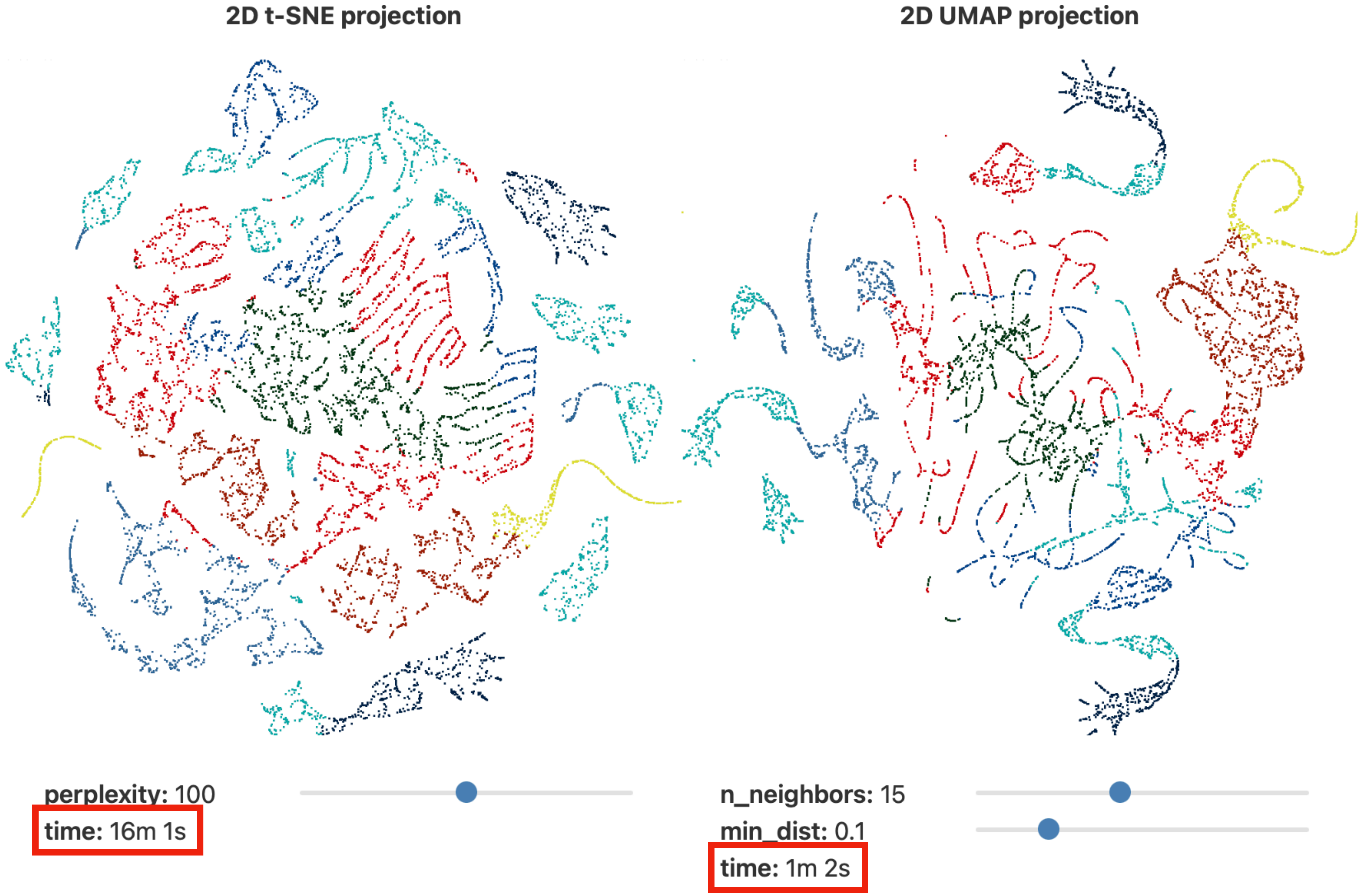
MNIST embeddings of t-SNE

(requires computing pairwise distances of 60,000 samples)


ICLR 2017–2024 submissions (n=24,347)
Coloured by year
2024
2017
Coloured by topic
offline reinf
multi-agent
imitation
reinforcement
causal
bandit
game
continual
physics
meta-learning
reasoning
language
question
recurrent
optimization
federated
vision-lang
autoencode
word
object-
privacy
bayesian
fair
attentio
audio
video
backdoor
network
3d
adversarial
diffusion
semi-sup
contrastive
generative adv
pruning
domain
time series
out-of-dis
protein
architecture
anomaly
distillation
graph
All ICLR submissions over eight years, scraped using the OpenReview API. Titles+abstracts in TF-IDF representation, SVD to 100 components, visualized with t-SNE. Papers are assigned labels based on specific words present in their titles.
Dmitry Kobak, @hippopedoid

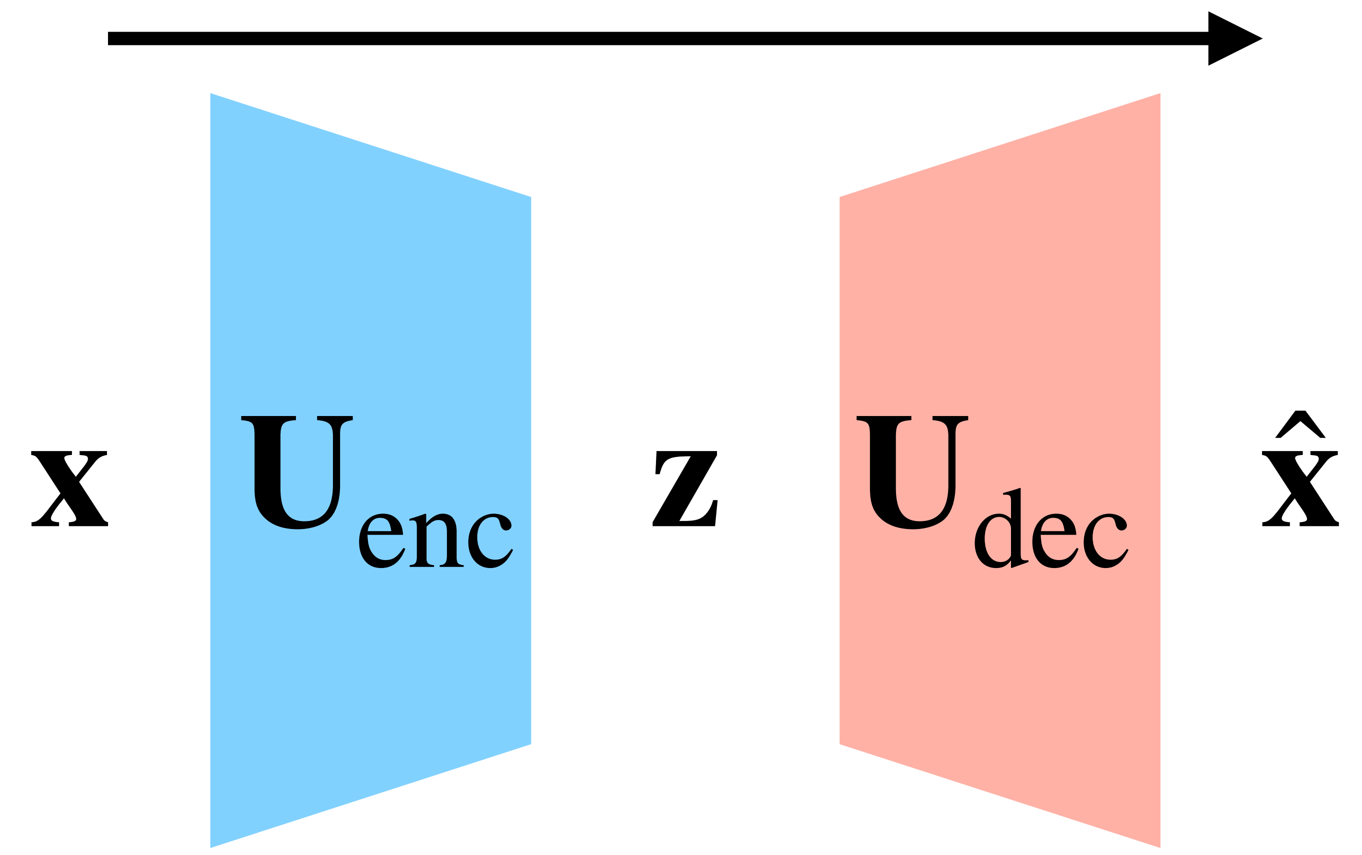
UMAP

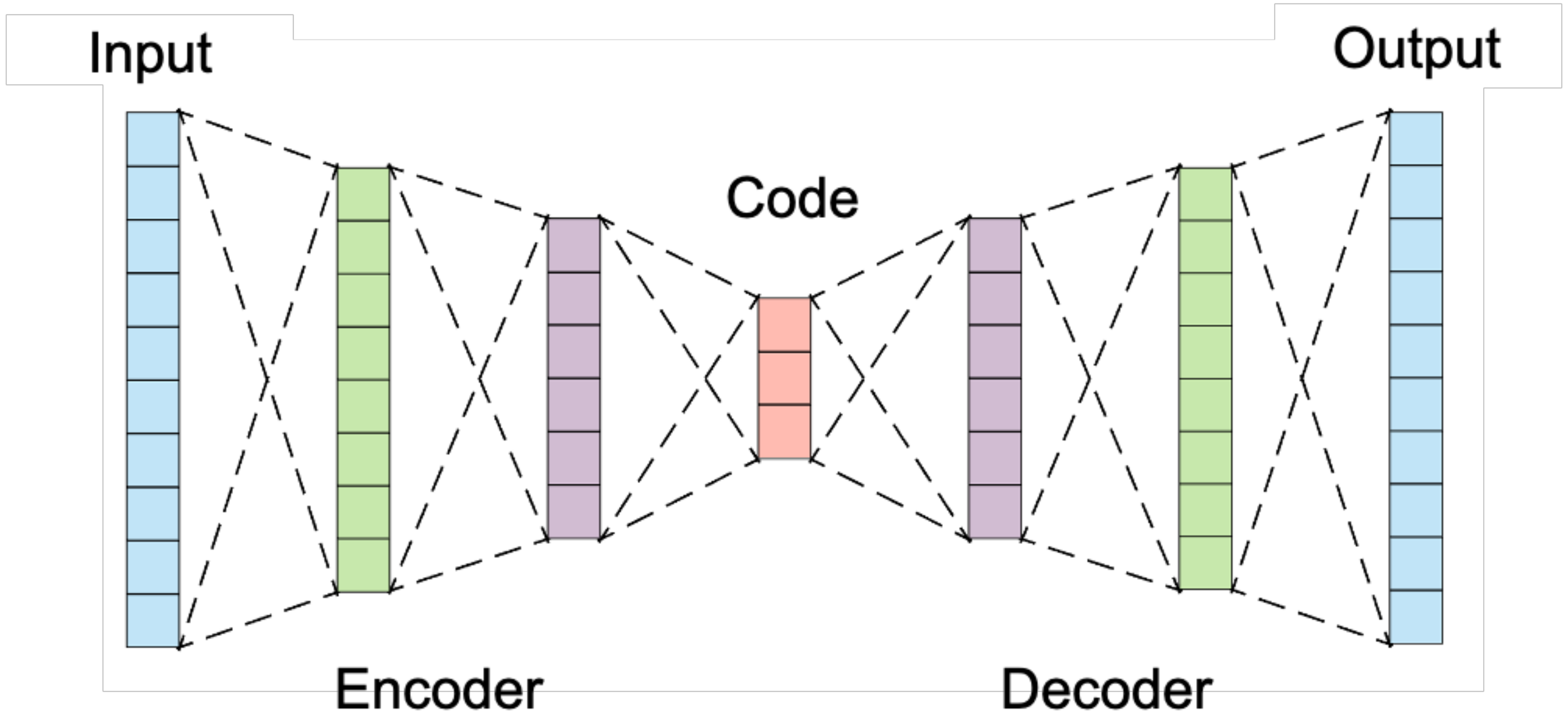
- An elaborate version of Isomap, but much faster!
  - Reference: https://pair-code.github.io/understanding-umap/



Autoencoders

- In PCA, we used linear matrices for encoding & decoding:
- Autoencoders do the same thing, but with neural nets:
  - Train nonlinear encoder & decoder with SGD.





Cheers

- *Next up.* Mid-term!