

---

# Dimensionality Reduction

## Part 1: PCA & NMF

Dalya Baron  
Carnegie Observatories

---

*Vatican Observatory Summer School on Big Data and  
Machine Learning 2023 (VOSS-2023)*

# Science is about compression



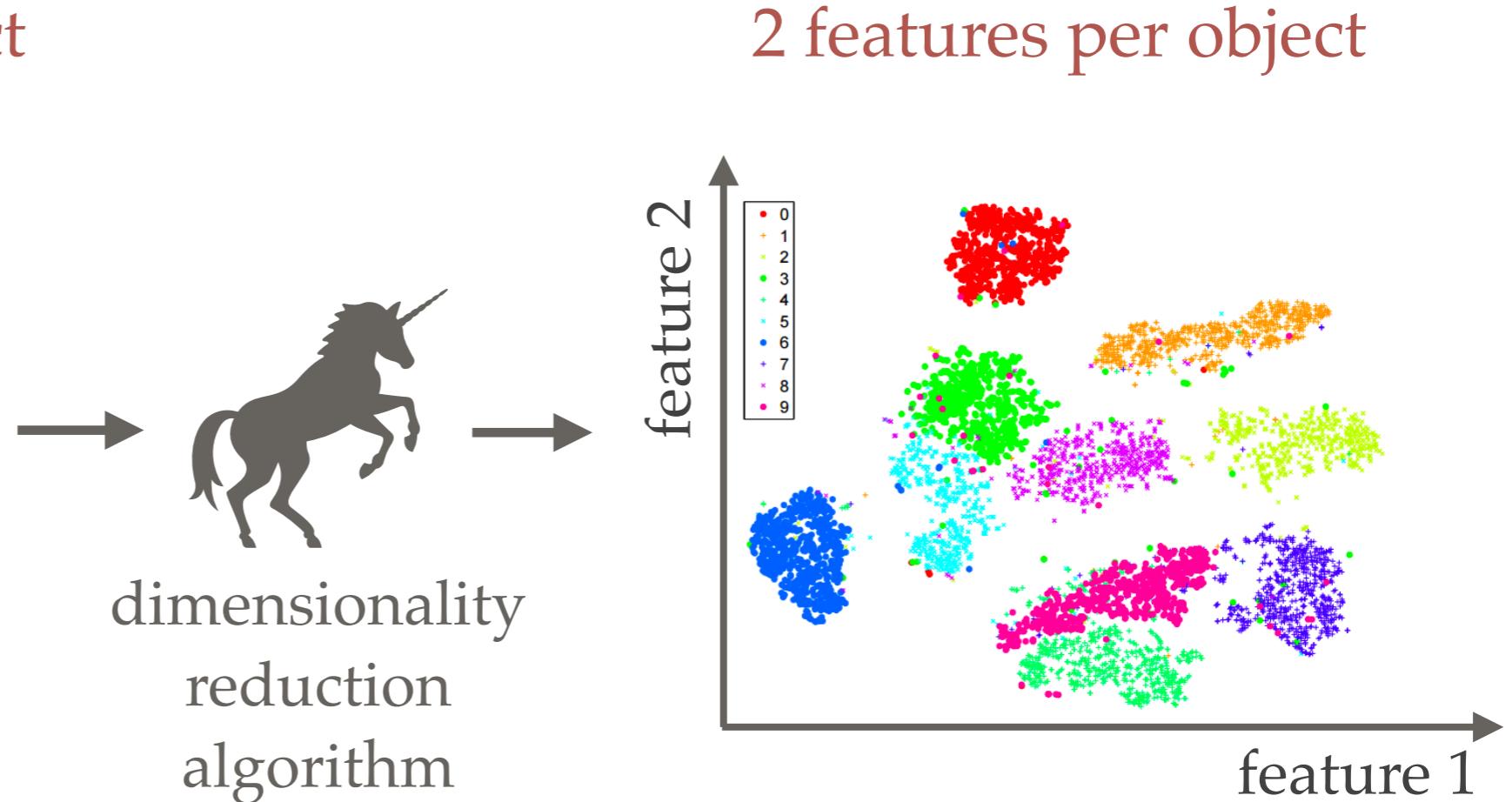
$$G_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}$$

How can we extract simplicity from the observed complex world?

# What is dimensionality reduction?

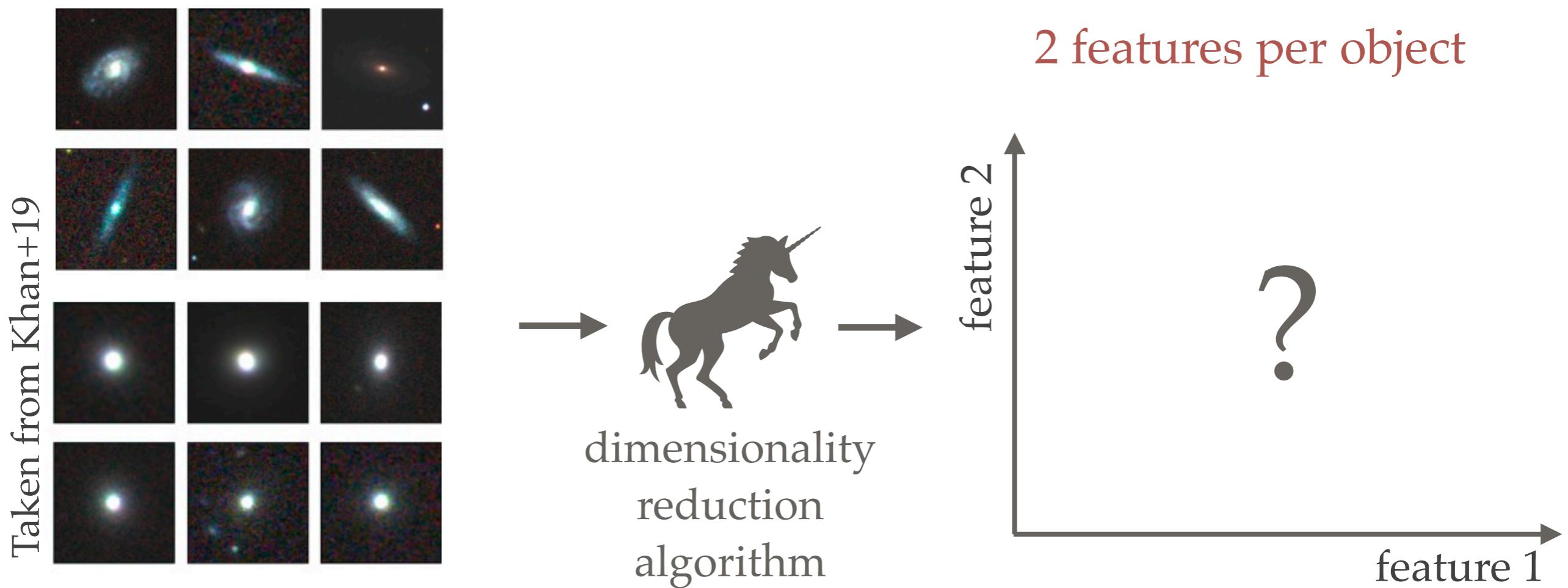
- ❖ Dimensionality reduction is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation *retains some meaningful* properties of the original data (taken from wiki).

28 x 28 features per object



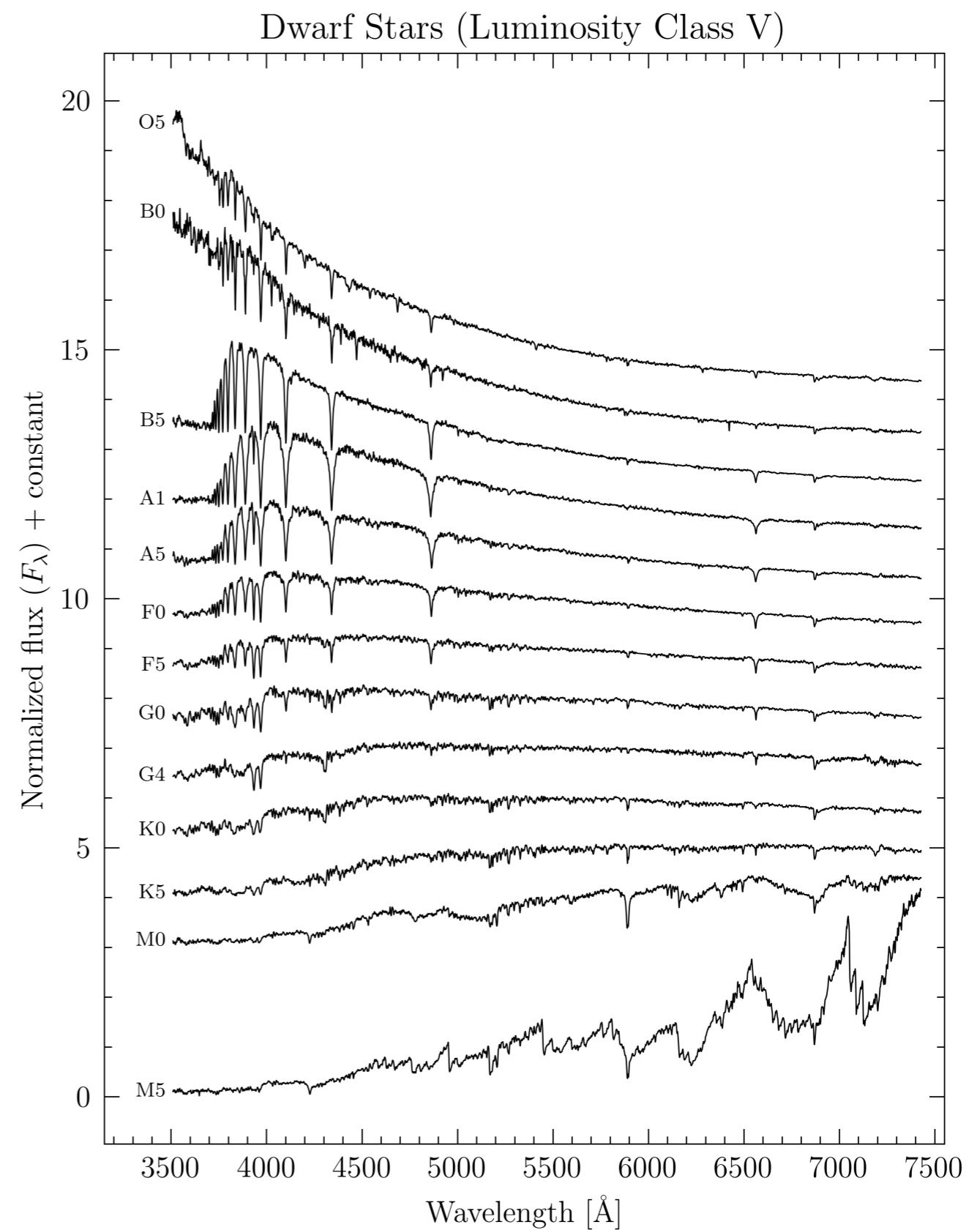
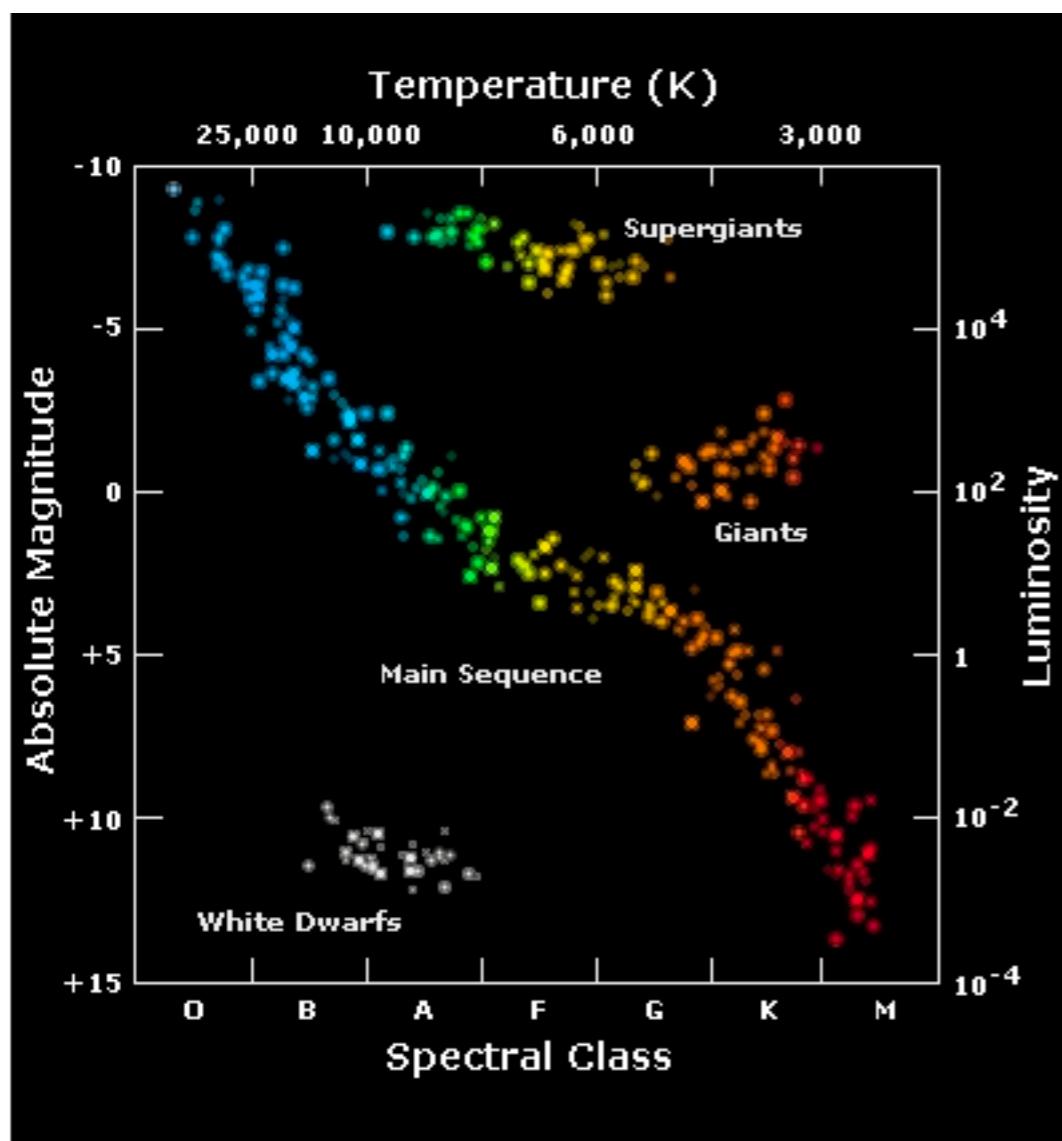
# What is dimensionality reduction?

- ❖ Dimensionality reduction is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation *retains some meaningful* properties of the original data (taken from wiki).



# “Traditional” dimensionality reduction in astronomy

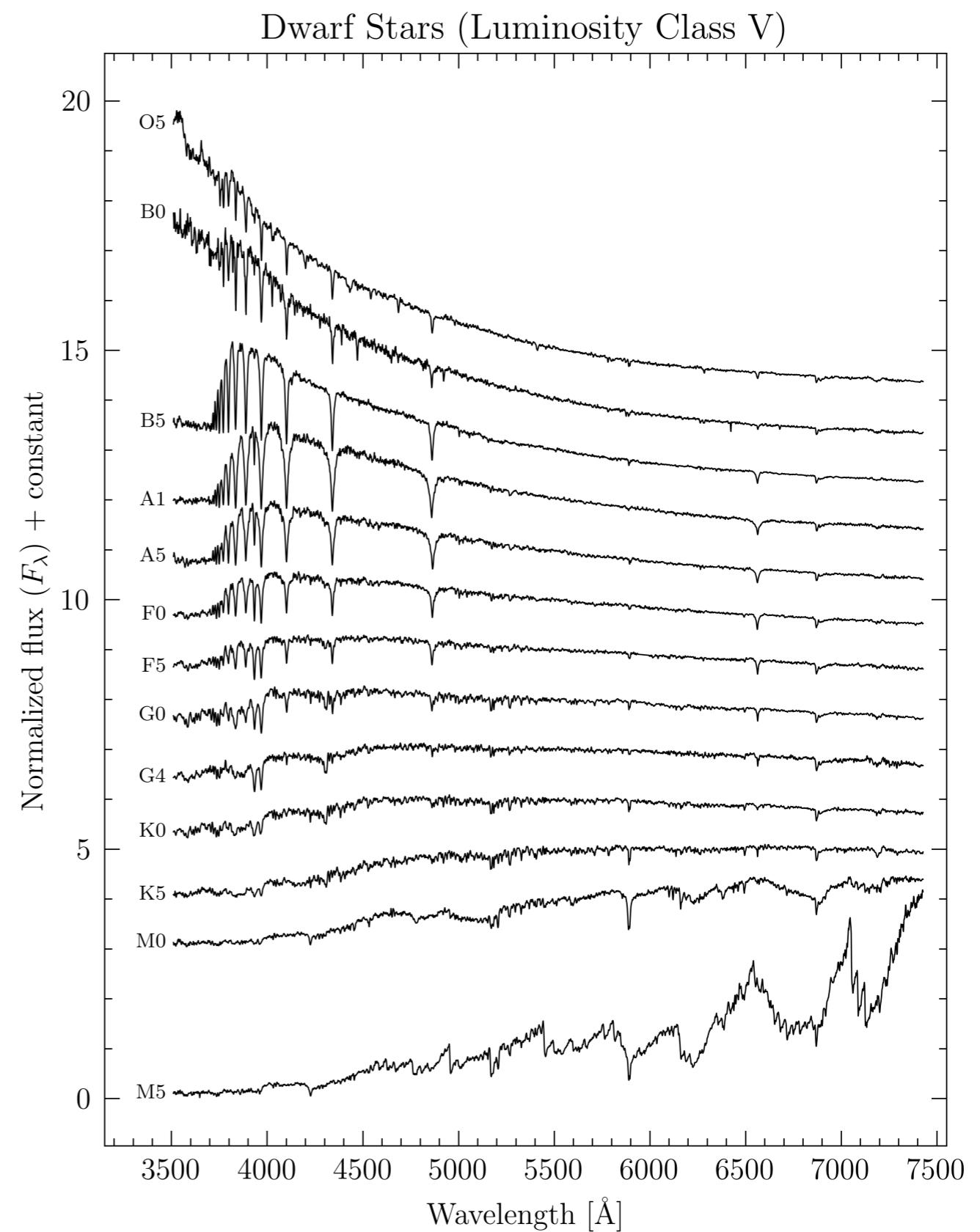
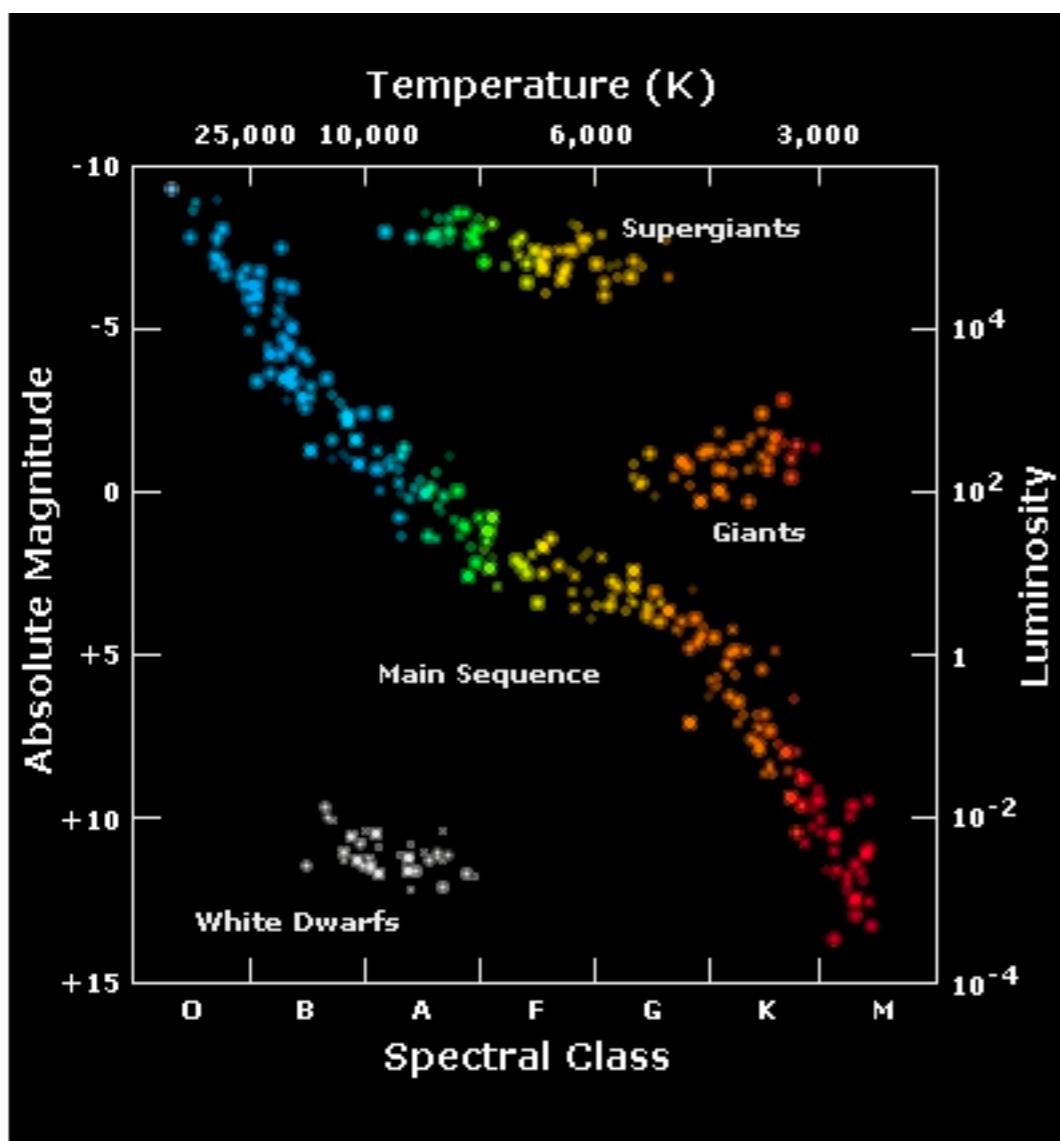
- ❖ The stellar sequence:



# “Traditional” dimensionality reduction in astronomy

- ❖ The stellar sequence:

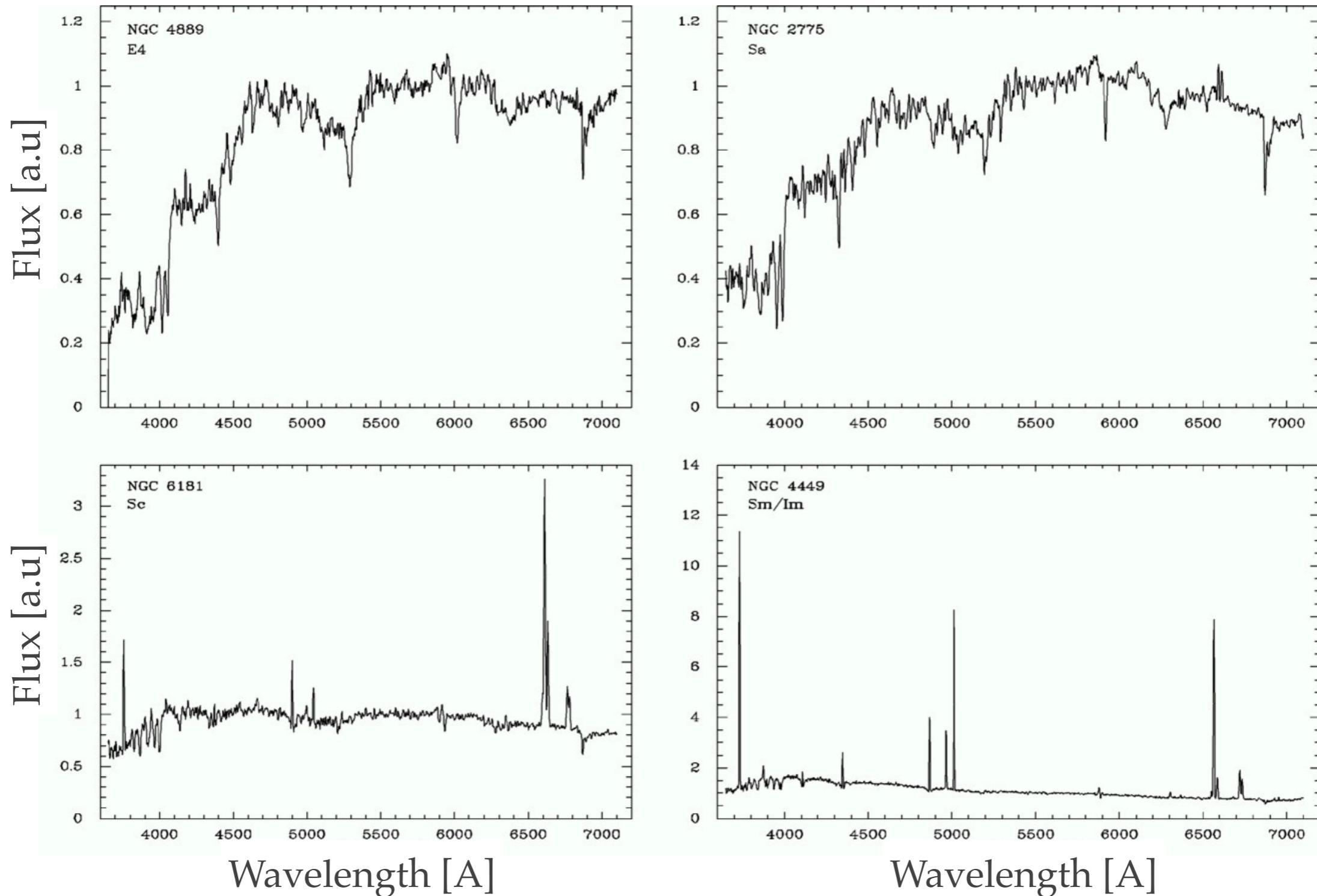
Having a déjà-vu?



# “Traditional” dimensionality reduction in astronomy

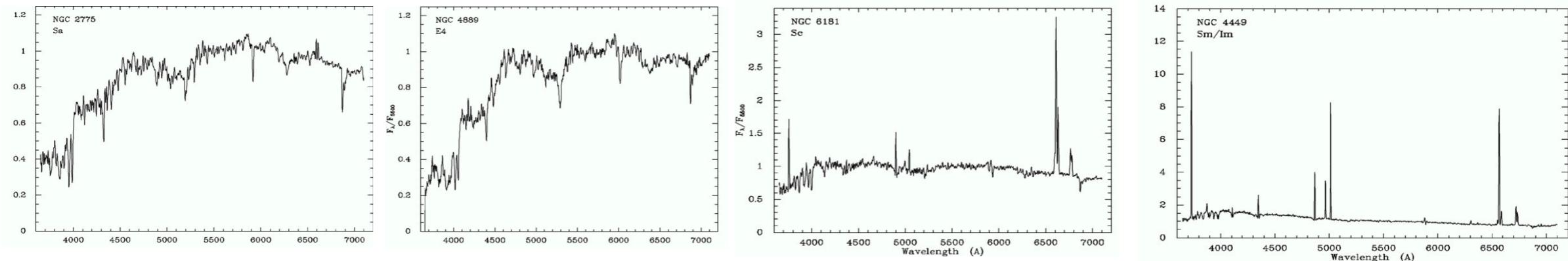
- ❖ Galaxy spectra: color-magnitude diagram and the BPT diagram

Optical spectra of galaxies (Kennicutt 1998)

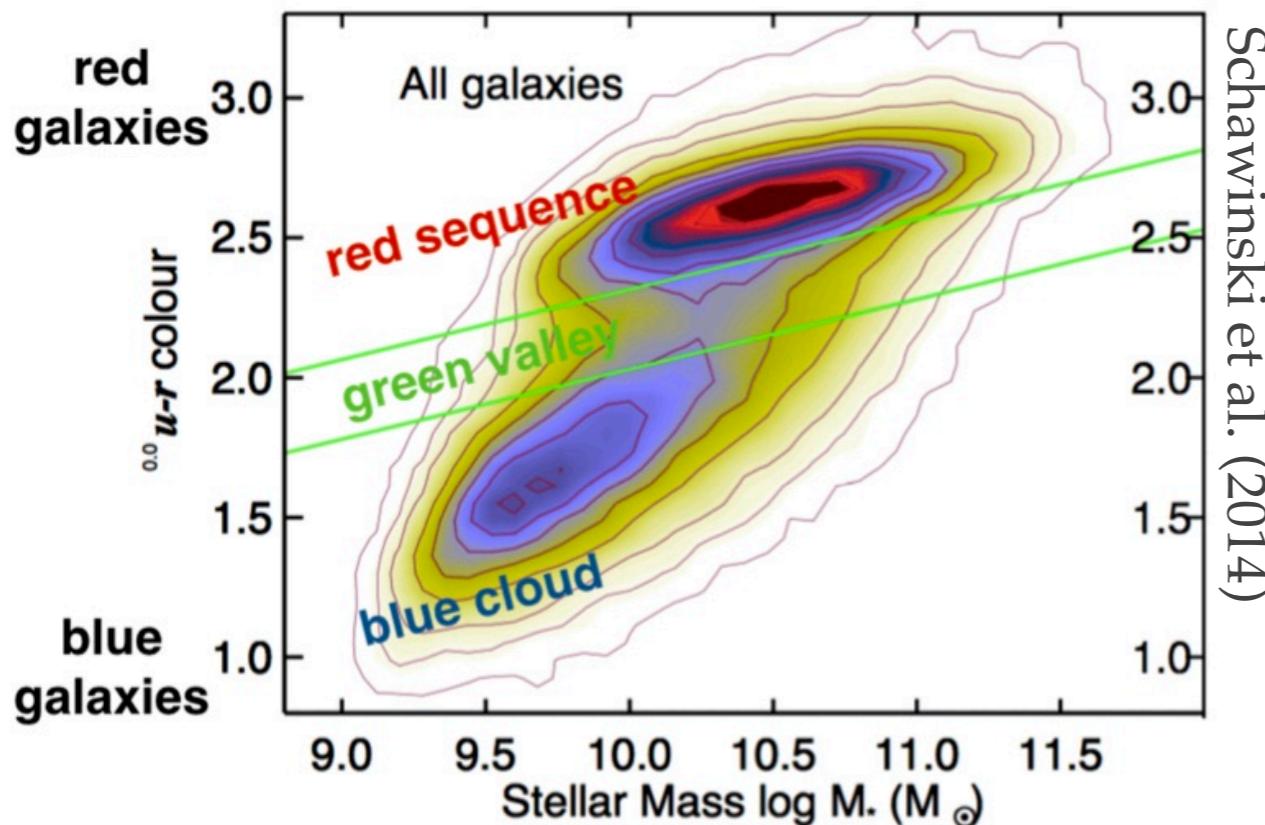


# “Traditional” dimensionality reduction in astronomy

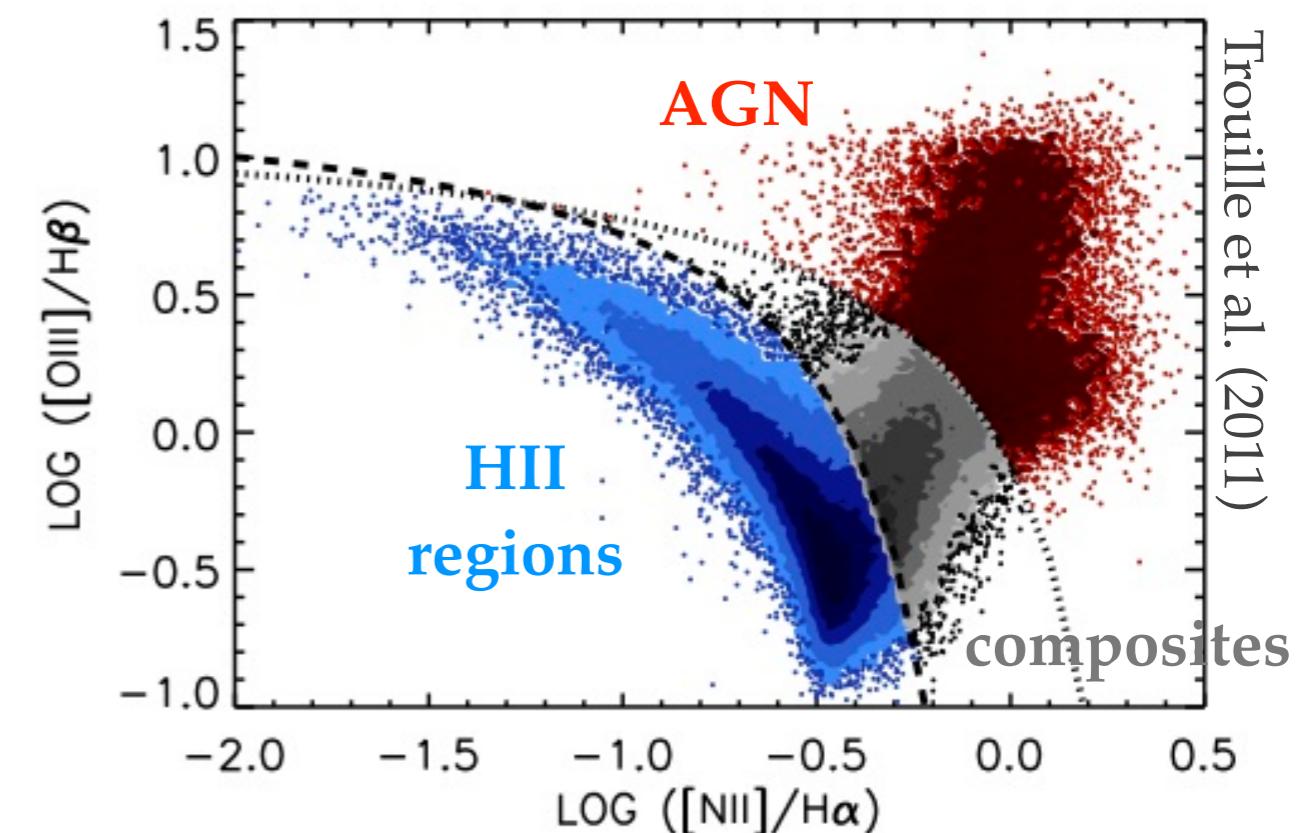
- ❖ Galaxy spectra: color-magnitude diagram and the BPT diagram



Color-magnitude diagram

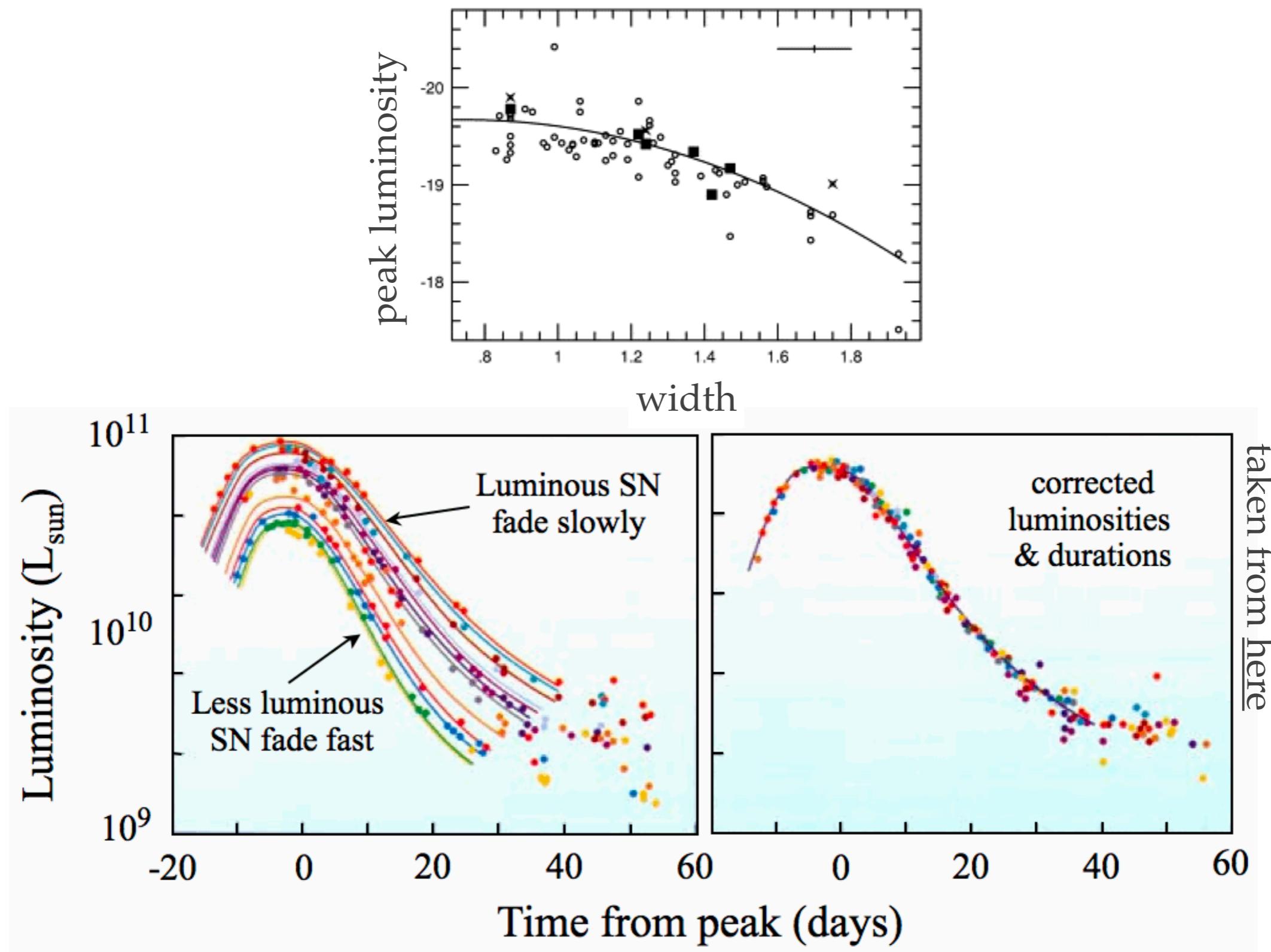


BPT diagram



# “Traditional” dimensionality reduction in astronomy

- ❖ Type Ia supernovae: Phillips relation



# What is fundamentally different now?

---

1. The volume and rate of information grows exponentially:
  - Sky survey now generate ~1 PB of data + derived products. Astronomical databases now routinely include  $10^6 - 10^9$  objects.
  - We can no longer manually-inspect all the data we collect.

# What is fundamentally different now?

2. A great increase in data dimensionality and complexity:
  - Data is heterogeneous and high-dimensional.
  - Patterns and correlations in the data can no longer be visualized in 3D.

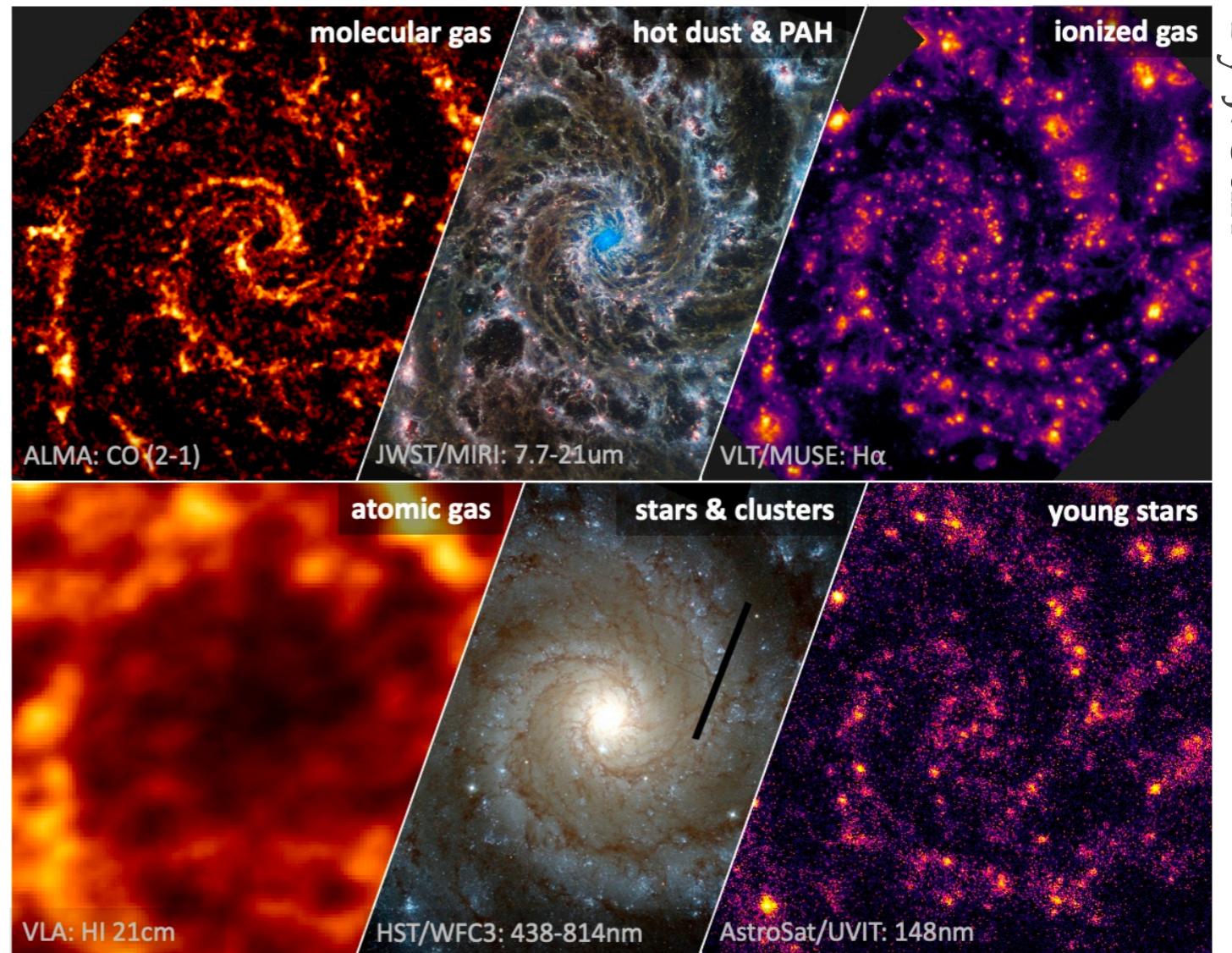
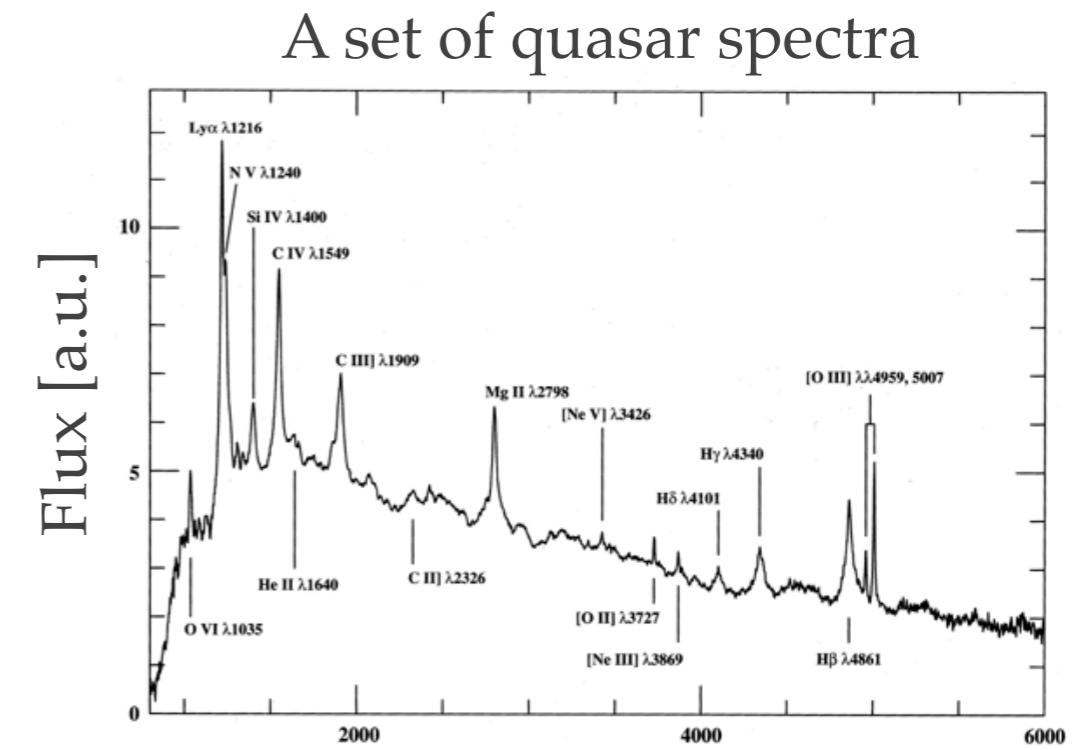
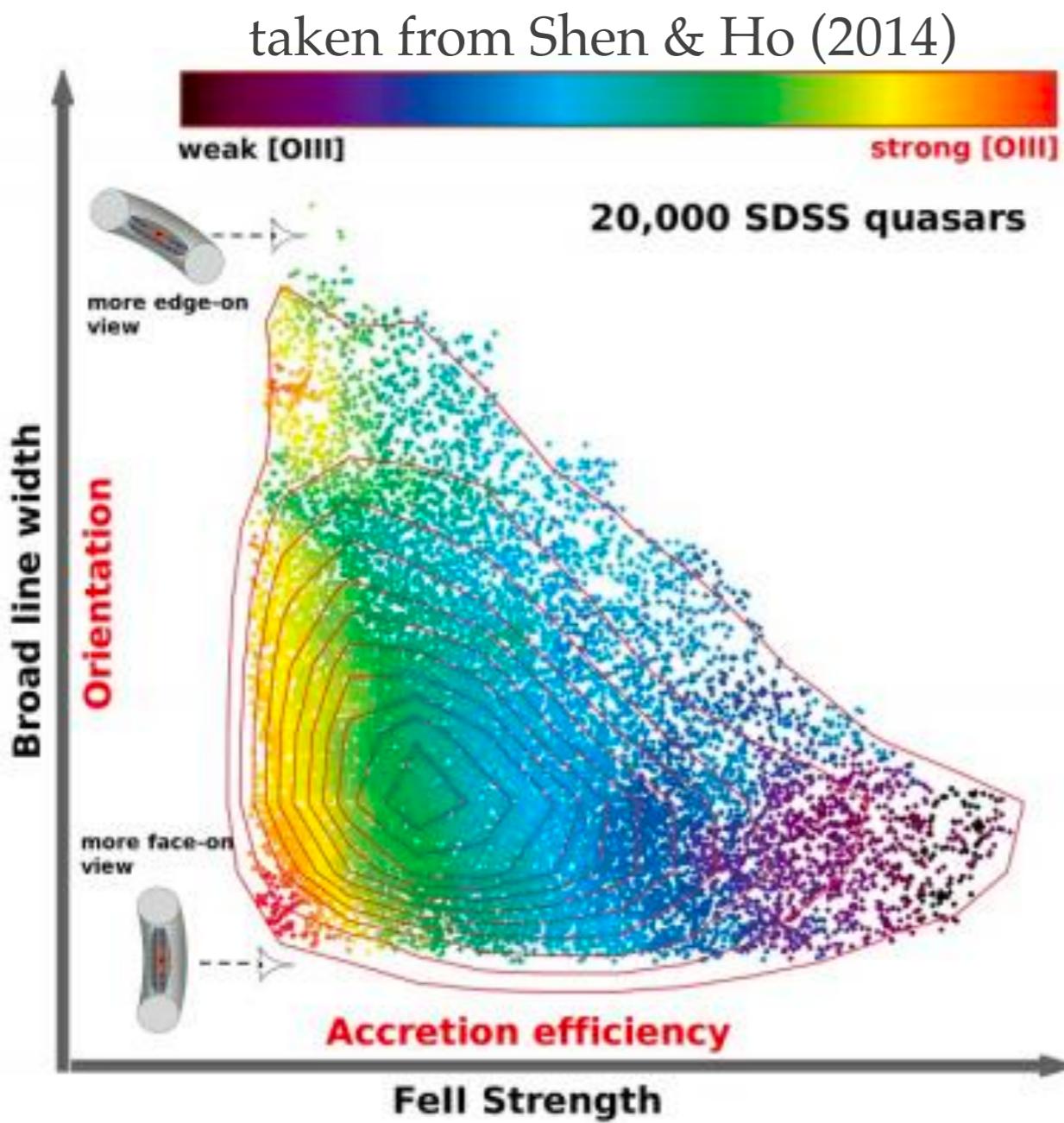


Image by the [PHANGS](#) collaboration

# More recent examples of dimensionality reduction

- ❖ PCA dimensionality reduction of quasar spectra (Boroson & Green 1992):

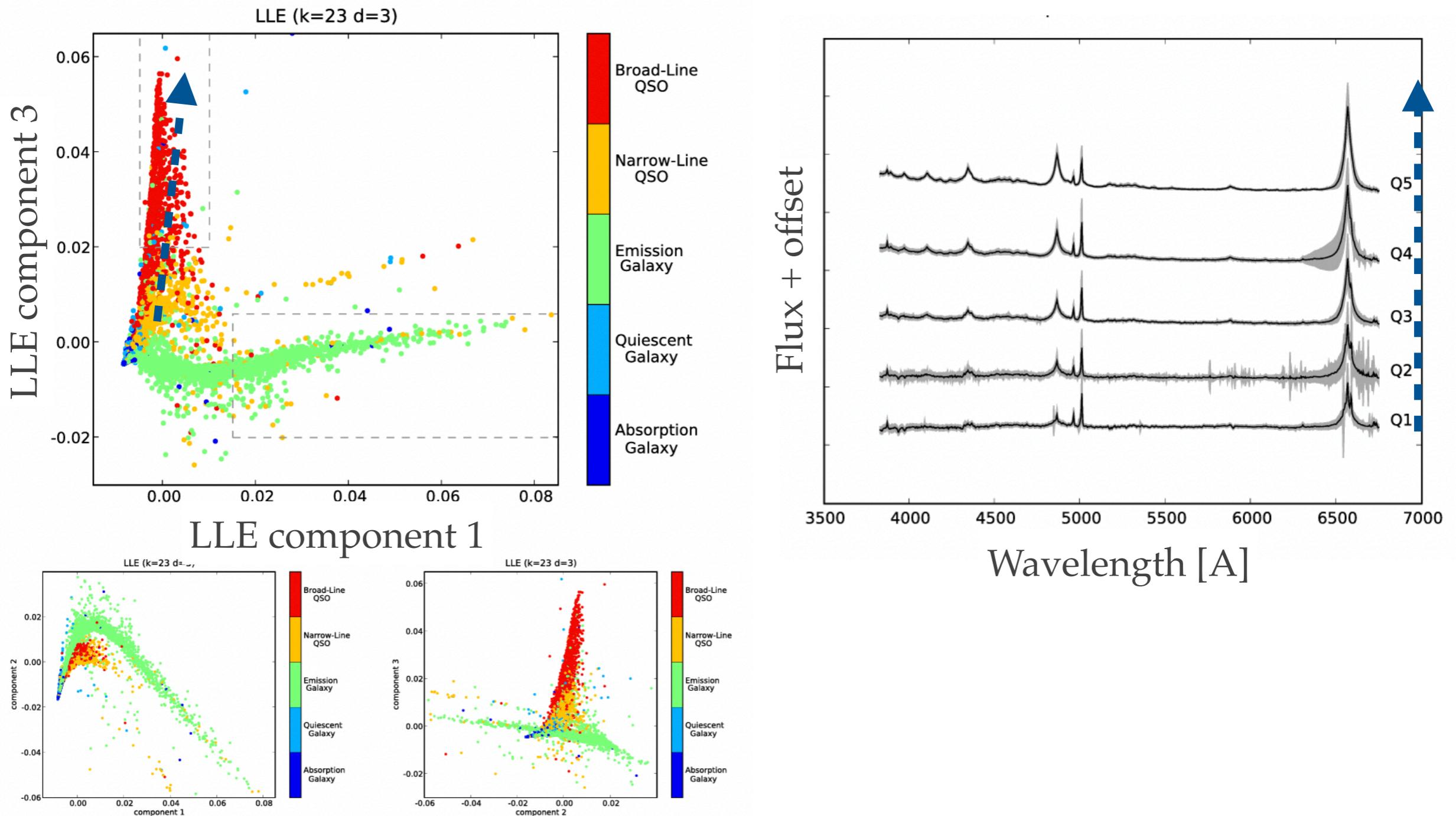


# A set of derived features

# PCA

# More recent examples of dimensionality reduction

- ❖ Locally Linear Embedding of optical galaxy spectra (Vanderplas & Connolly 2009):



# Different use cases

---

- ❖ Uncover new trends / correlations.
- ❖ Data visualization and interpretation.
- ❖ Look for outliers or interesting objects.

# Different use cases

---

- ❖ Uncover new trends / correlations.
- ❖ Data visualization and interpretation.
- ❖ Look for outliers or interesting objects.
- ❖ Improve performance of supervised machine learning:
  - ❖ Original features can be correlated and redundant.
  - ❖ Most algorithms cannot handle thousands of features.

# Different use cases

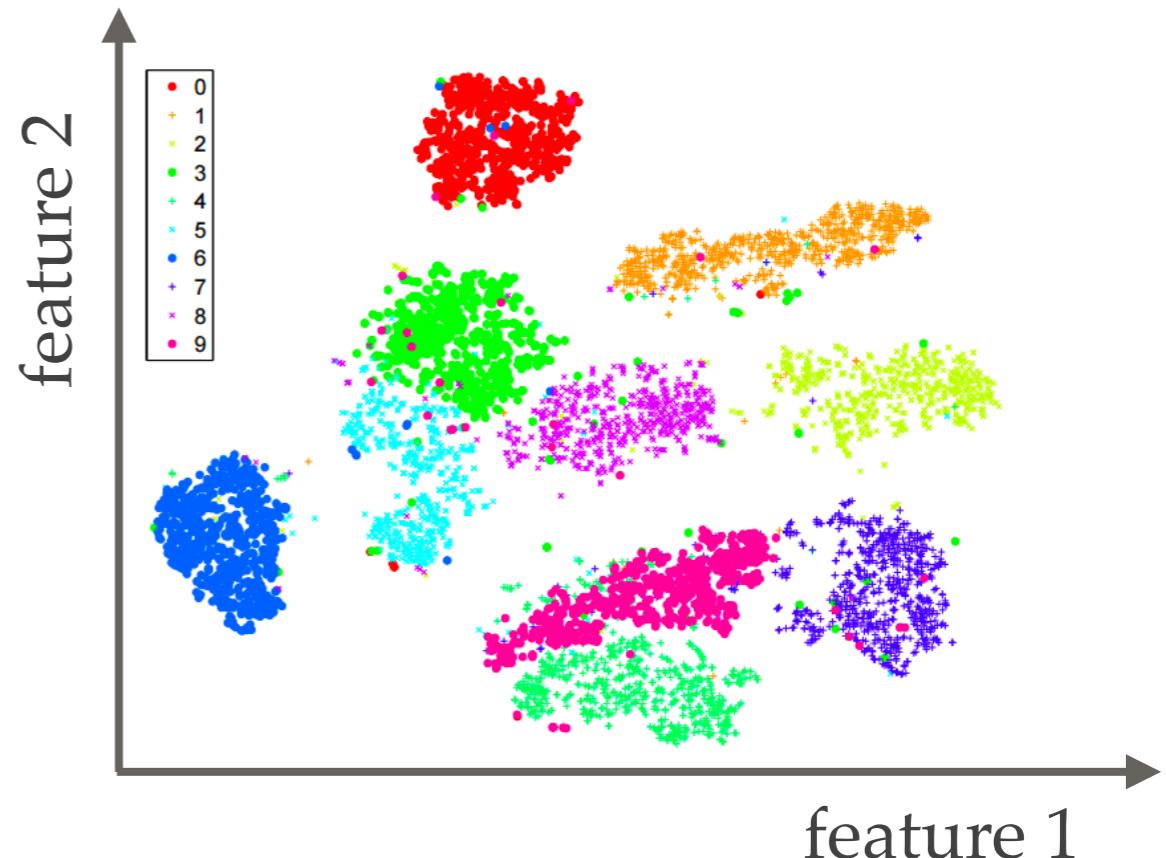
---

- ❖ Uncover new trends / correlations.
- ❖ Data visualization and interpretation.
- ❖ Look for outliers or interesting objects.
- ❖ Improve performance of supervised machine learning:
  - ❖ Original features can be correlated and redundant.
  - ❖ Most algorithms cannot handle thousands of features.
  - ❖ Compressing data (e.g., the Square Kilometre Array; SKA).

# Two types of outputs

1. Low-dimensional embedding of our dataset.

This is a common output of all dimensionality reduction algorithms: PCA, ICA, NNMF, LLE, SOM, tSNE, UMAP, etc.



2. Prototypes or Eigen-vectors.

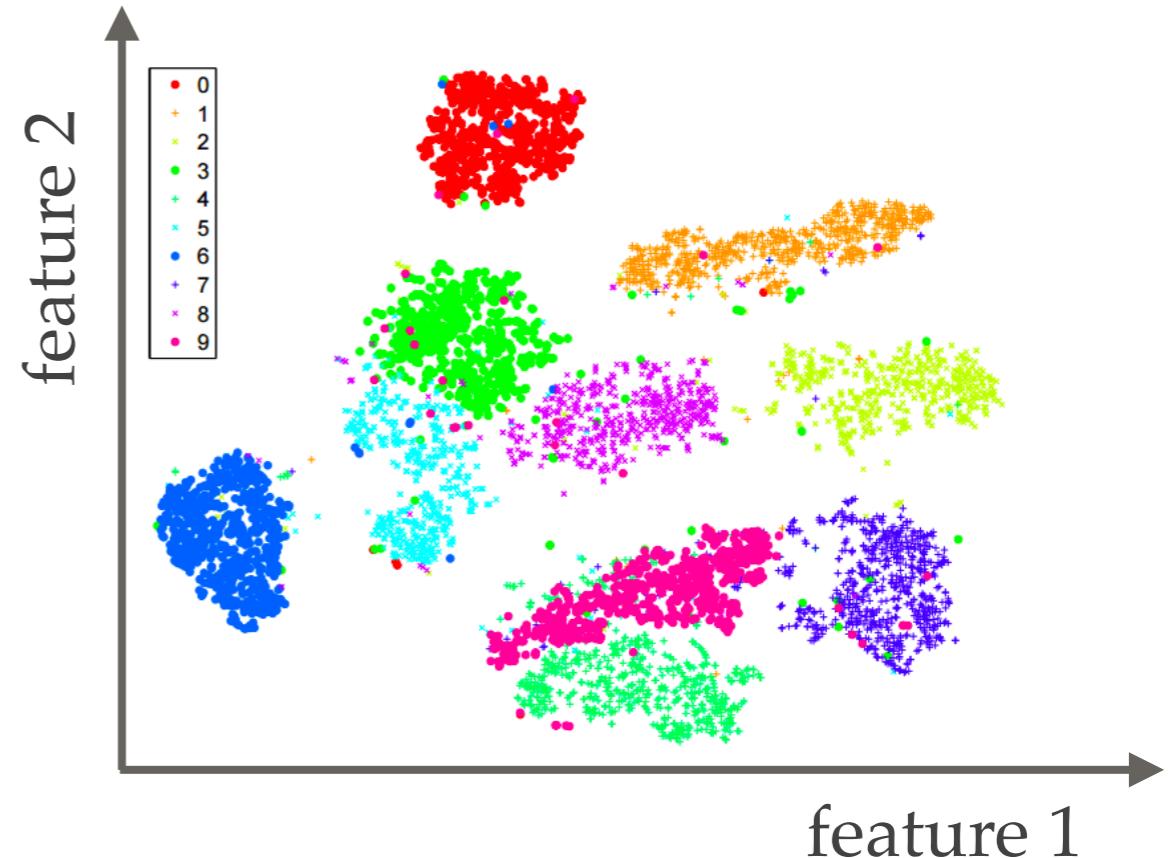
Along with the low-dimensional embedding, this is the output of: PCA, ICA, NNMF, and SOM.



# Two types of outputs

1. Low-dimensional embedding of our dataset.

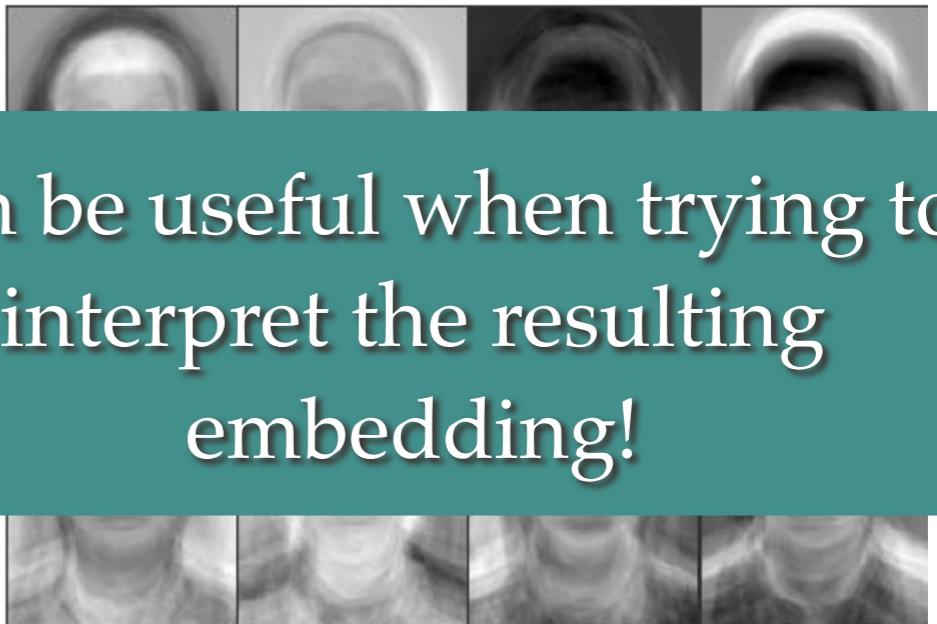
This is a common output of all dimensionality reduction algorithms: PCA, ICA, NNMF, LLE, SOM, tSNE, UMAP, etc.



2. Prototypes or Eigen-vectors.

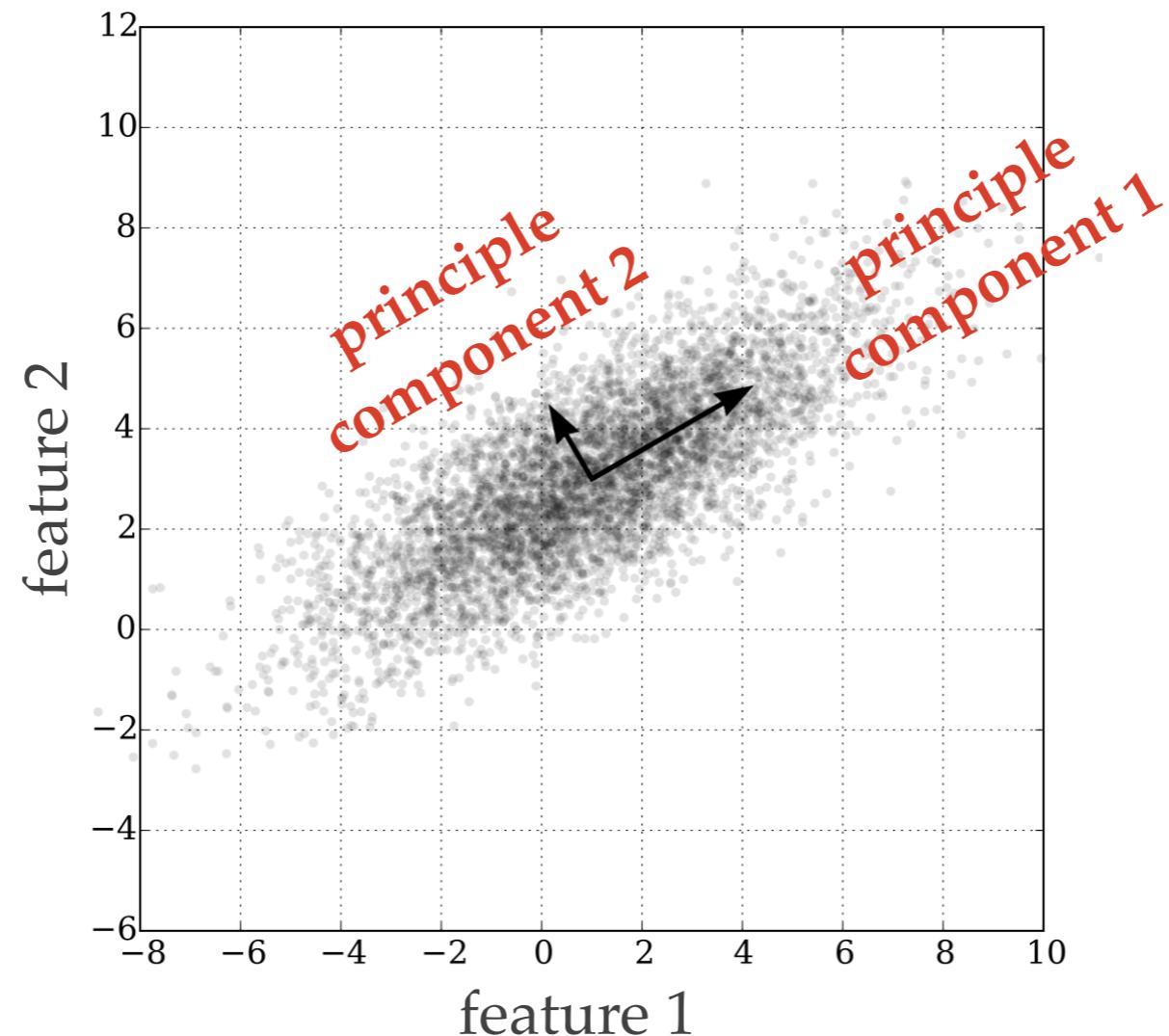
Along with the low-dimensional embedding, this is the output of: PCA, ICA, NNMF, and SOM.

Can be useful when trying to interpret the resulting embedding!



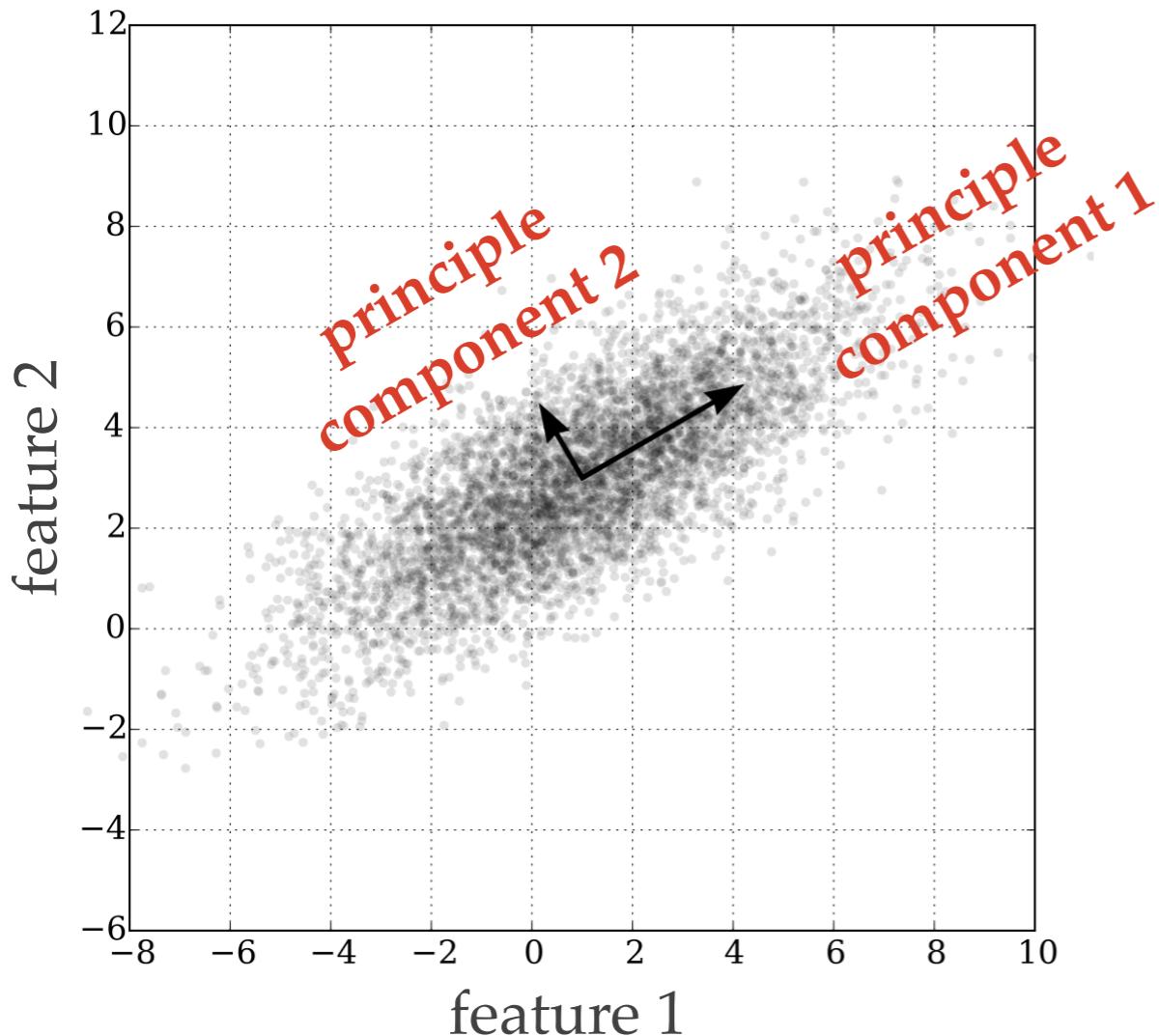
# Principle Component Analysis (PCA)

- ❖ PCA is one of the most popular dimensionality reduction algorithms in machine learning and data science.
- ❖ It is a linear transformation of the input data into a new coordinate system, defined by the *principle components*.



# Principle Component Analysis (PCA)

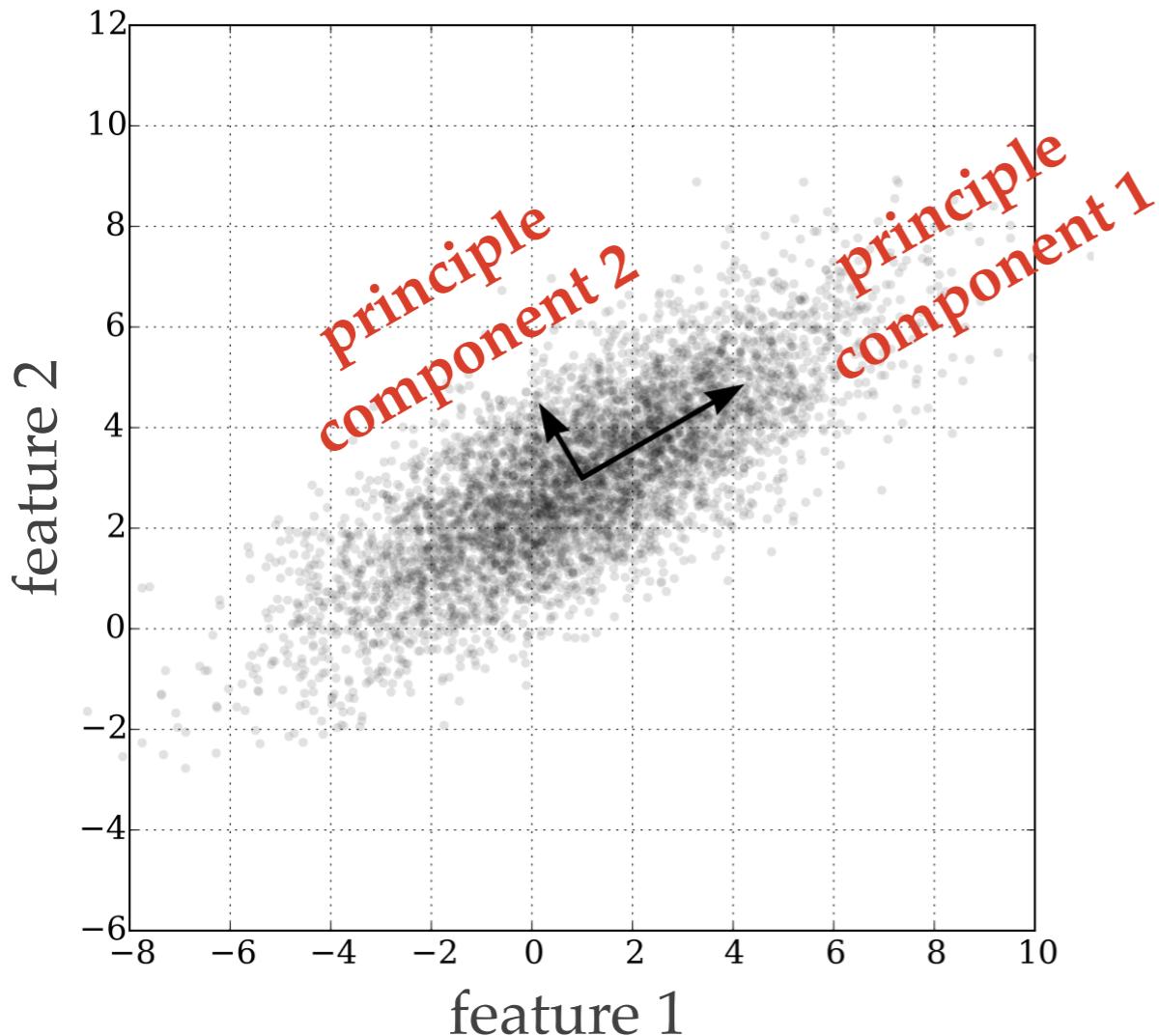
- ❖ PCA is one of the most popular dimensionality reduction algorithms in machine learning and data science.
- ❖ It is a linear transformation of the input data into a new coordinate system, defined by the *principle components*.



- ❖ The first principle component has the largest possible variance.

# Principle Component Analysis (PCA)

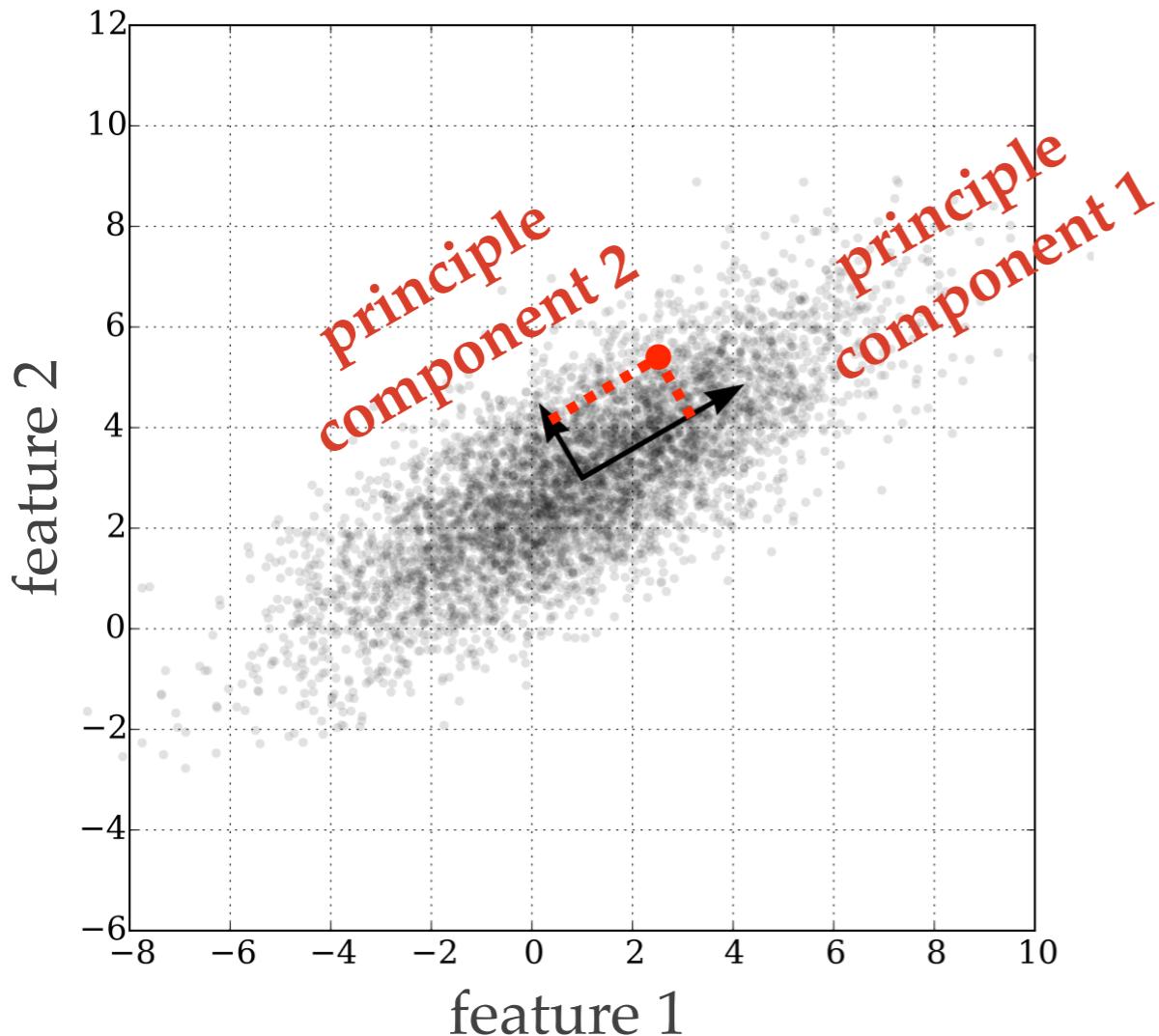
- ❖ PCA is one of the most popular dimensionality reduction algorithms in machine learning and data science.
- ❖ It is a linear transformation of the input data into a new coordinate system, defined by the *principle components*.



- ❖ The first principle component has the largest possible variance.
- ❖ Each succeeding component has the highest possible variance, under the constrain that it is orthogonal to the preceding components.

# Principle Component Analysis (PCA)

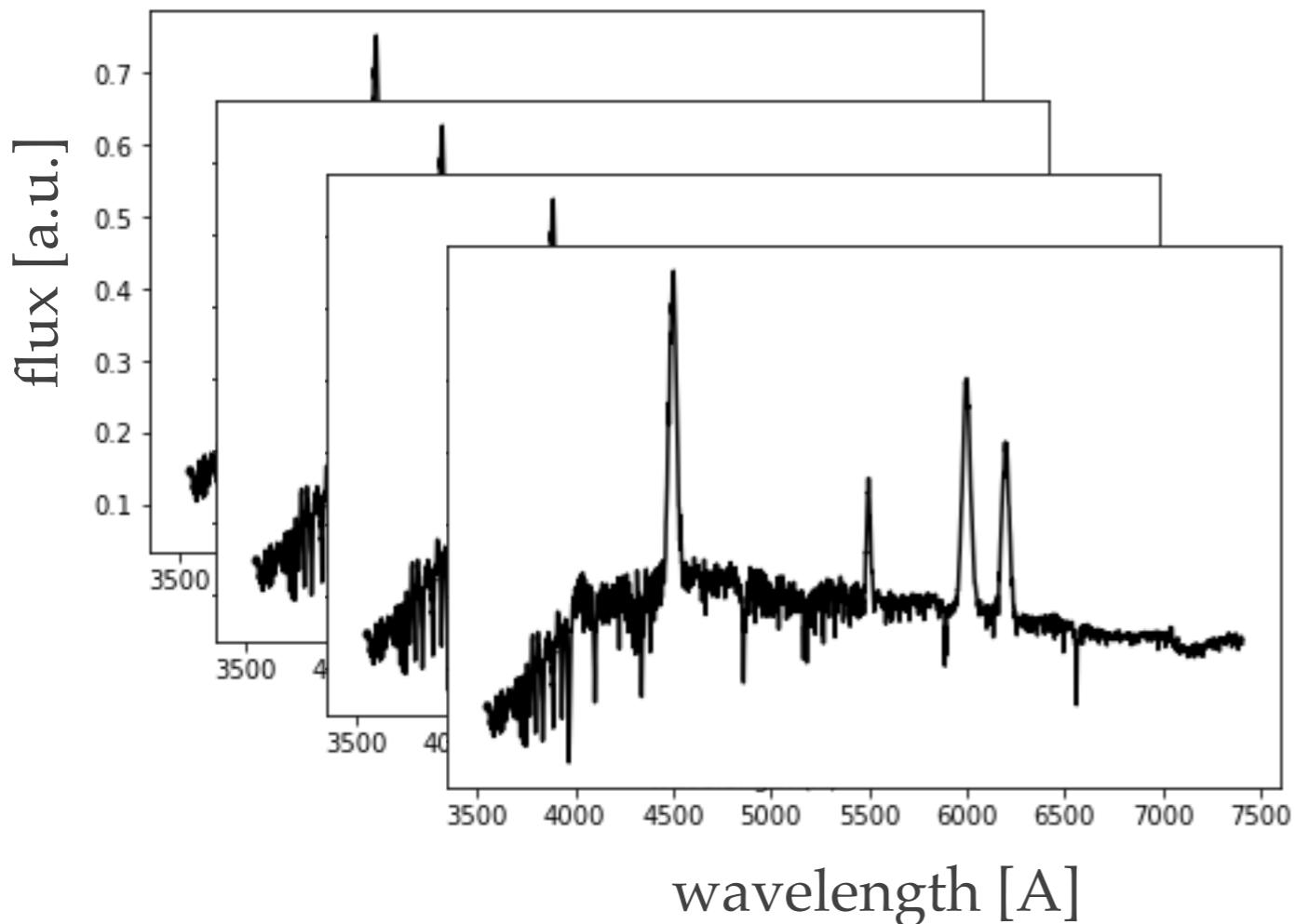
- ❖ PCA is one of the most popular dimensionality reduction algorithms in machine learning and data science.
- ❖ It is a linear transformation of the input data into a new coordinate system, defined by the *principle components*.



- ❖ The first principle component has the largest possible variance.
- ❖ Each succeeding component has the highest possible variance, under the constrain that it is orthogonal to the preceding components.
- ❖ Every object in the input data is described as a n-dimensional point in this new coordinate system.

# PCA - step by step

- We will go through the basic steps of PCA using an example of simulated galaxy spectra [see Jupyter notebook!]:

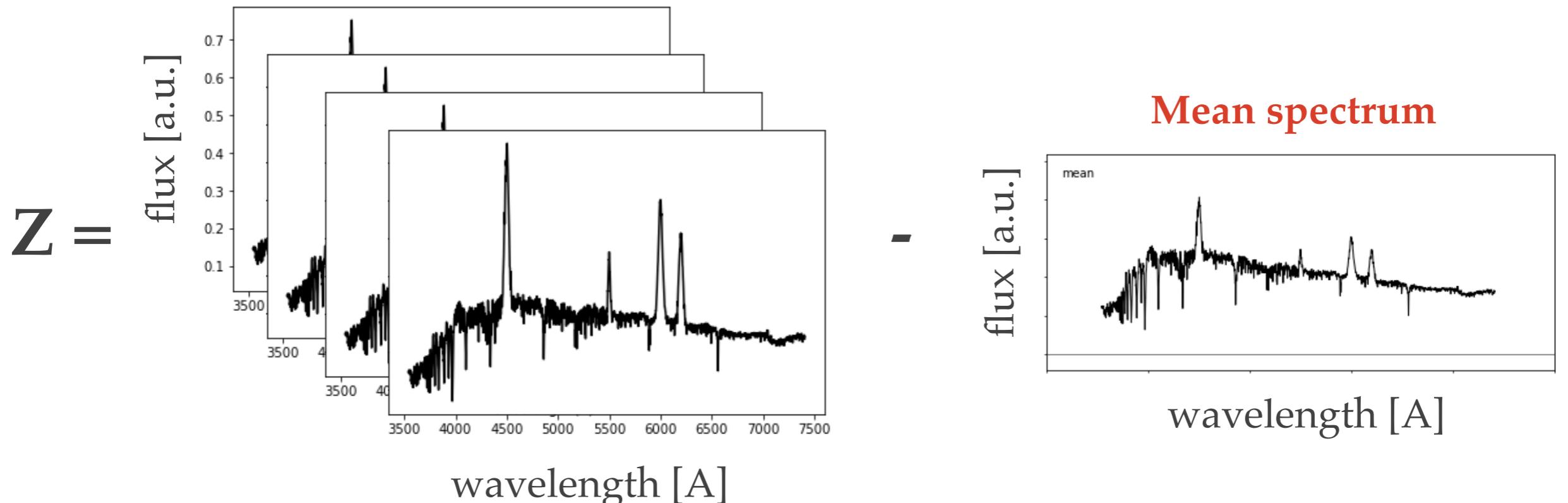


- There are 1,000 spectra.
- Each spectrum has 4300 flux measurements => each object in the input is a 4300-dimensional object.

Our goal is use PCA to reduce the dimensions of this high-dimensional dataset.

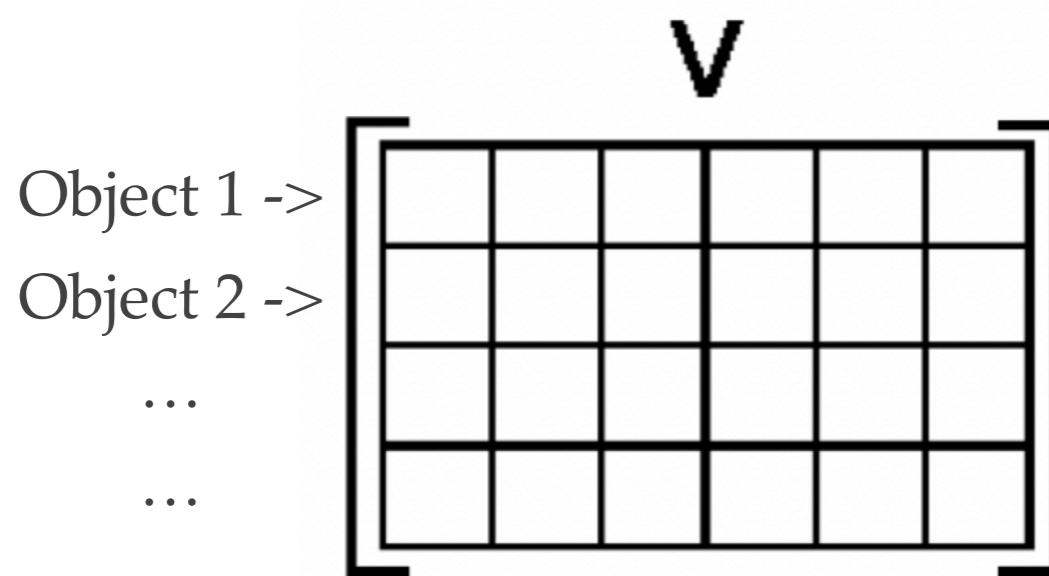
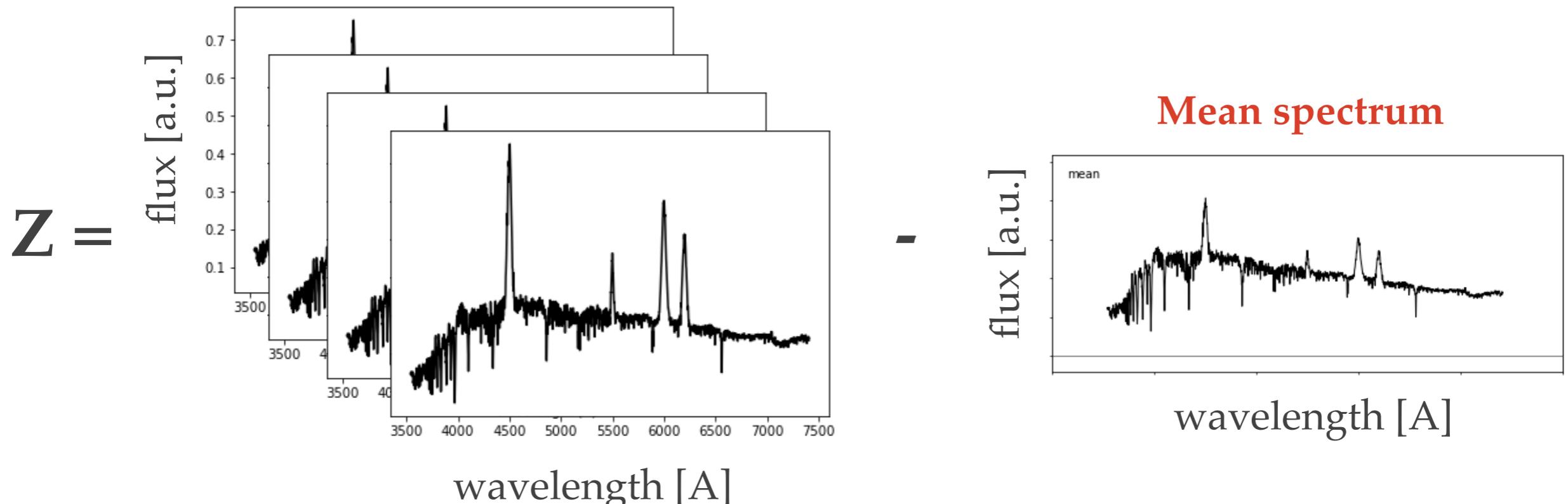
# PCA - step by step

Step I - calculate and subtract the mean of the different objects.



# PCA - step by step

Step I - calculate and subtract the mean of the different objects.



# PCA - step by step

Step I - calculate and subtract the mean of the different objects.

$Z =$



-

Mean image



$V$

Object 1 ->

Object 2 ->

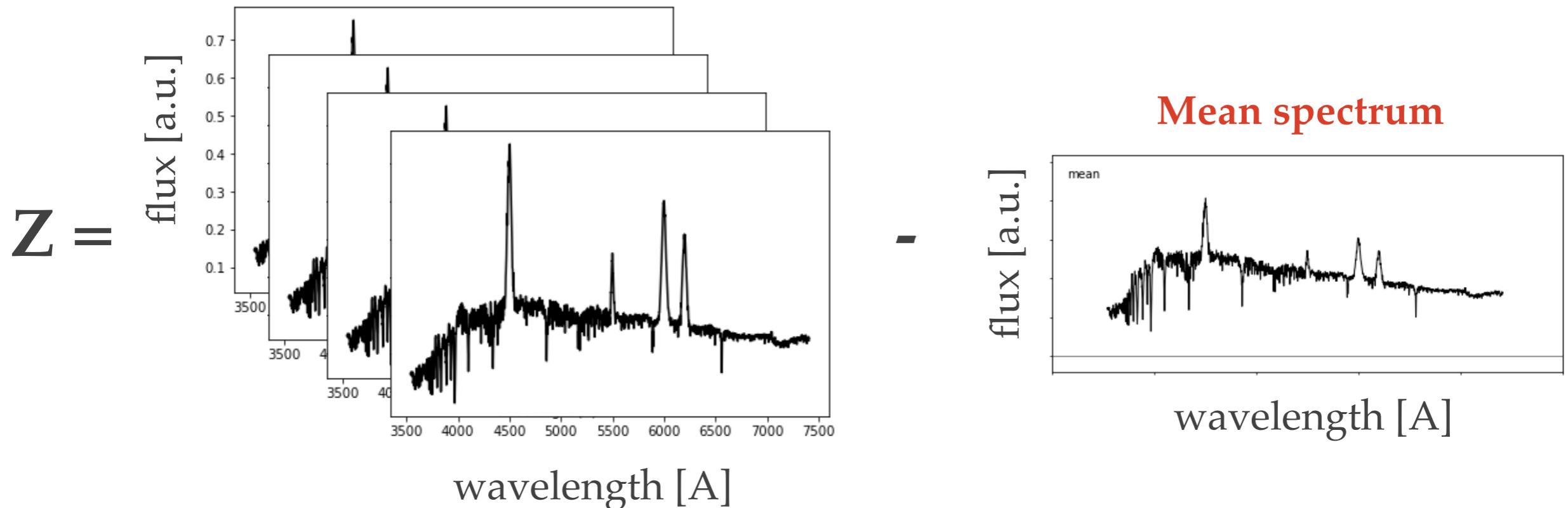
...

...

Object 1 ->				
Object 2 ->				
...				
...				

# PCA - step by step

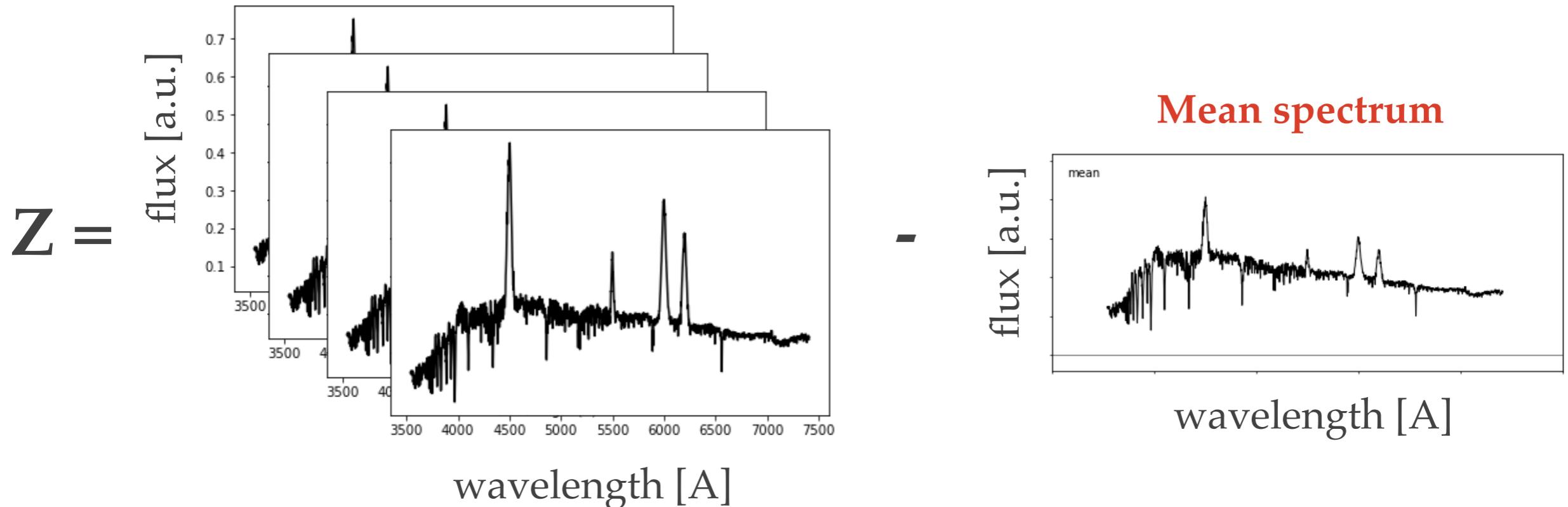
Step I - calculate and subtract the mean of the different objects.



Step II - calculate the covariance matrix of  $Z$ :  $C = Z^T Z$

# PCA - step by step

Step I - calculate and subtract the mean of the different objects.

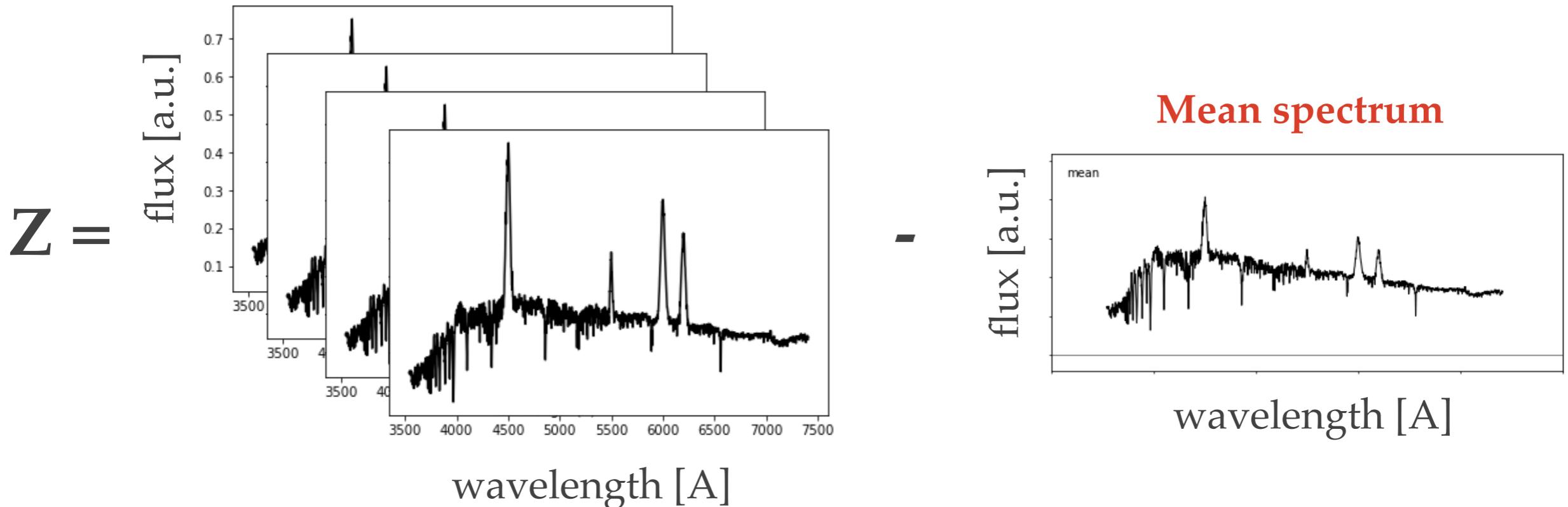


Step II - calculate the covariance matrix of  $Z$ :  $C = Z^T Z$

Step III - calculate the eigenvectors and the corresponding eigenvalues of the covariance matrix:  $Z^T Z = PDP^{-1}$

# PCA - step by step

Step I - calculate and subtract the mean of the different objects.



Step II - calculate the covariance matrix of  $Z$ :  $C = Z^T Z$

Step III - calculate the eigenvectors and the corresponding eigenvalues of the covariance matrix:  $Z^T Z = PDP^{-1}$

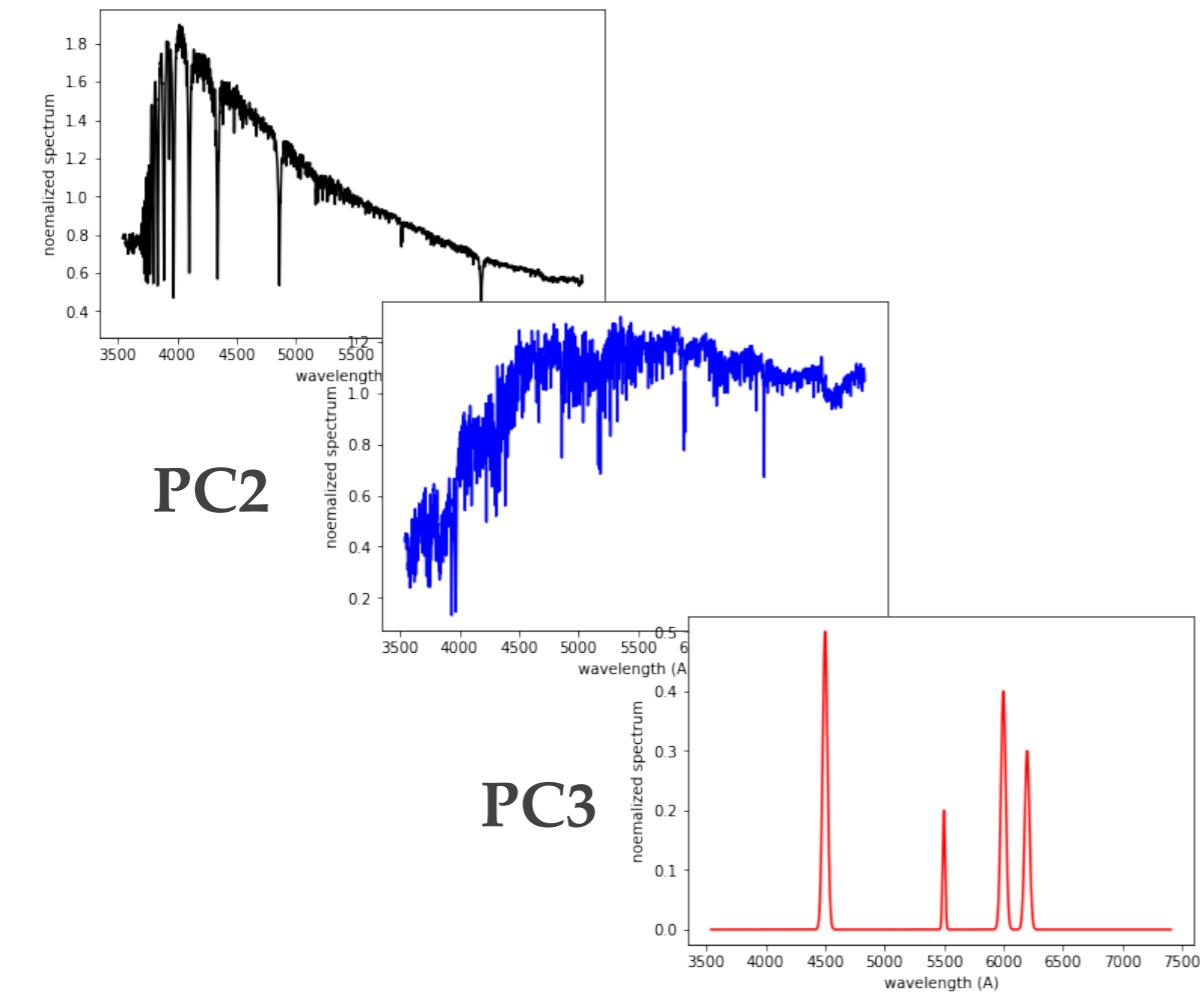
Diagonal matrix containing eigenvalues ordered from the largest to the smallest

# PCA - step by step

Step III - calculate the eigenvectors and the corresponding eigenvalues of the covariance matrix:

$$Z^T Z = PDP^{-1}$$

The matrix of the eigenvectors, or 'loadings'



Diagonal matrix containing eigenvalues ordered from the largest to the smallest

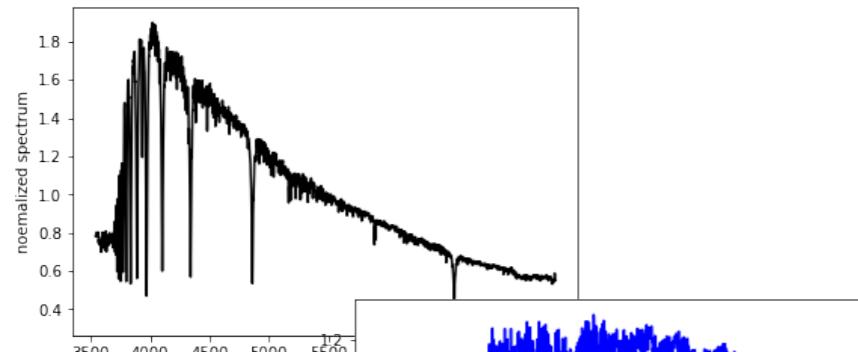
# PCA - step by step

Step III - calculate the eigenvectors and the corresponding eigenvalues of the covariance matrix:

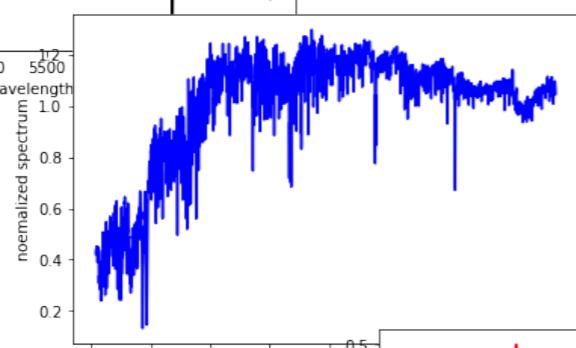
$$Z^T Z = PDP^{-1}$$

The matrix of the eigenvectors, or 'loadings'

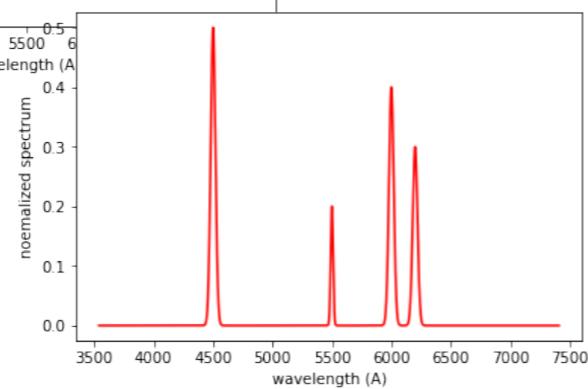
PC1



PC2

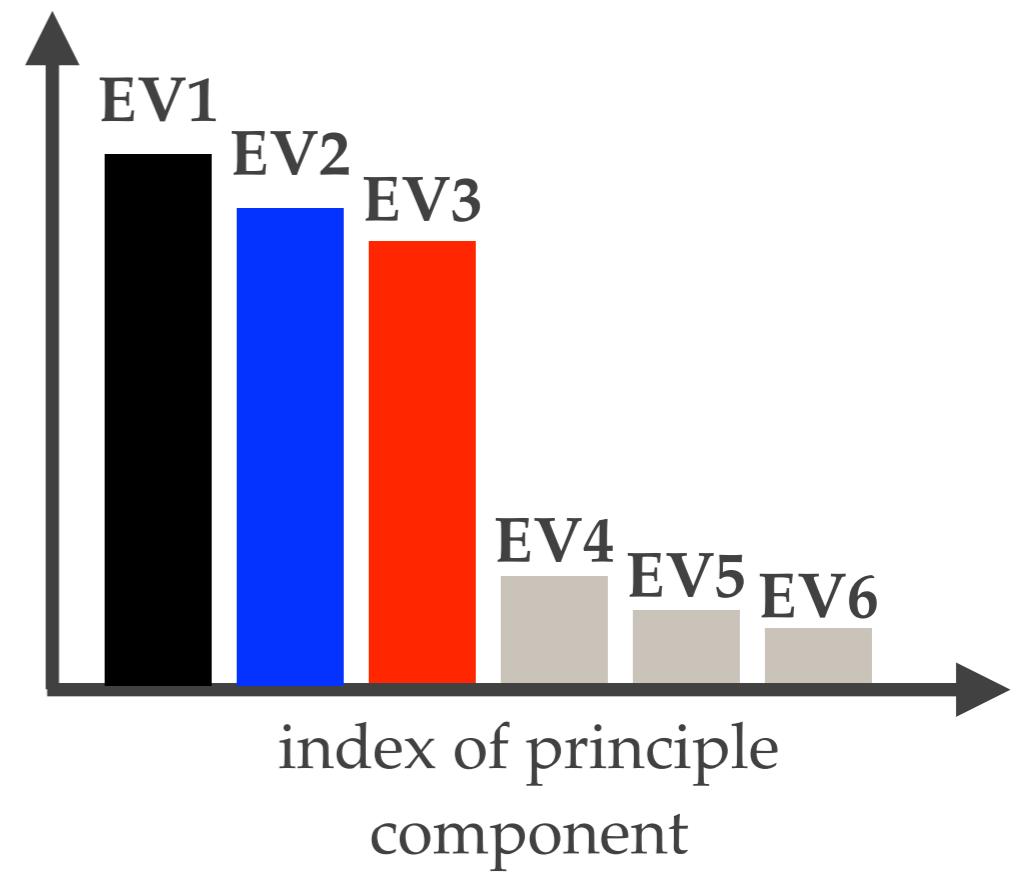


PC3



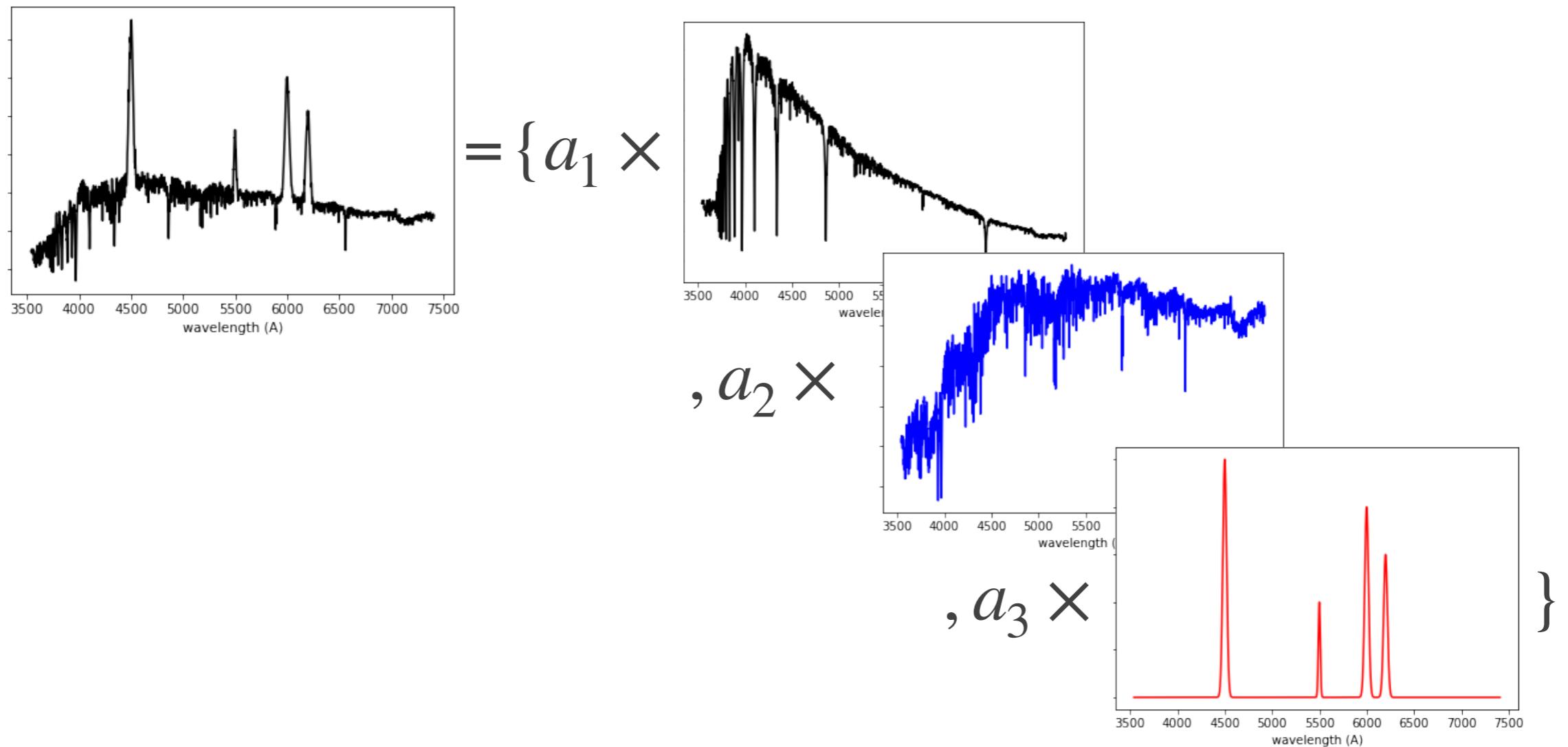
Diagonal matrix containing eigenvalues ordered from the largest to the smallest

eigenvalue = variance



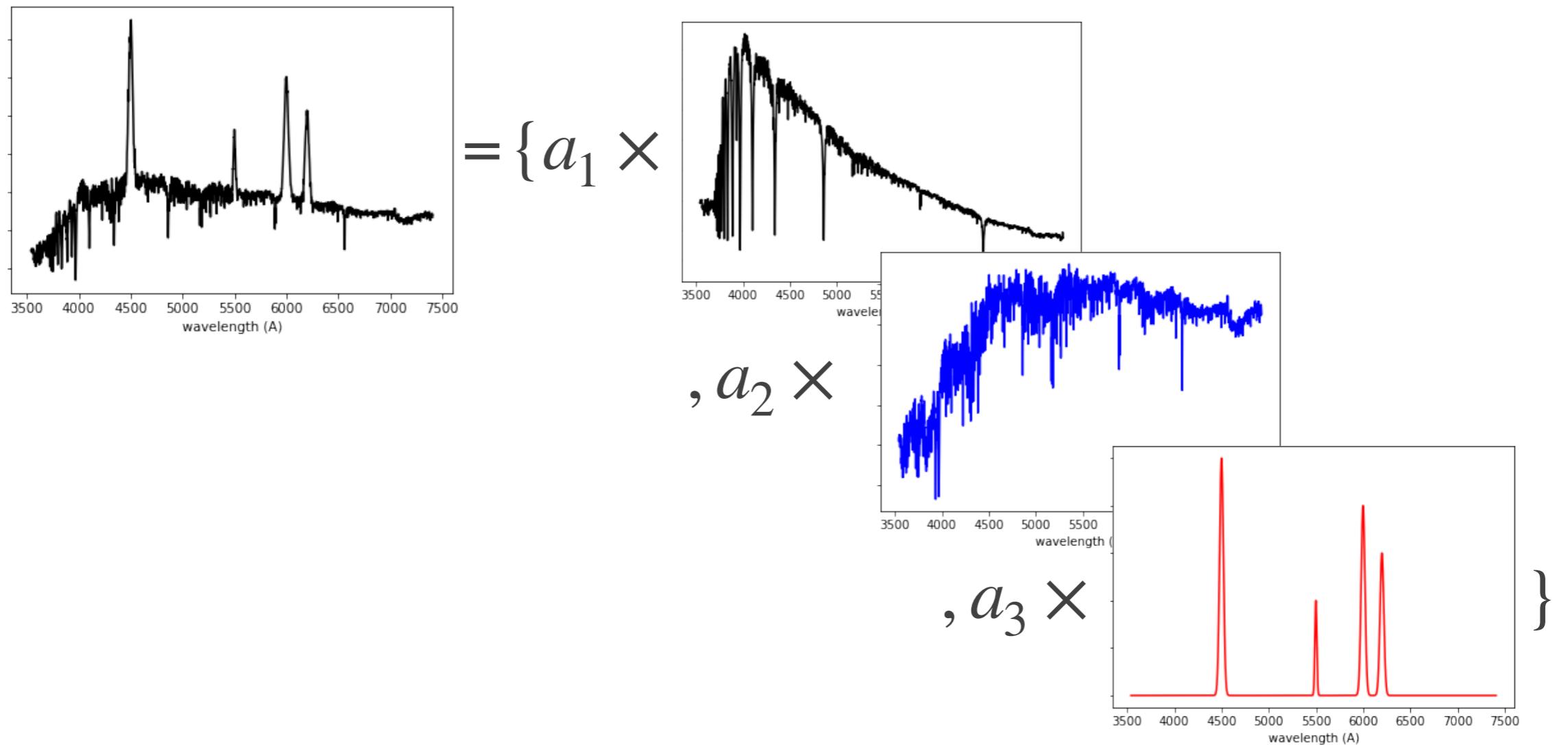
# PCA - step by step

Step IV - transform the normalized input data to the new coordinate system:  $V = ZP$ . Each object is a linear combination of the principle components.



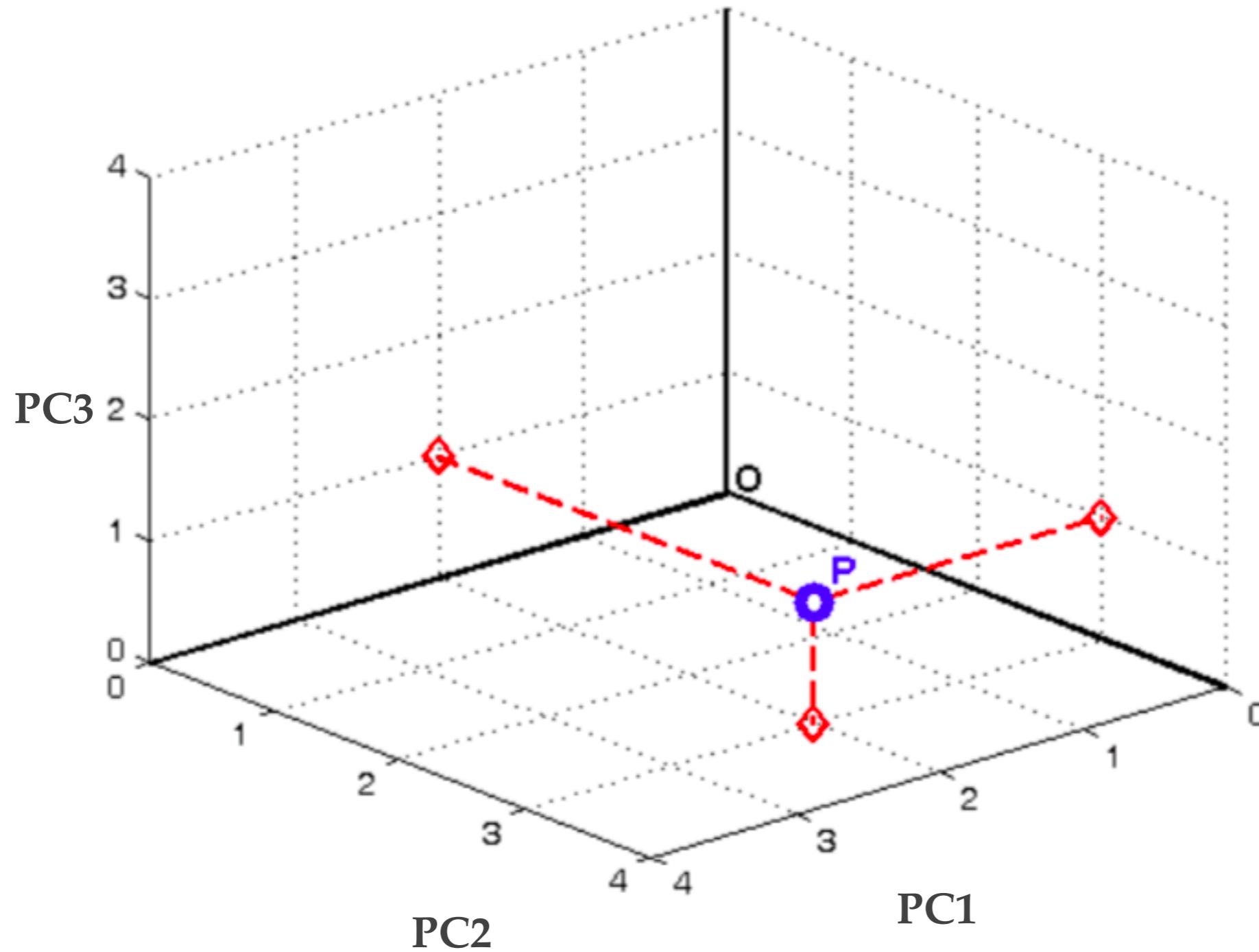
# PCA - step by step

Step IV - transform the normalized input data to the new coordinate system:  $V = ZP$ . Each object is a linear combination of the principle components.



**Now every spectrum is described using 3 numbers, rather than 4300.**

# PCA - step by step



Now every spectrum is described using 3 numbers, rather than 4300.

# PCA - pros and cons

---

- ❖ **Advantages:**
  - ❖ Very simple and intuitive to use.
  - ❖ No free parameters!
  - ❖ Optimized to reduce variance.
- ❖ **Disadvantages:**
  - ❖ Linear decomposition: we will not be able to describe absorption lines, dust extinction, distance, etc..
  - ❖ Can produce negative principle components, which is not always physical in astronomy.

- 
- ❖ To the Jupyter notebook!

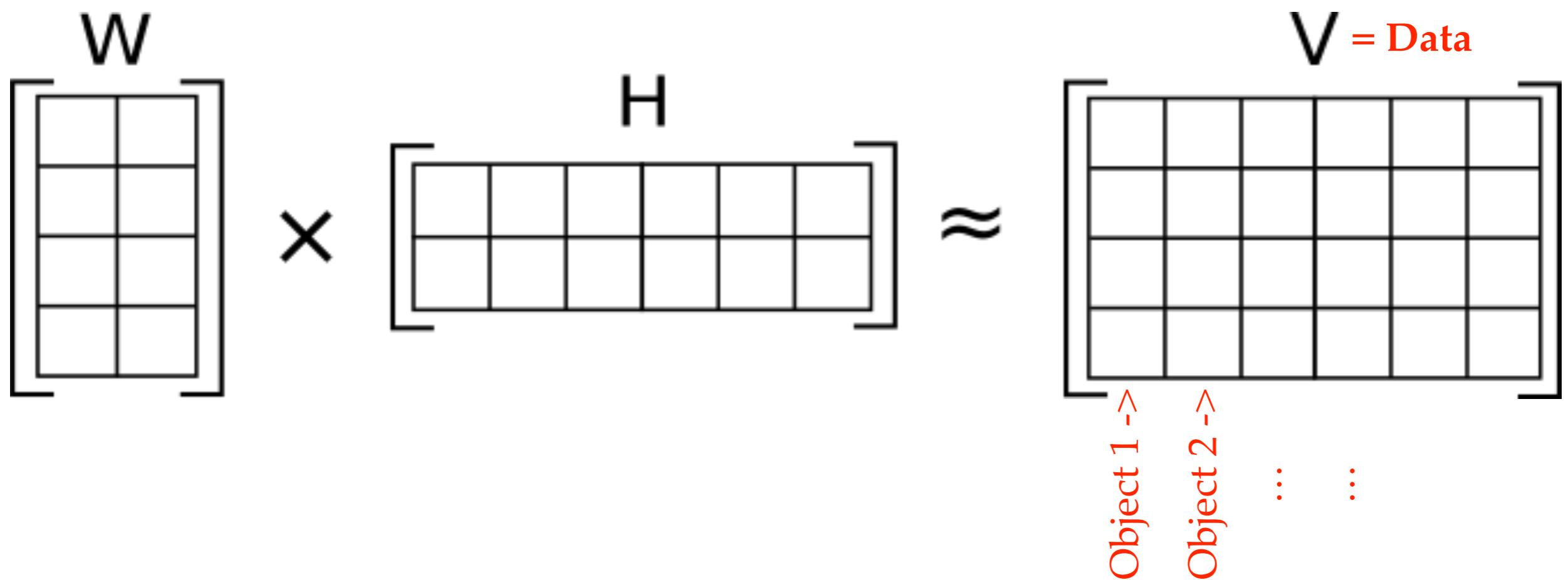
# Non-Negative Matrix Factorization (NMF)

- ❖ In non-negative matrix factorization, a large matrix  $V$  into (usually much smaller) matrices  $W$  and  $H$ , where all three matrices have no negative elements.
  - ❖ By applying such a decomposition, we essentially assume that the (non-negative) signal is composed of a sum of positive contributions: **combine parts to form a whole.**

$$\begin{matrix} W \\ \left[ \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \right] \end{matrix} \times \begin{matrix} H \\ \left[ \begin{array}{|c|c|c|c|c|} \hline & & & & \\ \hline \end{array} \right] \end{matrix} \approx \begin{matrix} V \\ \left[ \begin{array}{|c|c|c|c|c|c|} \hline & & & & & \\ \hline \end{array} \right] \end{matrix}$$

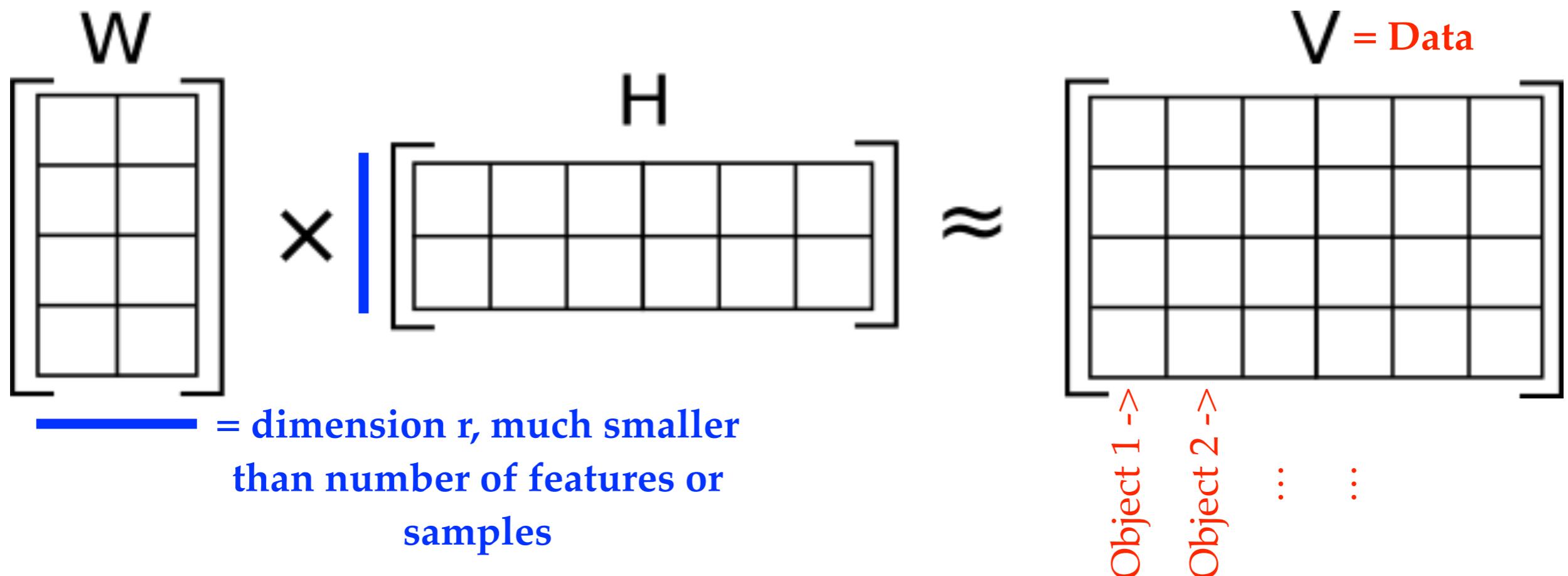
# Non-Negative Matrix Factorization (NMF)

- ❖ In non-negative matrix factorization, a large matrix  $V$  into (usually much smaller) matrices  $W$  and  $H$ , where all three matrices have no negative elements.
  - ❖ By applying such a decomposition, we essentially assume that the (non-negative) signal is composed of a sum of positive contributions: **combine parts to form a whole.**



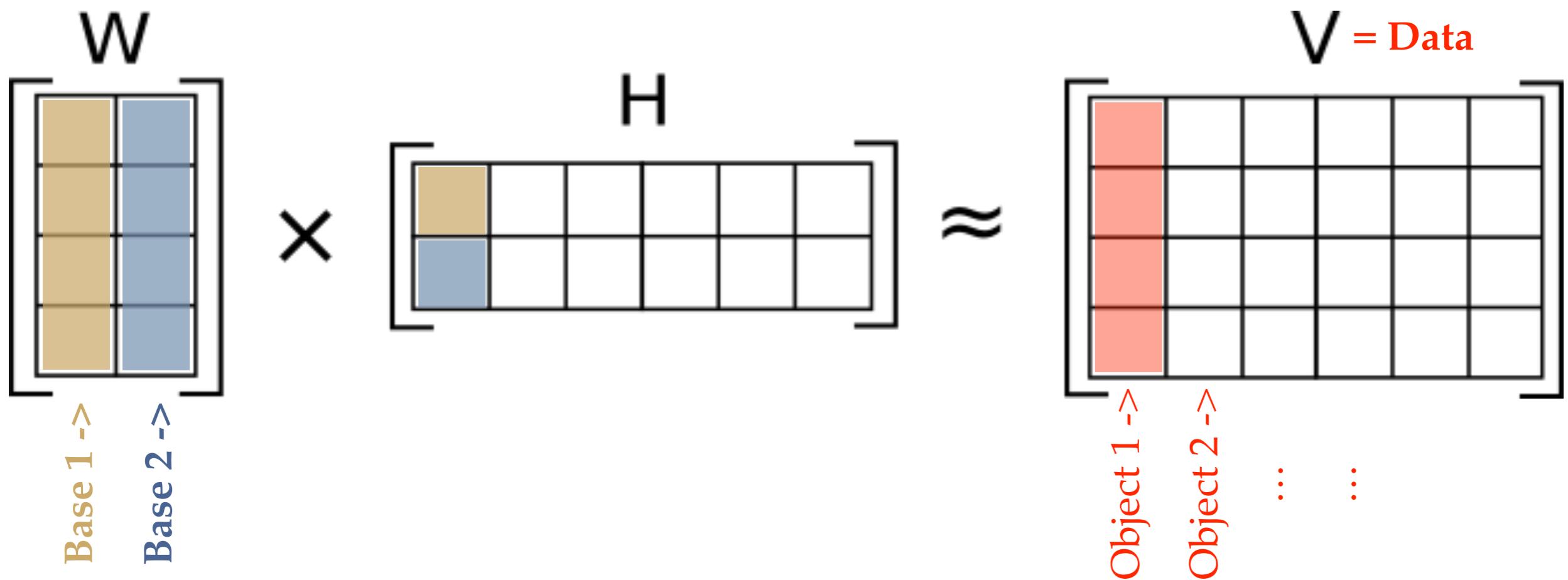
# Non-Negative Matrix Factorization (NMF)

- ❖ In non-negative matrix factorization, a large matrix  $V$  into (usually much smaller) matrices  $W$  and  $H$ , where all three matrices have no negative elements.
  - ❖ By applying such a decomposition, we essentially assume that the (non-negative) signal is composed of a sum of positive contributions: **combine parts to form a whole.**



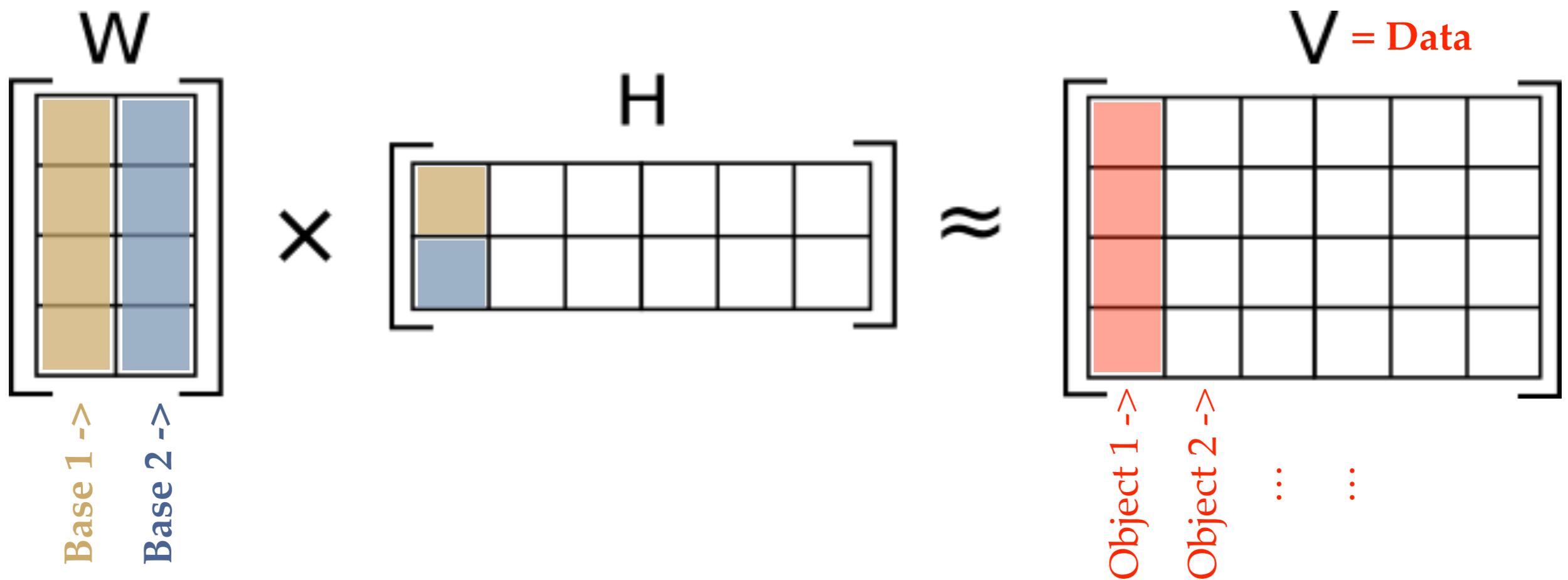
# Non-Negative Matrix Factorization (NMF)

- ❖ In non-negative matrix factorization, a large matrix  $V$  into (usually much smaller) matrices  $W$  and  $H$ , where all three matrices have no negative elements.
  - ❖ By applying such a decomposition, we essentially assume that the (non-negative) signal is composed of a sum of positive contributions: **combine parts to form a whole.**



# Non-Negative Matrix Factorization (NMF)

- ❖ In non-negative matrix factorization, a large matrix  $V$  into (usually much smaller) matrices  $W$  and  $H$ , where all three matrices have no negative elements.
  - ❖ By applying such a decomposition, we essentially assume that the (non-negative) signal is composed of a sum of positive contributions: **combine parts to form a whole.**



# NMF: notes

---

- ❖ There is no analytical solution for  $W \geq 0, H \geq 0$  that satisfy  $W \times H \approx V$ , but there exist several numerical methods to do that.
- ❖ Assuming underlying structure to the data, NMF constructs sparse bases and sparse weightings.

# Learning the parts of objects by nonnegative matrix factorization

1999

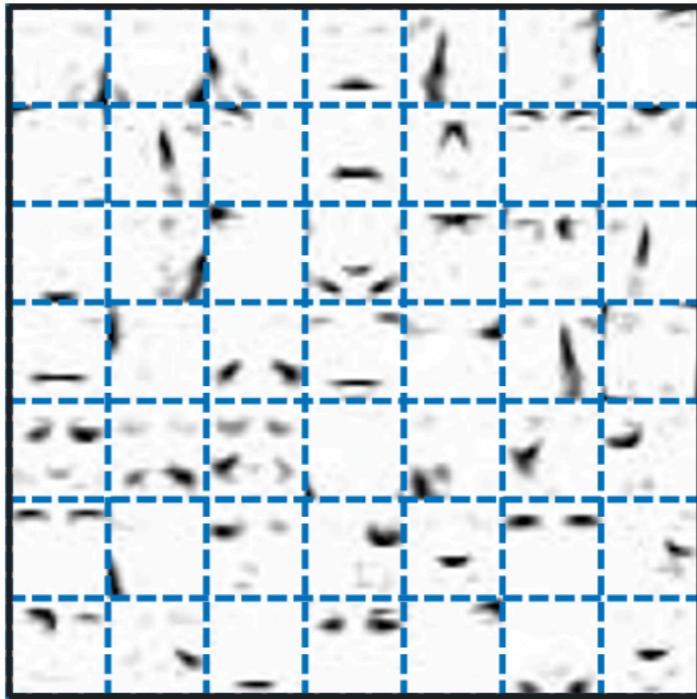
D. D. Lee\*

\*Bell Laboratories  
Lucent Technologies  
Murray Hill, NJ 07974

H. S. Seung\*,†

†Dept. of Brain and Cognitive Sciences  
Massachusetts Institute of Technology  
Cambridge, MA 02139

NMF



Original



$$\times \begin{matrix} & \\ & \end{matrix} = \begin{matrix} & \\ & \end{matrix}$$

VQ

$$\times \begin{matrix} & \\ & \end{matrix}$$

PCA

$$\times \begin{matrix} & \\ & \end{matrix}$$

# Learning the parts of objects by nonnegative matrix factorization

1999

D. D. Lee\*

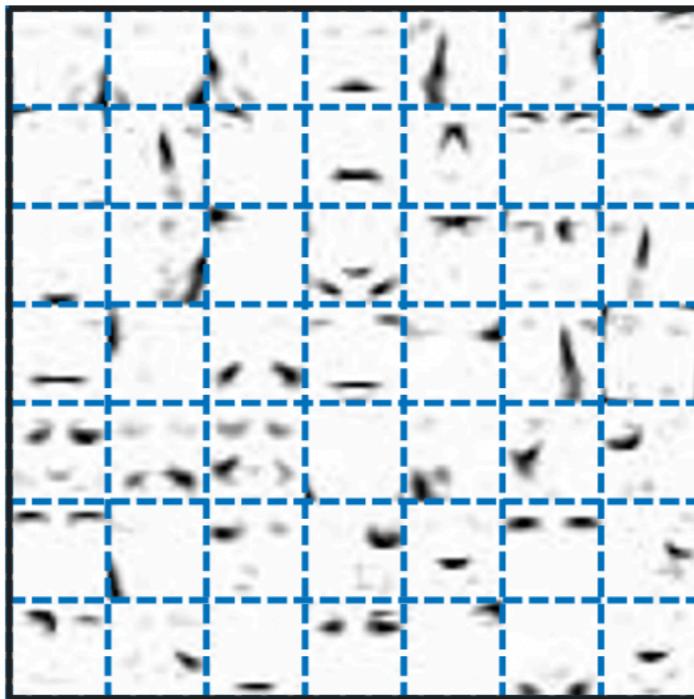
\*Bell Laboratories  
Lucent Technologies  
Murray Hill, NJ 07974

H. S. Seung\*,†

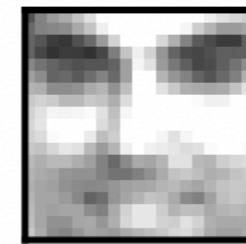
†Dept. of Brain and Cognitive Sciences  
Massachusetts Institute of Technology  
Cambridge, MA 02139

Parts of faces ->

NMF



Original



$$\times \begin{matrix} & \\ & \end{matrix} = \begin{matrix} & \\ & \end{matrix}$$

For NMF, most of  
the bases contain  
numerous empty  
spaces,  
suggesting that  
this is a sparse  
representation

VQ

$$\begin{matrix} & \\ & \end{matrix} \times \begin{matrix} & \\ & \end{matrix} \parallel \begin{matrix} & \\ & \end{matrix}$$

PCA

$$\begin{matrix} & \\ & \end{matrix} \times \begin{matrix} & \\ & \end{matrix} \parallel \begin{matrix} & \\ & \end{matrix}$$

PCA has dense  
matrices  
containing  
positive and  
negative values.

- 
- ❖ To the Jupyter notebook!