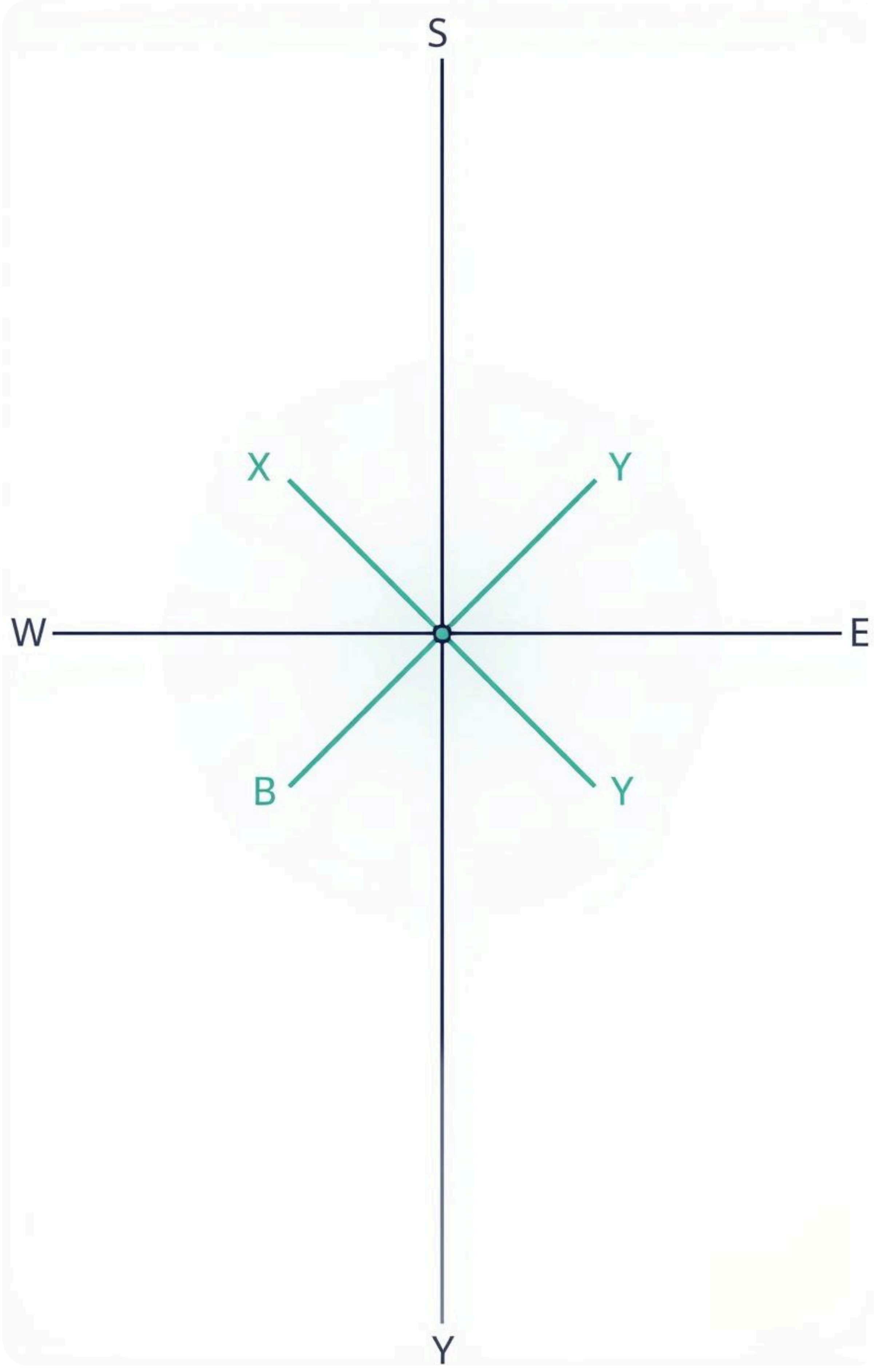
Principal Component Analysis

Jonathan Shlens (Google Research)
2014



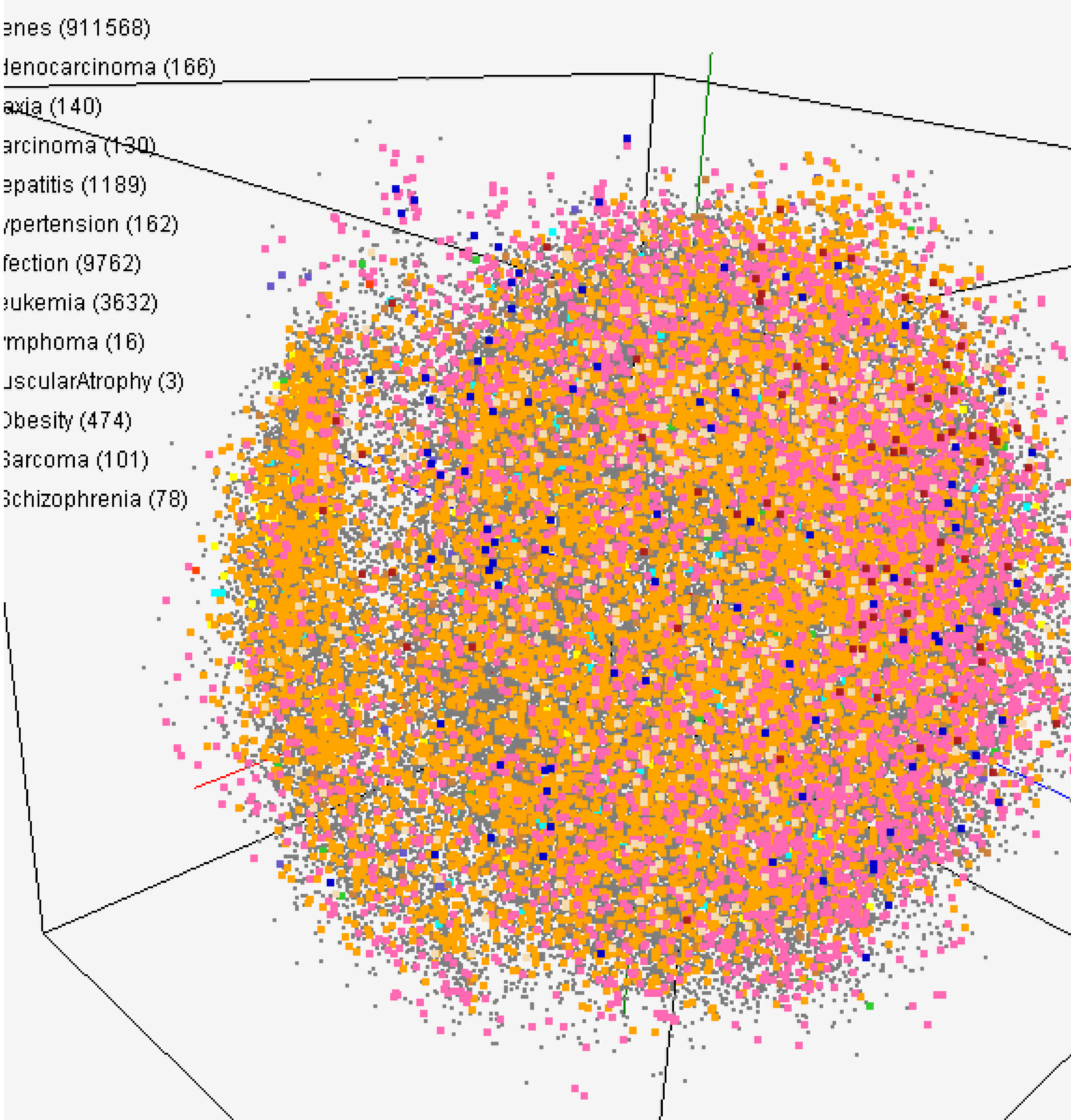
Why Was This Problem Important (Historically)?

Why this problem mattered

- Growth of data across domains:
  - Neuroscience (multi-electrode recordings)
  - Signal processing
  - Computer vision
  - Climate & physical systems
- Data dimensionality increased faster than interpretability

Challenges at the time

- Raw high-dimensional data:
  - Was difficult to visualize
  - Contained large amounts of redundancy
  - Obscured the true degrees of freedom
- Many users applied PCA as a black-box tool without understanding:
  - What assumptions it makes
  - When it works
  - When it fails



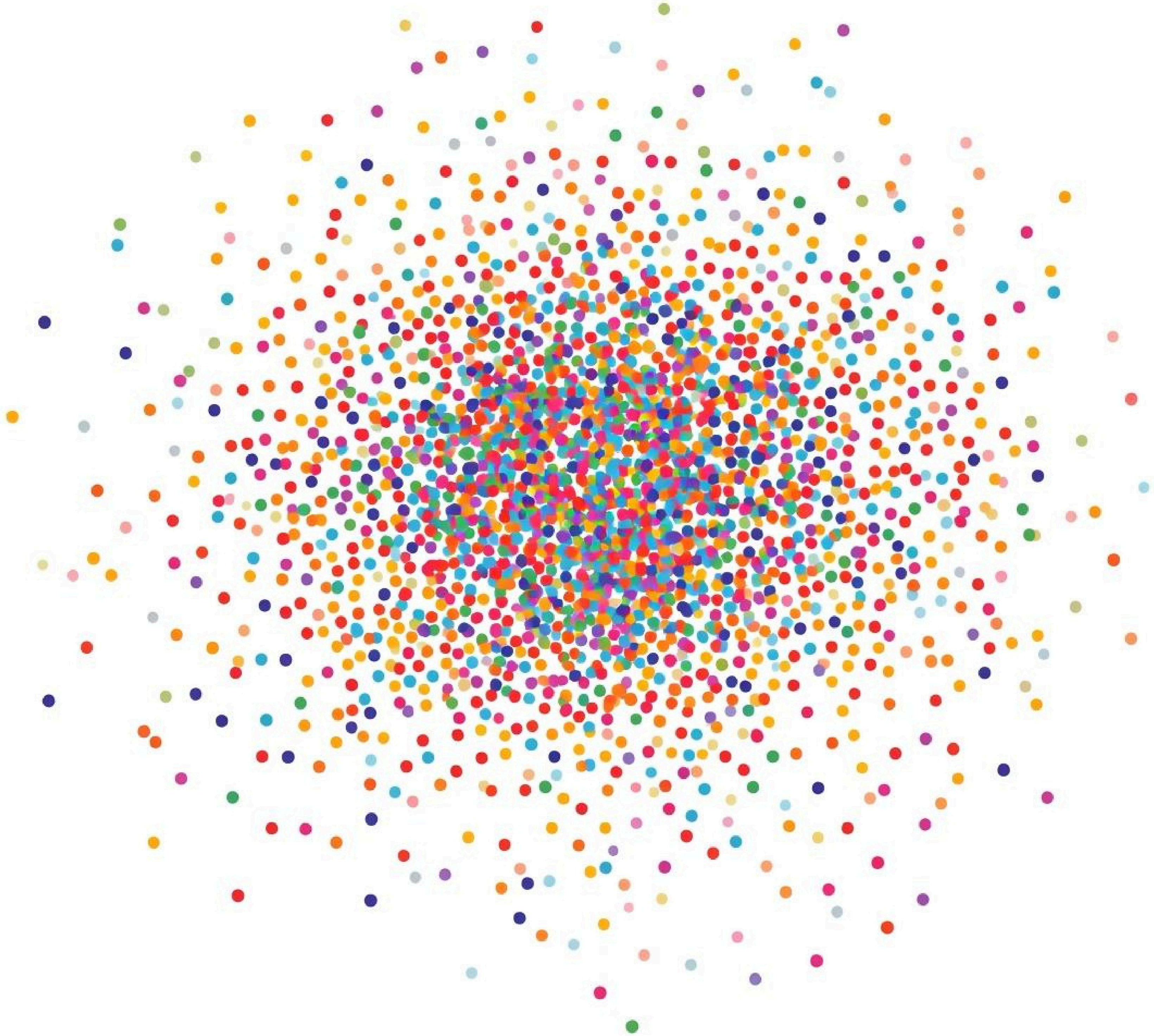
High-Dimensional Data Challenges

What problem does the paper address?

- Real-world datasets are often high-dimensional, noisy, and redundant
- Measurements are usually taken in sensor-defined coordinates, not in coordinates aligned with the true underlying system
- This makes it difficult to:
  - Understand the structure of the data
  - Identify which variables are actually important
  - Reduce dimensionality without losing information

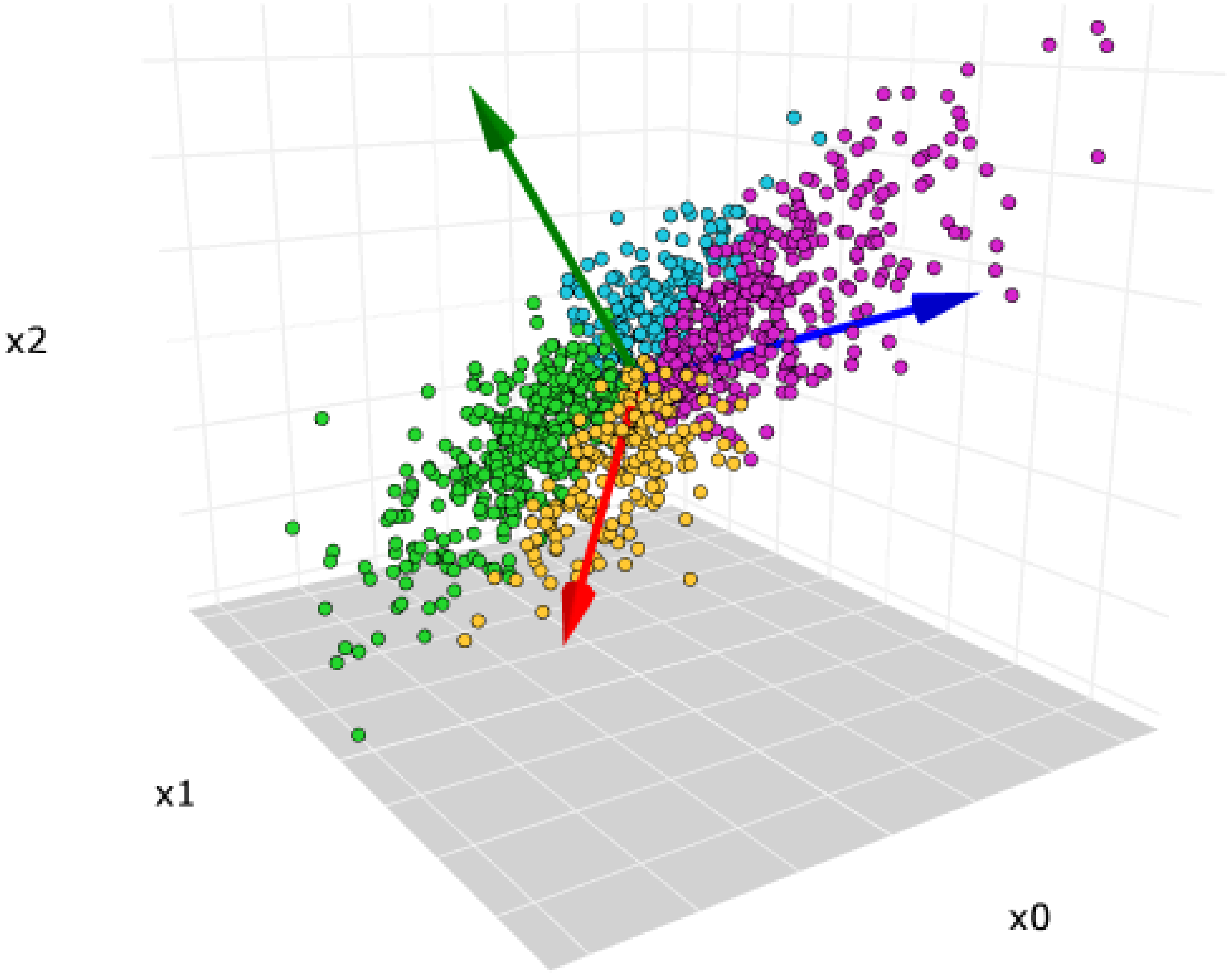
Core question addressed by the paper:

How can we systematically re-express high-dimensional data in a more meaningful coordinate system that reveals its underlying structure?

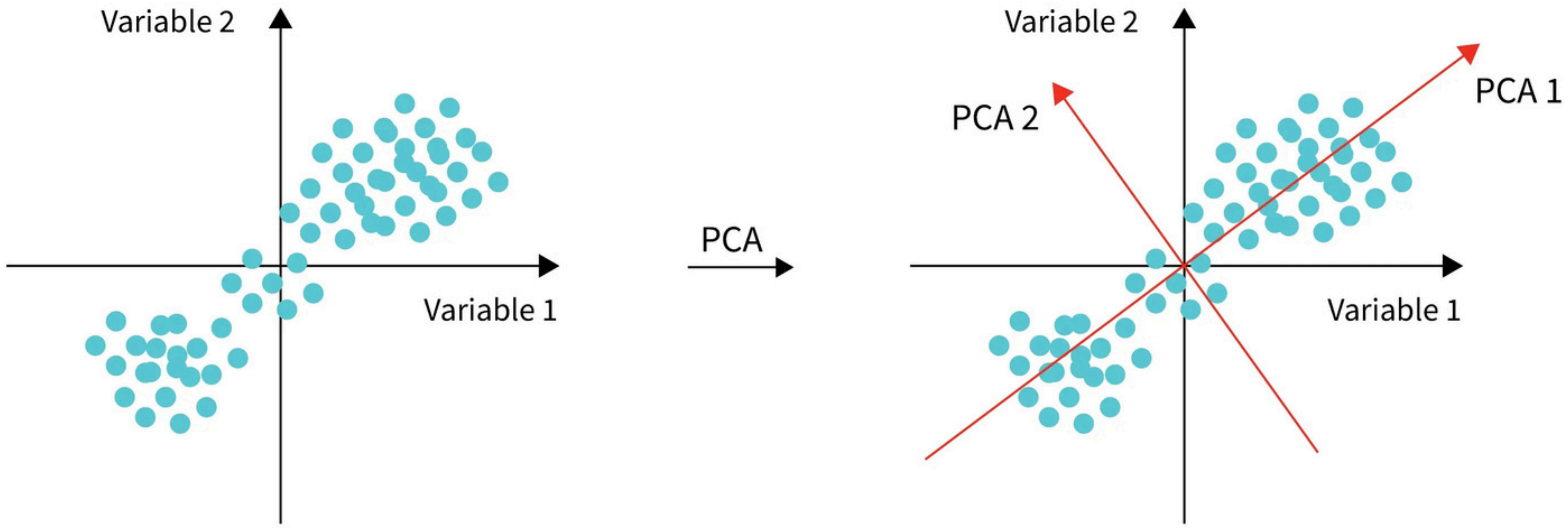


Limitations and gaps

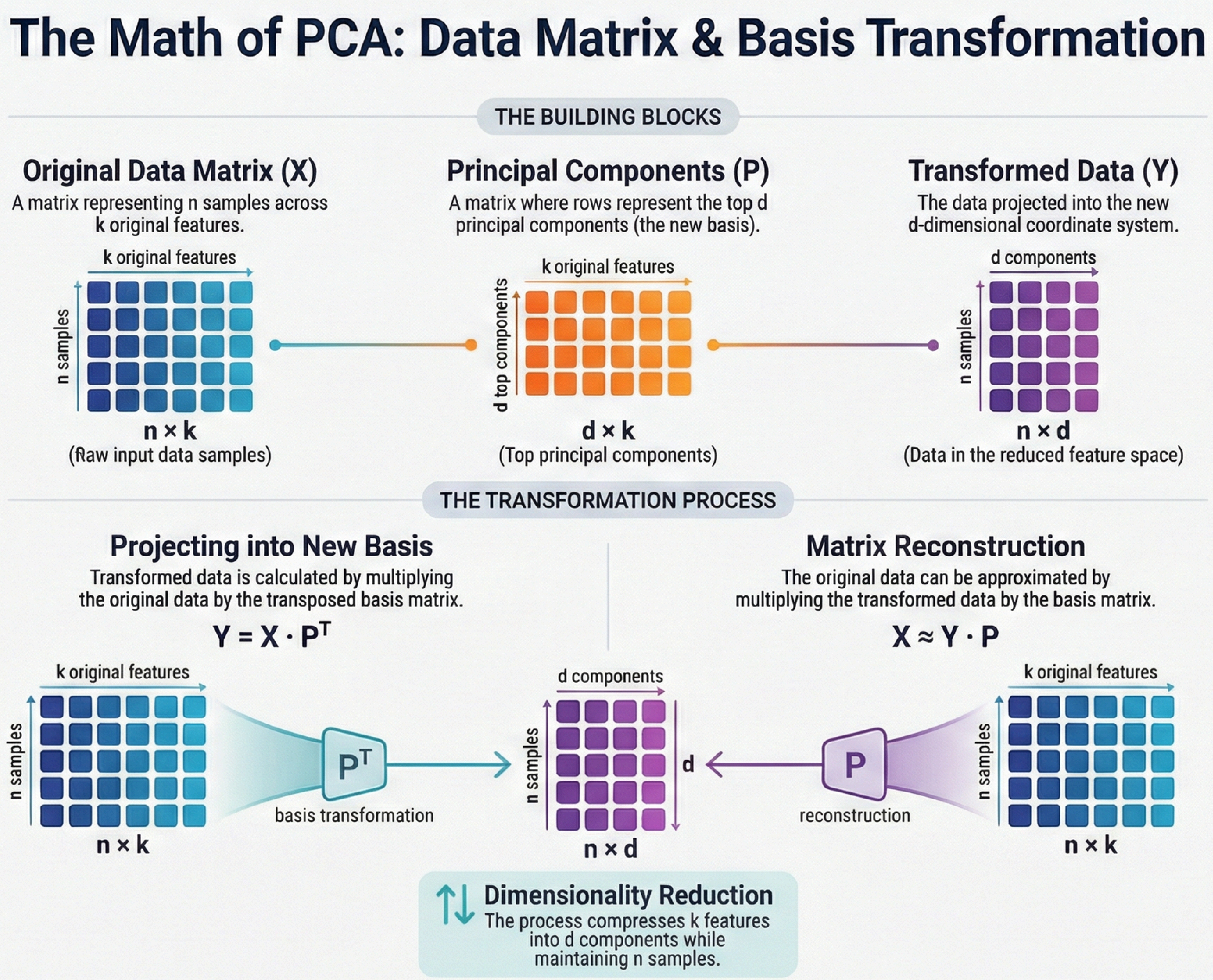
Before PCA, methods relied heavily on **manual feature extraction** and heuristics, leading to suboptimal solutions with poor scalability, lack of optimality, and no principled approaches to dimensionality reduction.



Core Idea: Rotate the Basis

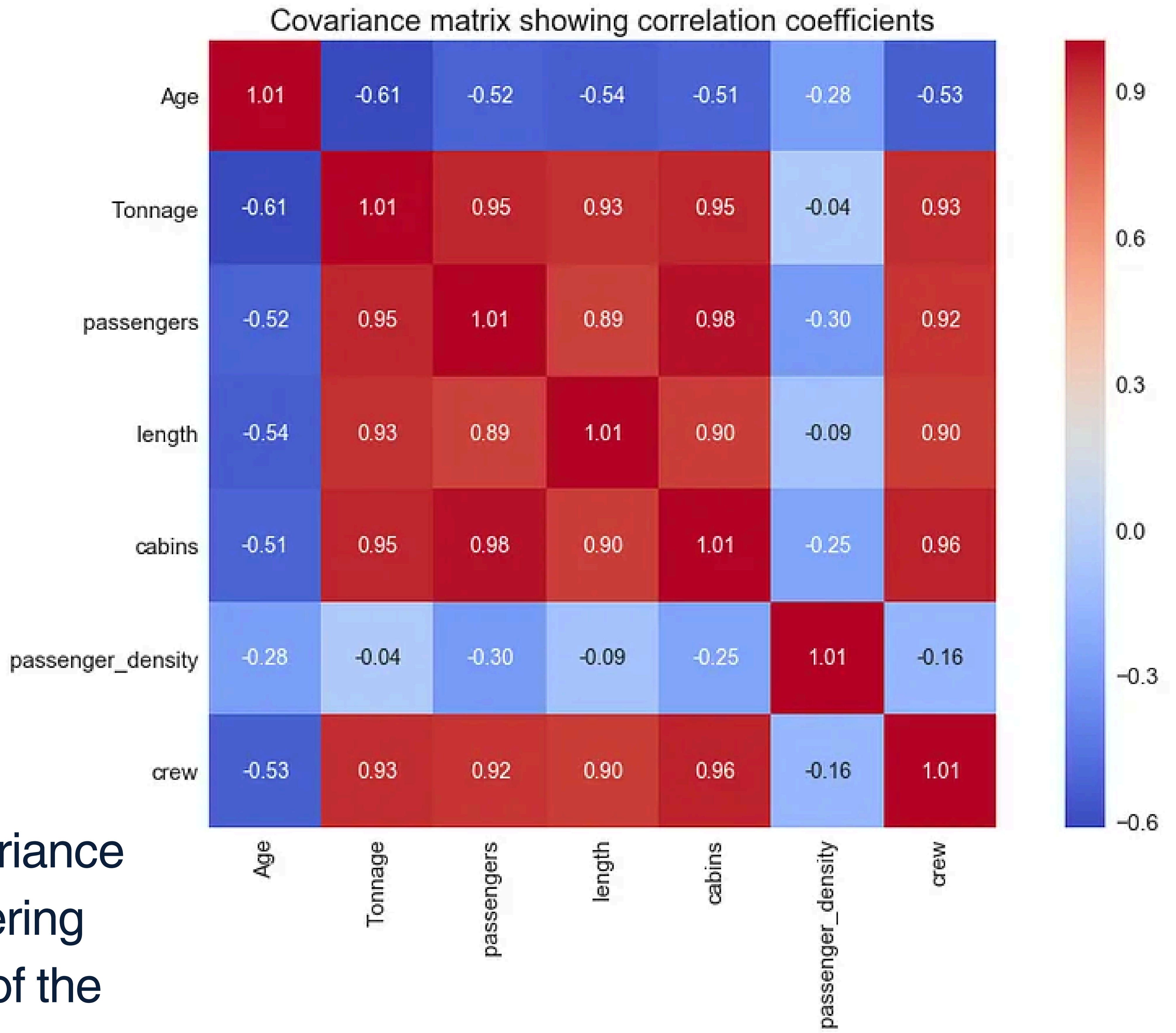


The core concept of PCA is to **rotate the coordinate system**, transforming the original data into a new basis that aligns with the maximum variance, effectively reducing dimensions while capturing essential information.



The Math of PCA: Data Matrix & Basis Transformation
THE BUILDING BLOCKS
Original Data Matrix (X)
A matrix representing n samples across k original features.
k original features
n samples
n × k
(Raw input data samples)
Principal Components (P)
A matrix where rows represent the top d principal components (the new basis).
k original features
d top components
d × k
(Top principal components)
Transformed Data (Y)
The data projected into the new d-dimensional coordinate system.
d components
n samples
n × d
(Data in the reduced feature space)
THE TRANSFORMATION PROCESS
Projecting into New Basis
Transformed data is calculated by multiplying the original data by the transposed basis matrix.
$Y = X \cdot P^T$
k original features
n samples
n × k
P^T
basis transformation
d components
n samples
d
n × d
Matrix Reconstruction
The original data can be approximated by multiplying the transformed data by the basis matrix.
$X \approx Y \cdot P$
P
reconstruction
k original features
n samples
n × k
Dimensionality Reduction
The process compresses k features into d components while maintaining n samples.

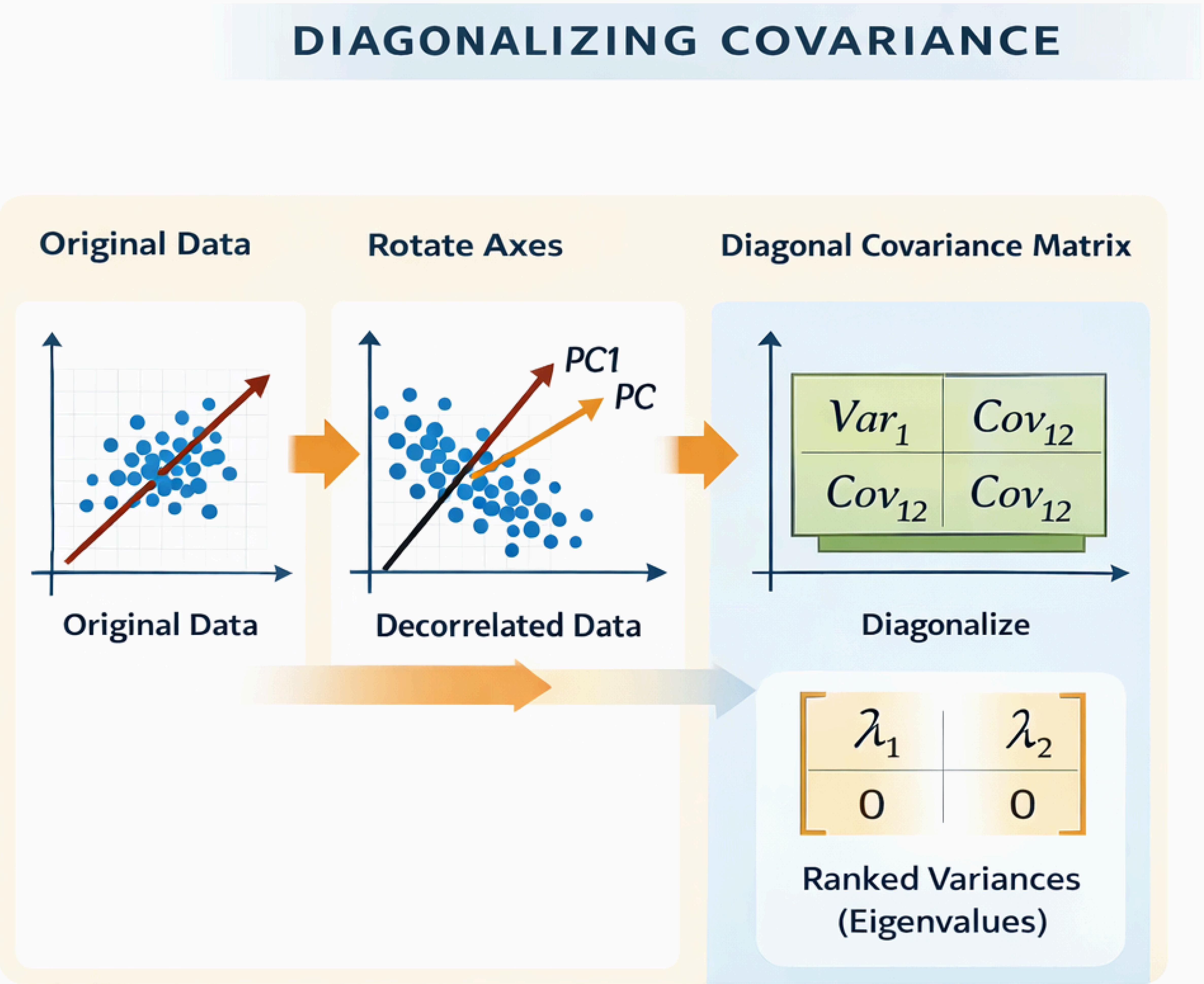
Understanding Variance and Covariance



Variance measures the spread of data, while covariance indicates the redundancy between features. Centering the data (zero means) allows for the computation of the covariance matrix ($C = (1/n)XX^{T}$), highlighting relationships through both diagonal and off-diagonal elements.

Objective: Diagonalizing Covariance

The goal is to **remove correlation** between variables, reduce redundancy, achieving decorrelation while maximizing variance. This process results in diagonalized covariance matrices, allowing for ranked interpretation of eigenvalues and principal components.

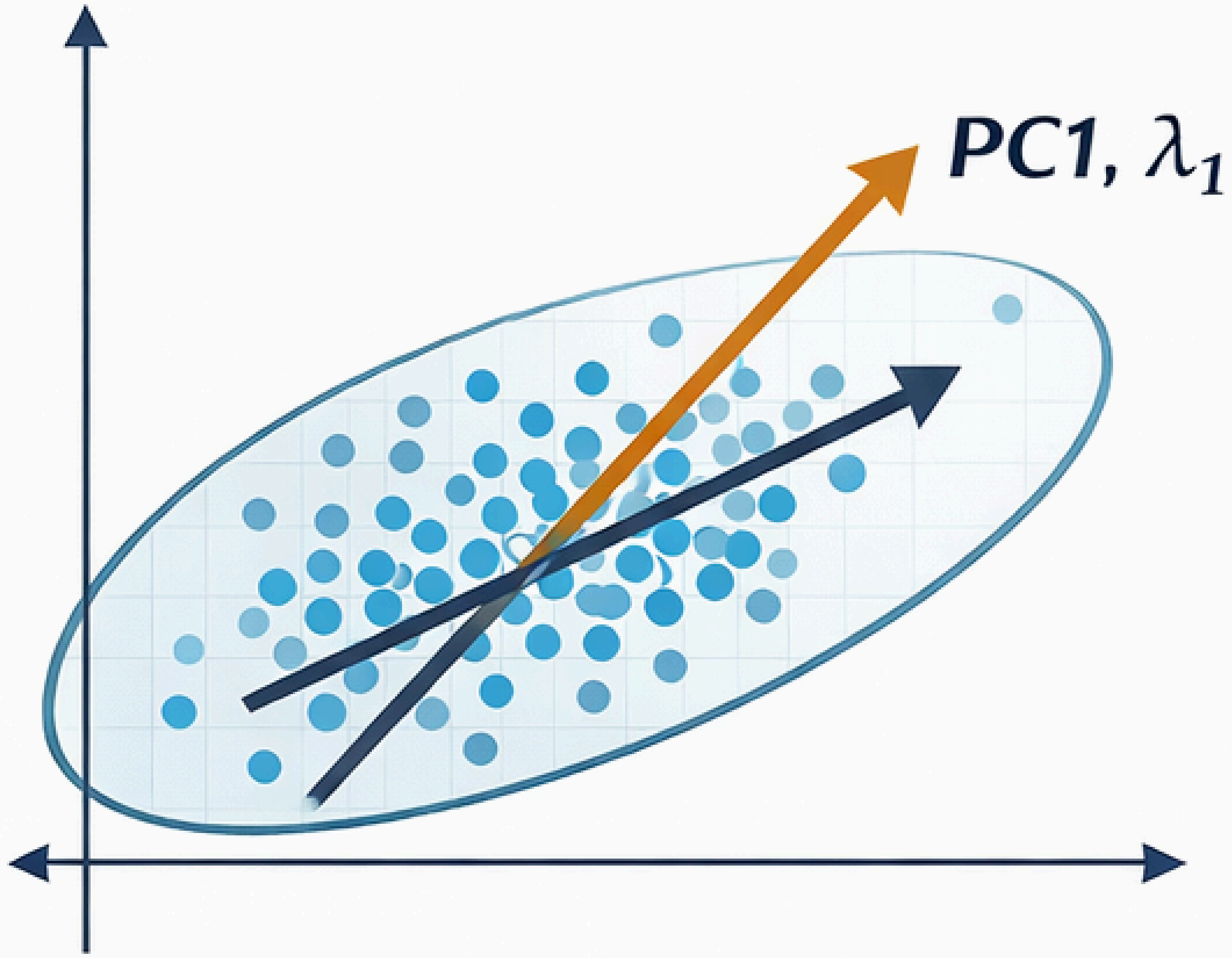


Eigenvectors Define Principal Directions

The covariance matrix C is symmetric and diagonalizable, expressed as C = $E\Lambda E^T$.

The columns of matrix E represent the eigenvectors, while eigenvalues indicate the variance captured by each direction.

$$C = E\Lambda E^T, \lambda_1 \geq \lambda_2$$

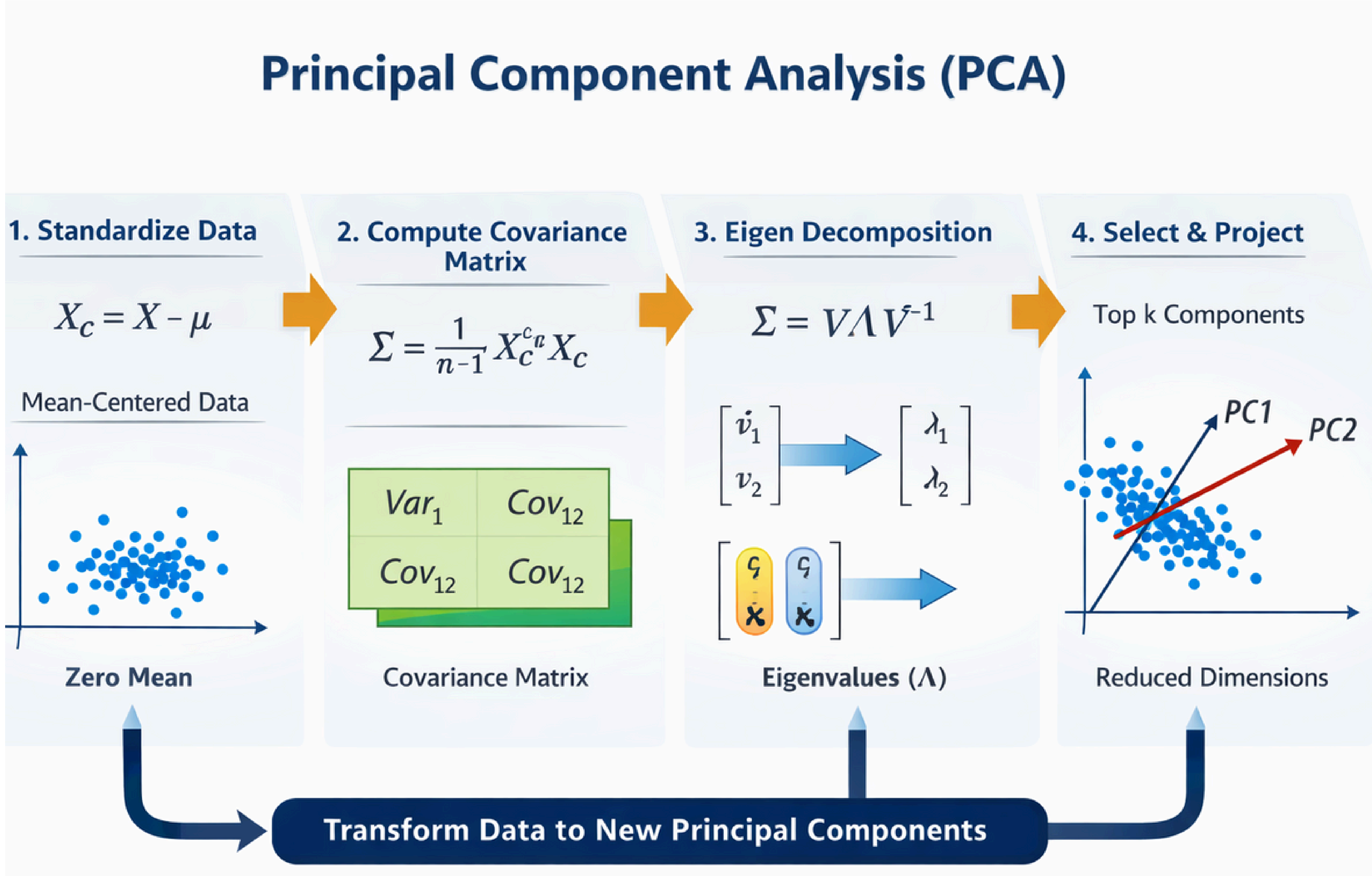


$$C = E\Lambda E^T$$

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \rightarrow \begin{bmatrix} \lambda_1 & \lambda_2 \\ 0 & 0 \end{bmatrix}$$

$$C = E\Lambda E^T$$

Implementation Steps for PCA



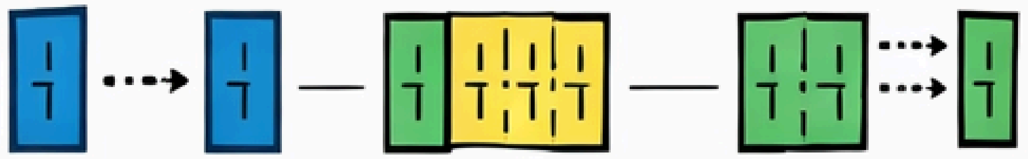
To implement PCA, we first **center our data** by subtracting the mean, compute the covariance matrix, perform eigen decomposition or SVD, sort the components, project the data, and select the desired number of principal components.

Singular Value Decomposition (SVD)

and its Connection to Eigenvalue Decomposition

SVD

$$A = U \Sigma V^T$$



$m \times n$ $m \times r$ $r \times r$

- **U**: Left Singular Vectors
- **Σ**: Diagonal Matrix of **Singular Values**
- **V^T**: Transpose of Right Singular Vectors

Eigenvalue Decomposition

$$A A^T = U \Lambda U^T$$

$n \times n$ $n \times r$ $n \times r$

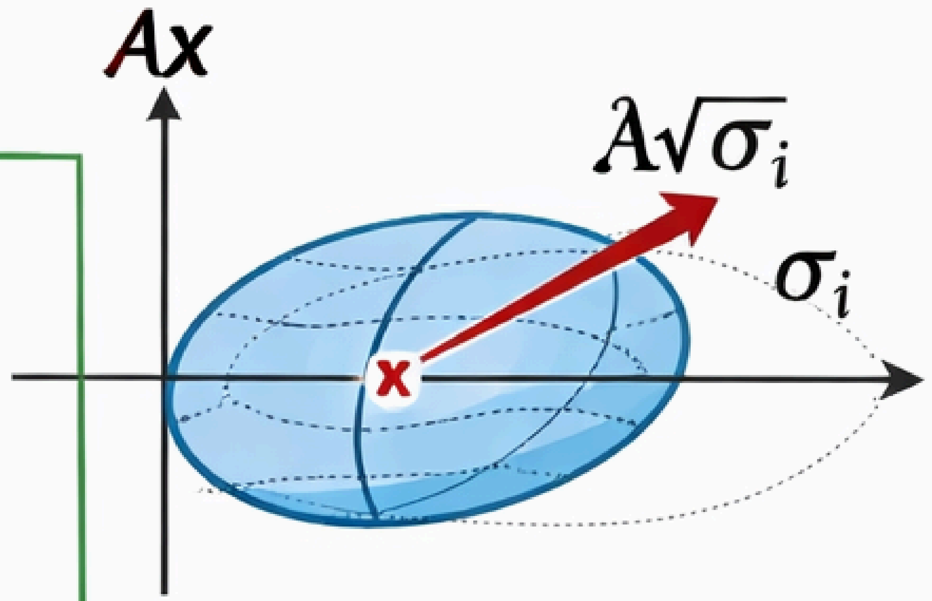
- **U**: Eigenvectors of A^{2T}
- **Λ**: Diagonal Matrix of Eigenvalues

Connection

SVD: $A = U \Sigma V^T$ → *Singular Values (Σ) are the square roots of Eigenvalues (Λ)*

Eigenvalue Decomp:

$A A^T = U \Lambda U^T$

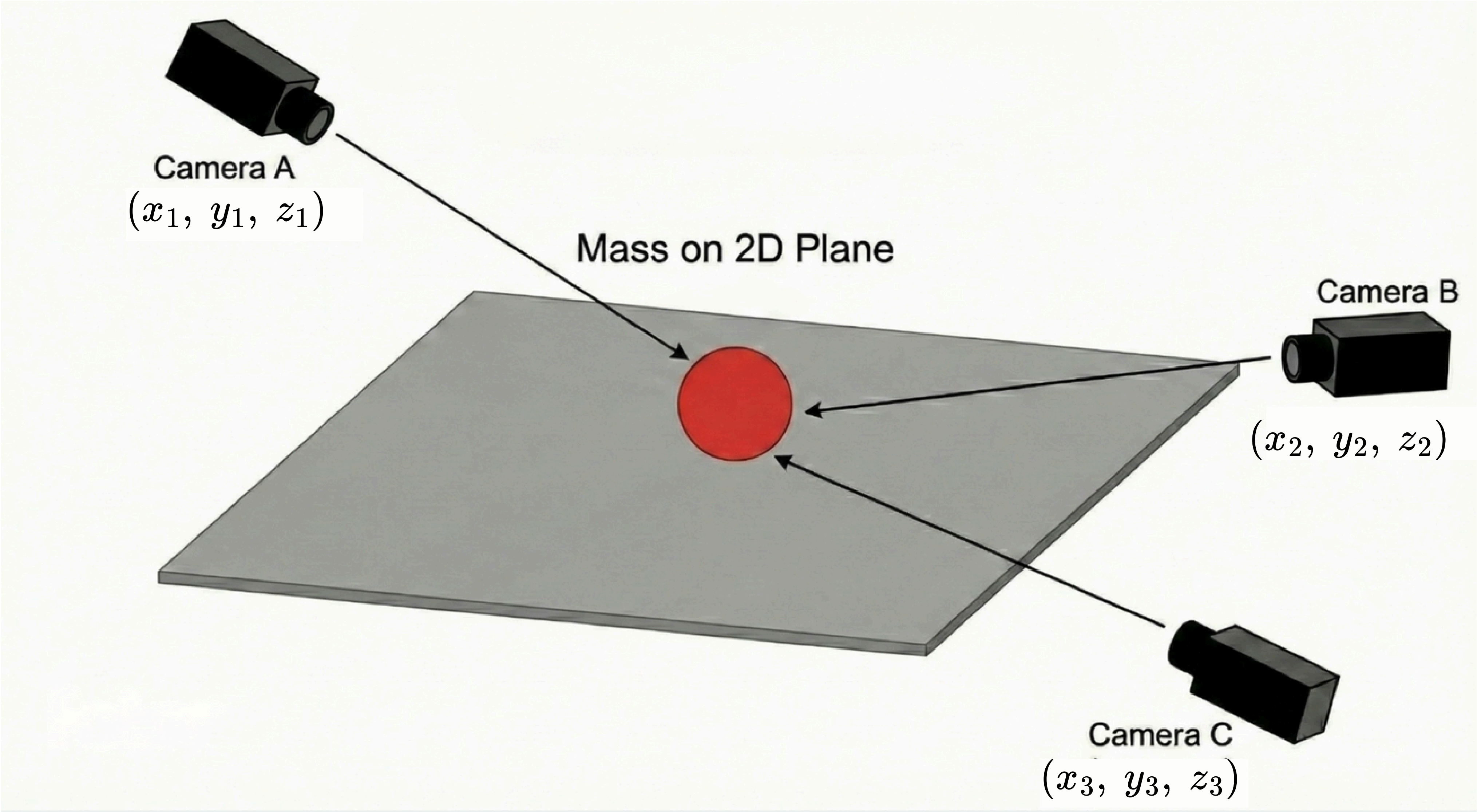


Assumptions and Design Choices

The PCA method relies on **linearity and orthogonality** of features, where variance represents information. It assumes mean-centering of data and Gaussian noise, ensuring effective dimensionality reduction and accurate results.

Experimental Setup

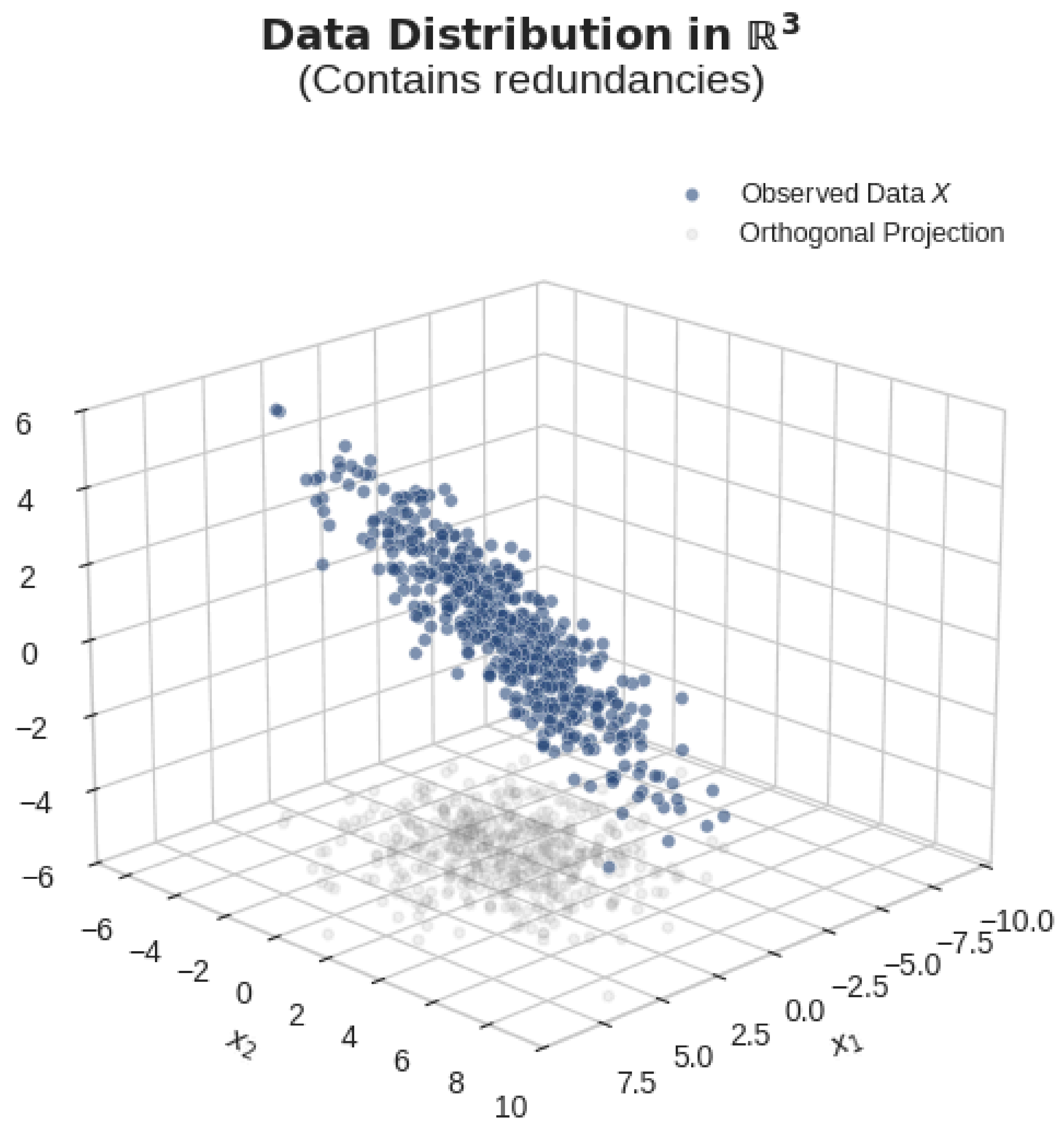
To demonstrate the core concepts of the paper, we simulate a system where three sensors measure a phenomenon that inherently has only two degrees of freedom.



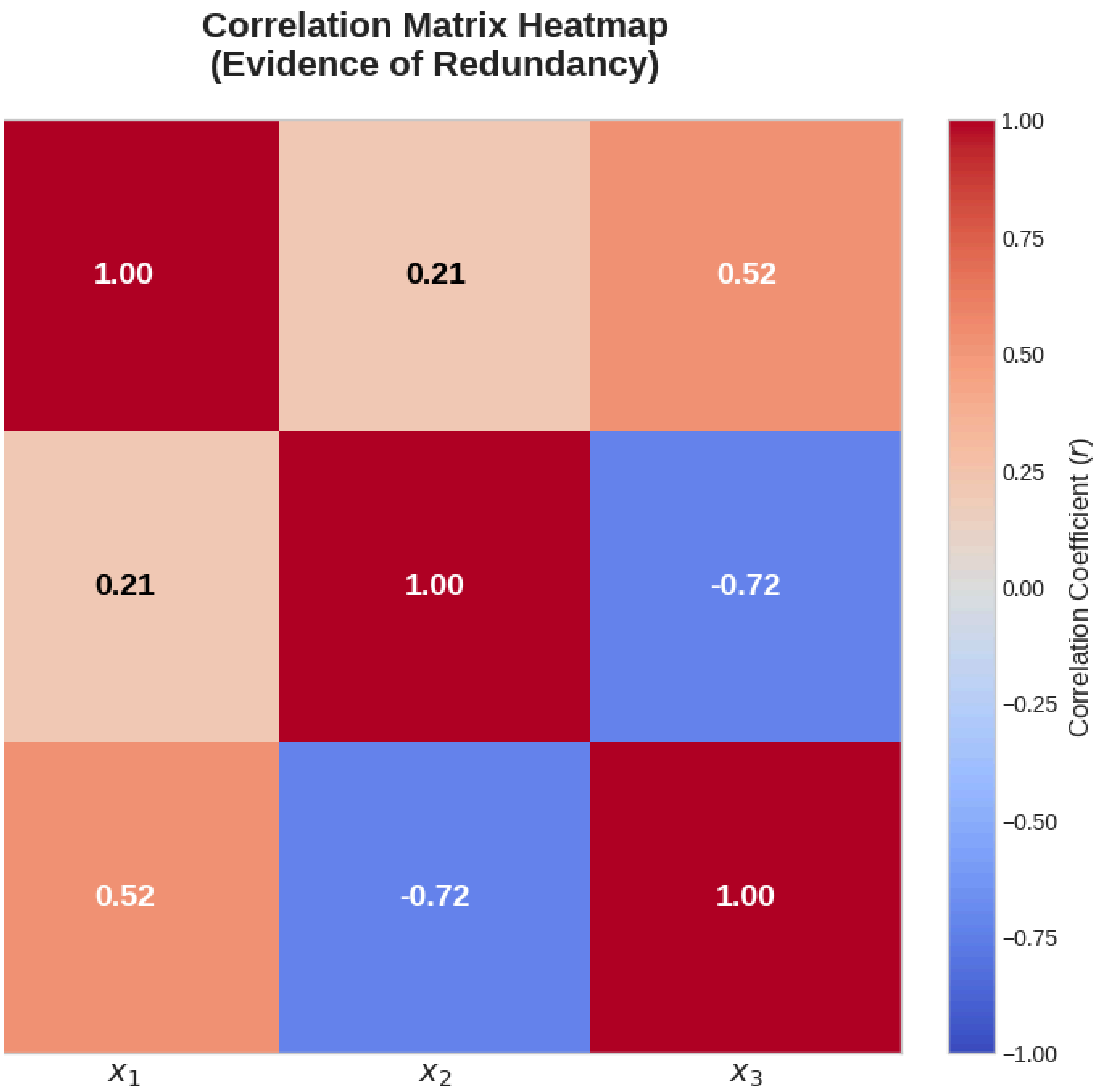
Experimental Setup

The table below shows the raw, correlated data points that serve as the input for our PCA analysis.

Sensor 1 (x_1)	Sensor 2 (x_2)	Sensor 3 (x_3)
1.1661	-0.5374	0.8755
3.3174	3.3365	-0.3598
-0.7534	-0.3579	-0.0356
4.6575	1.0261	1.6979
-0.2681	1.5149	-1.0189
-1.6513	-0.9048	-0.3244
-1.7473	-4.6946	2.1372
-4.8719	-0.317	-1.8671
-1.8809	1.4055	-1.7983
-3.862	-2.7834	-0.1932
⋮	⋮	⋮



Observations exist in a 3-dimensional space ($\mathbb{R}^3$), but clearly collapse onto a flat 2D plane.



The correlation matrix confirms the linear dependence observed in the 3D plot. The strong negative correlation (blue regions) proves that the variables are coupled, meaning the standard basis is an inefficient way to represent this data.

Process Overview

Standardize Data

$$X_c = X - \mu$$

Covariance Matrix

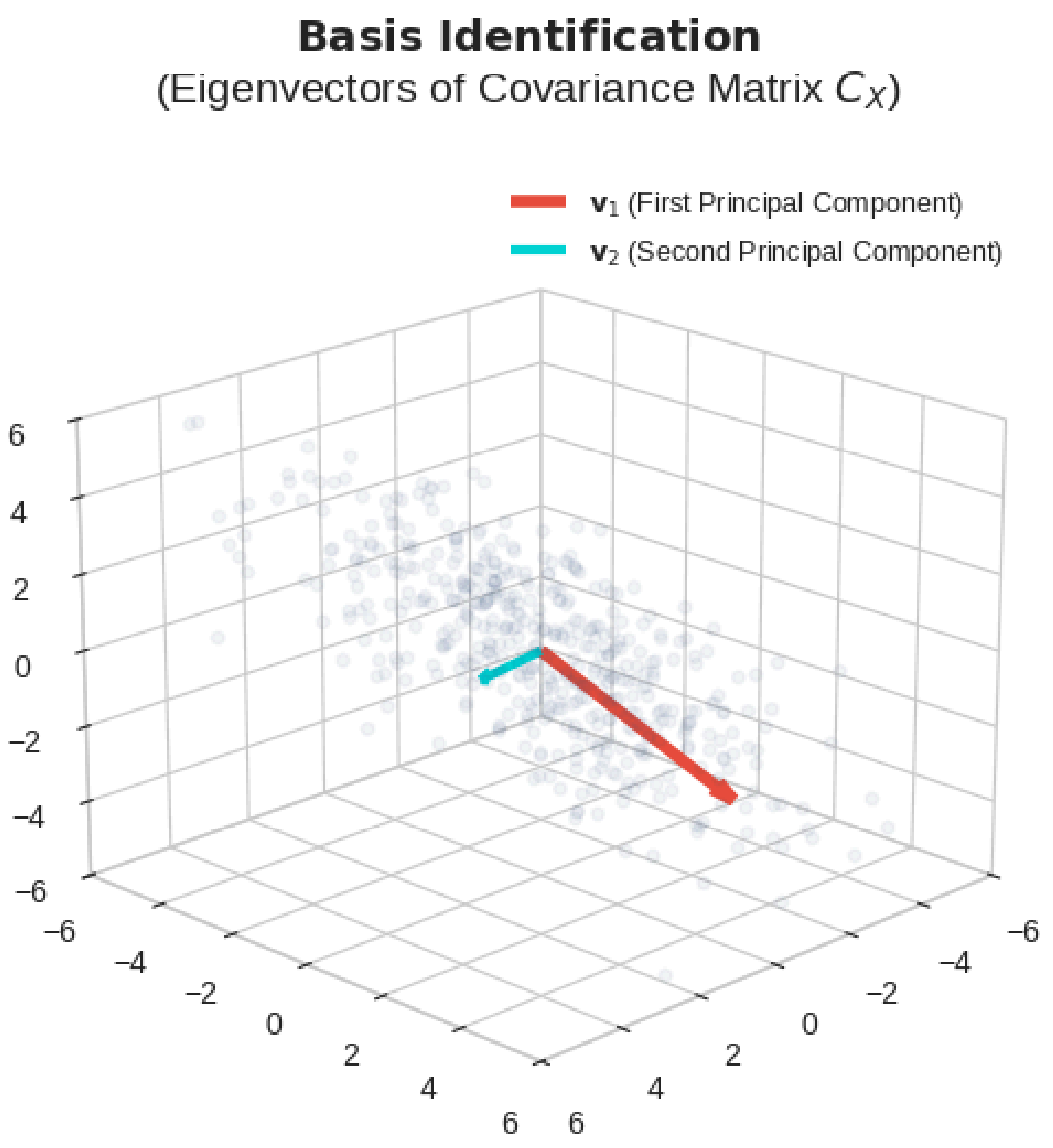
$$C_x = \frac{1}{n-1} X X^T$$

Eigen Decomposition

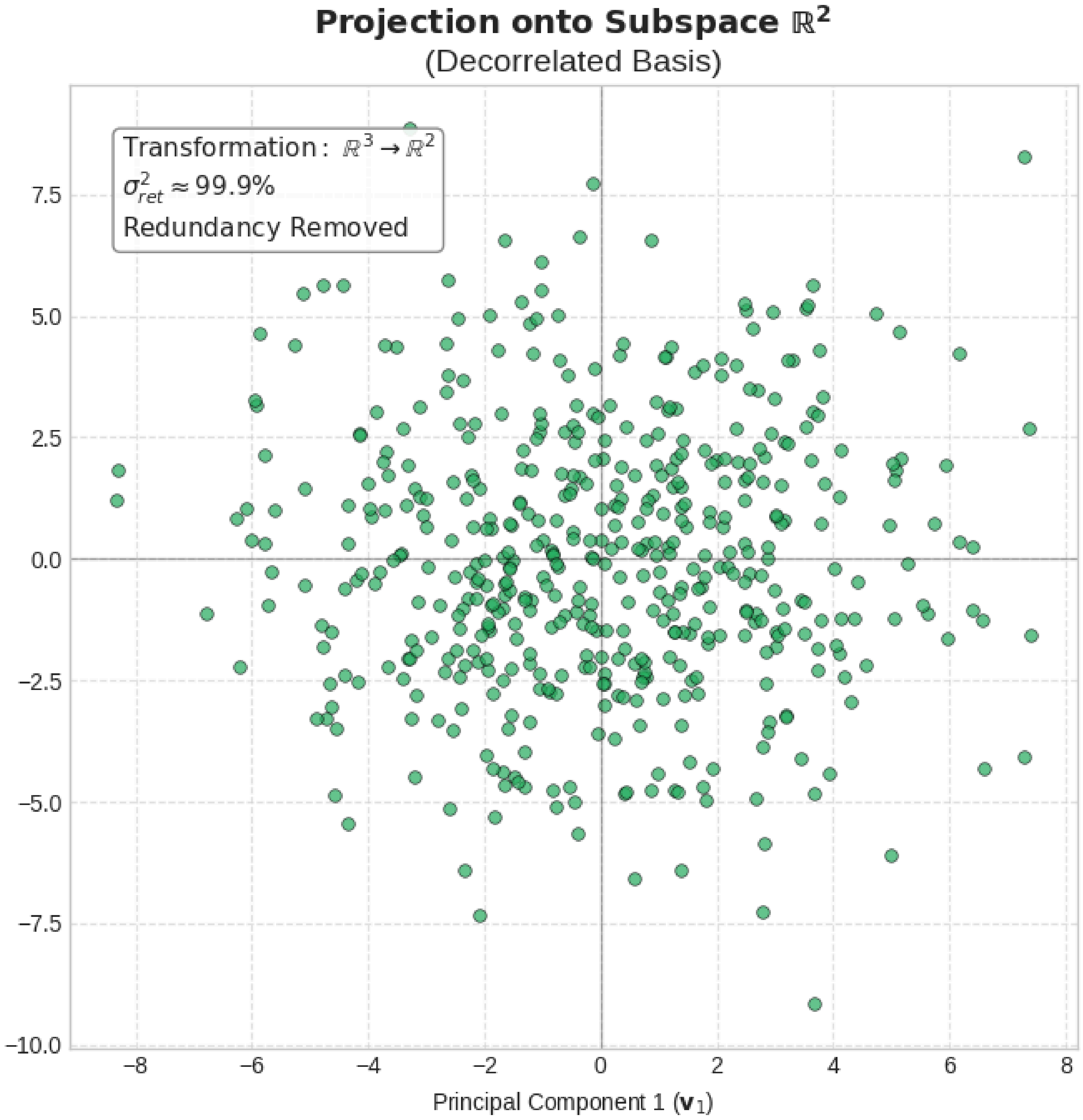
$$C_x \mathbf{v} = \lambda \mathbf{v}$$

Select and Project

$$Y = P^T X_c$$



PCA identifies the 'Principal Axes' (arrows) that define this plane. The red vector ($\mathbf{v}_1$) aligns with the maximum variance (signal), while the orthogonal cyan vector ($\mathbf{v}_2$) captures the remaining structure.



By projecting the data onto this new basis, we reduce the system from 3D to 2D. We have successfully filtered out the noise and decorrelated the variables while retaining >99% of the original information.

Limitations of Principal Component Analysis

While PCA is powerful, it struggles with **nonlinear patterns**, is sensitive to outliers, and is limited to second-order statistics, potentially hindering interpretability and effectiveness in complex datasets.

Impact and Legacy of PCA

Principal Component Analysis has become a **standard preprocessing technique** across various fields, including computer vision and neuroscience. PCA is the default choice when we need speed, interpretability, stability, and linear structure preservation. Nonlinear methods are mainly for visualization, not general-purpose preprocessing.

Timeline

Impact of PCA

Historical influence on data science and ML

Karl Pearson's Introduction

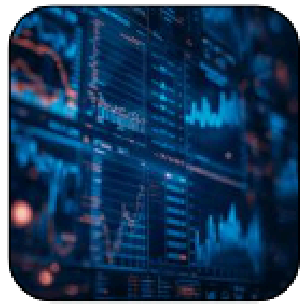
In 1901, Karl Pearson introduced the concept of principal axes, laying the groundwork for PCA.

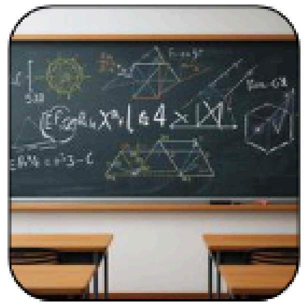
Data Compression

In the 1990s, PCA was utilized for data compression and visualization techniques across various fields.

Hotelling's Formalization

In 1933, Harold Hotelling mathematically formalized PCA, establishing its robust theoretical foundation.



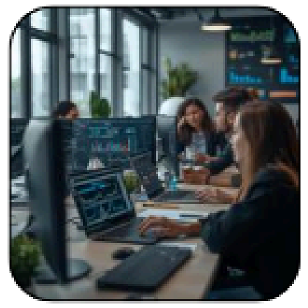


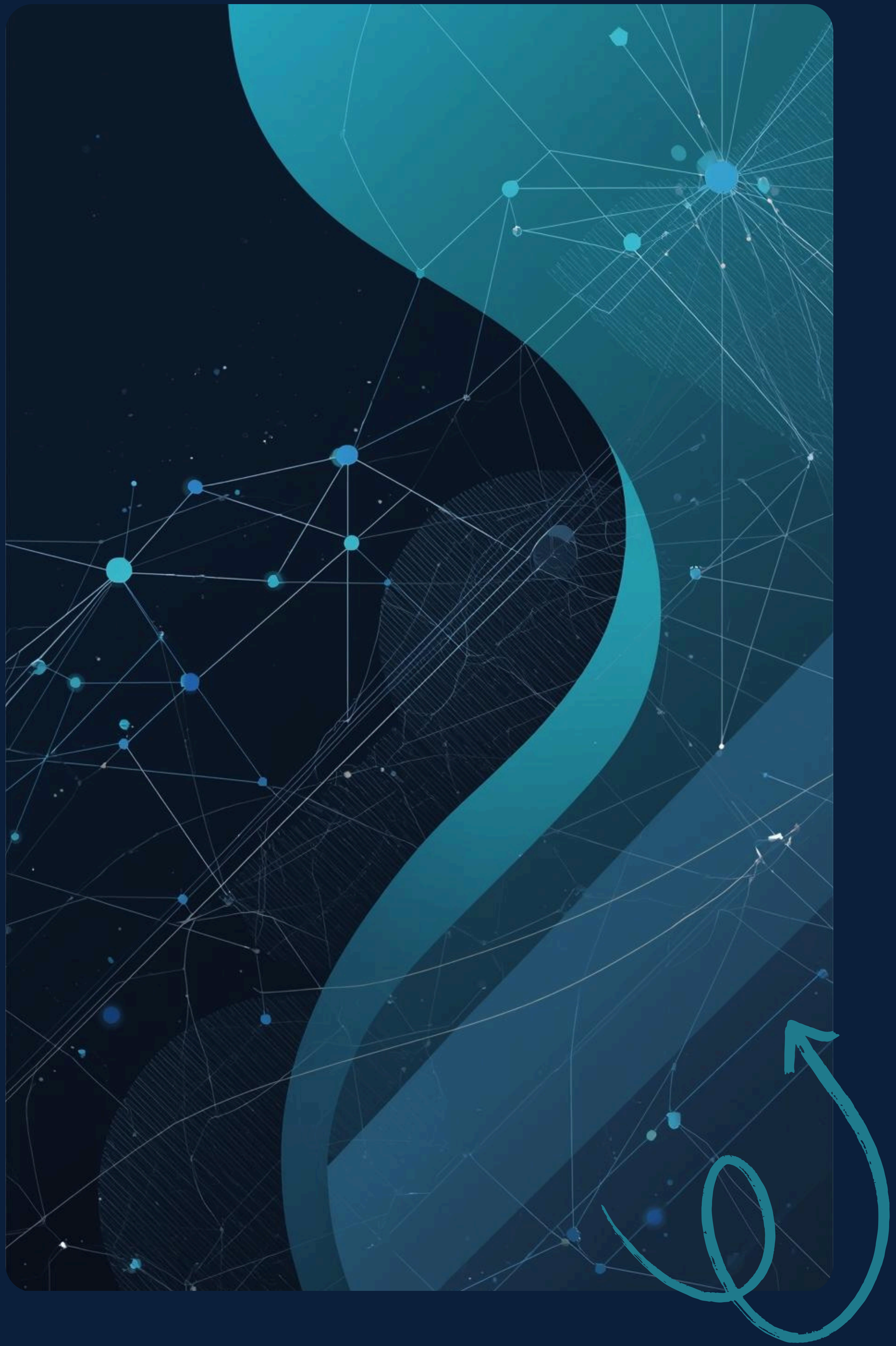
Standard in ML

By the 2000s, PCA became a standard preprocessing step in machine learning workflows and applications.

Widespread Statistical Use

From the 1960s to the 1980s, PCA became widely employed in statistics and signal processing methodologies.





Conclusion

PCA is foundational for data analysis