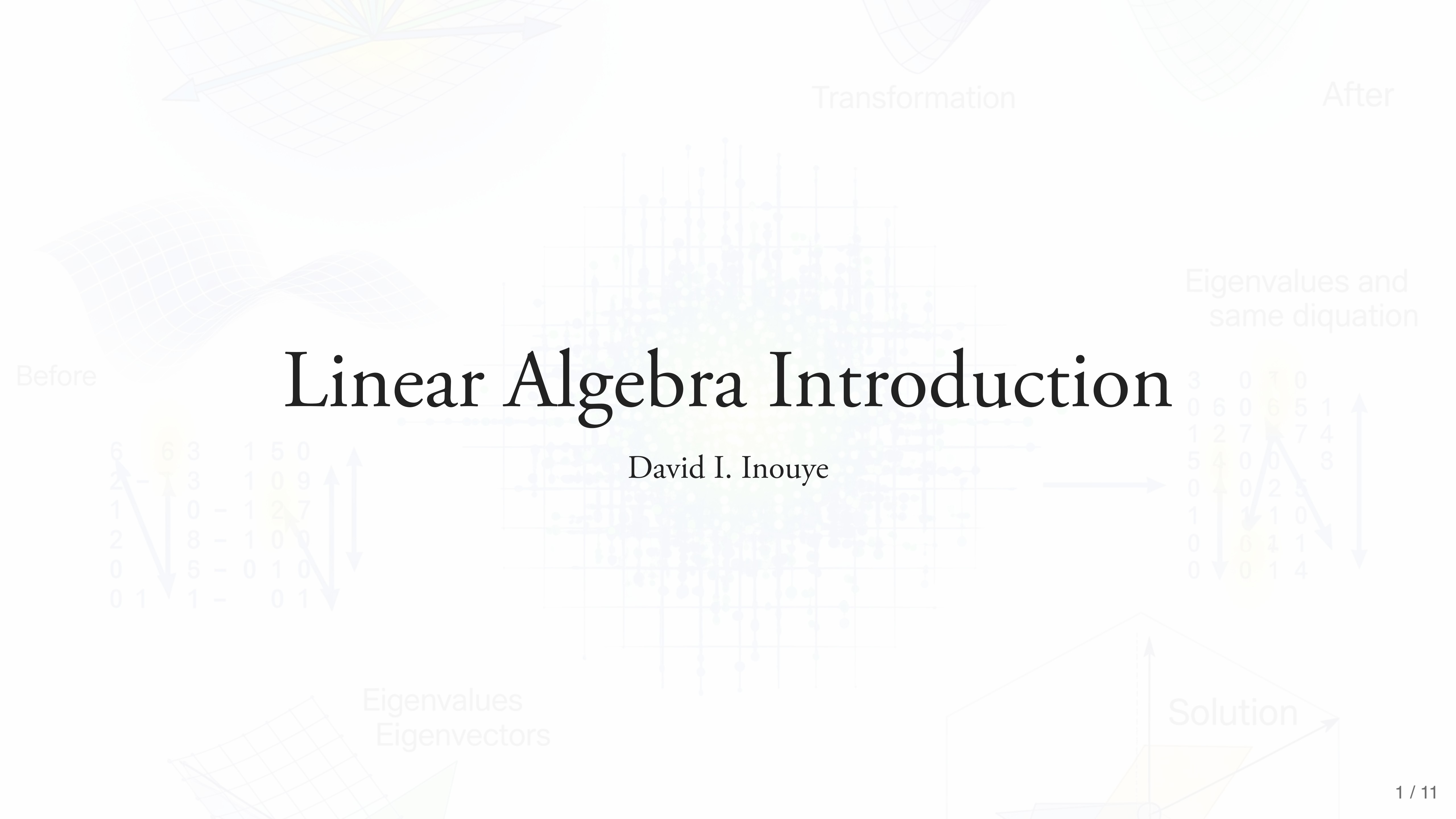
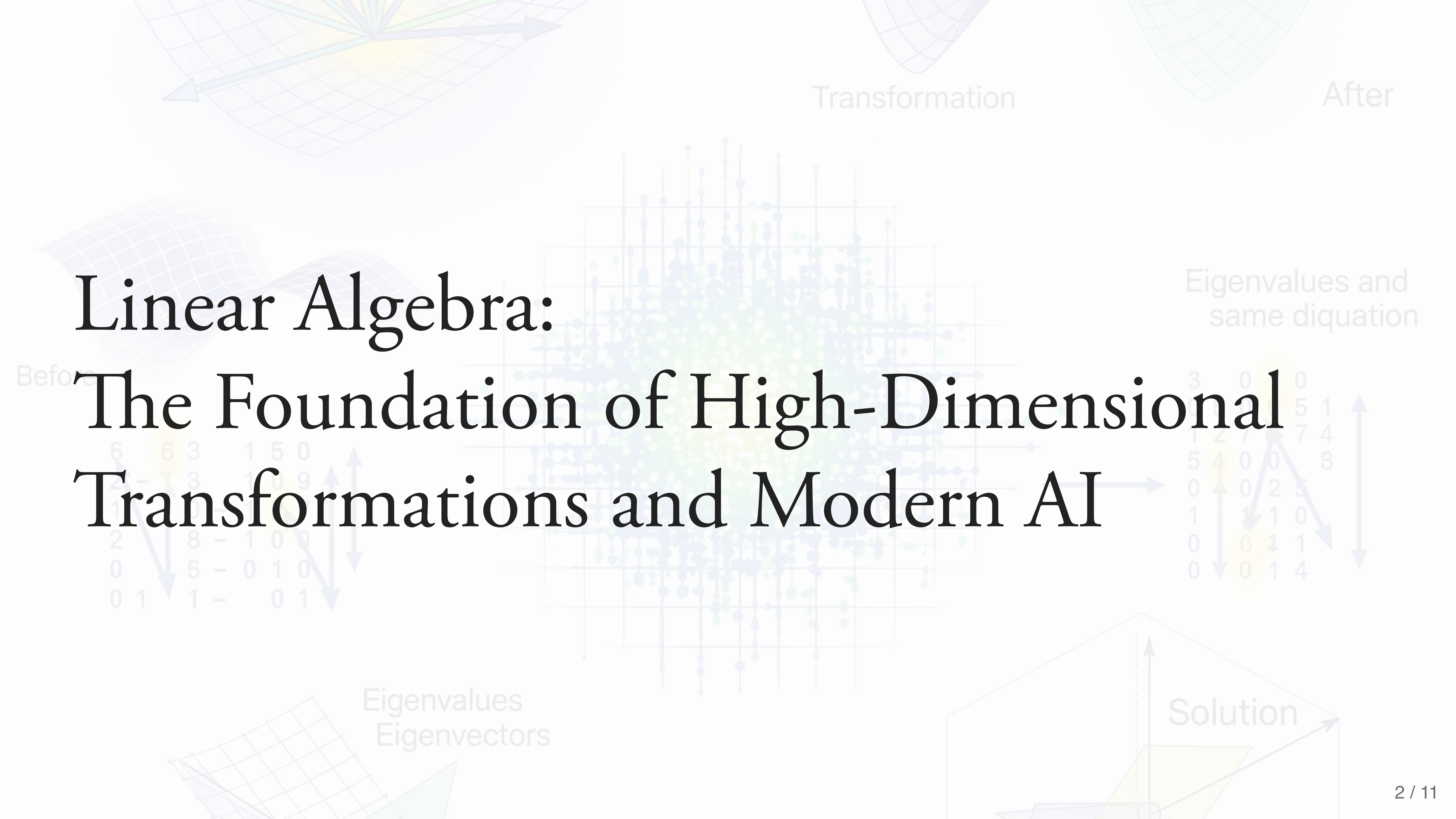


Linear Algebra Introduction

David I. Inouye



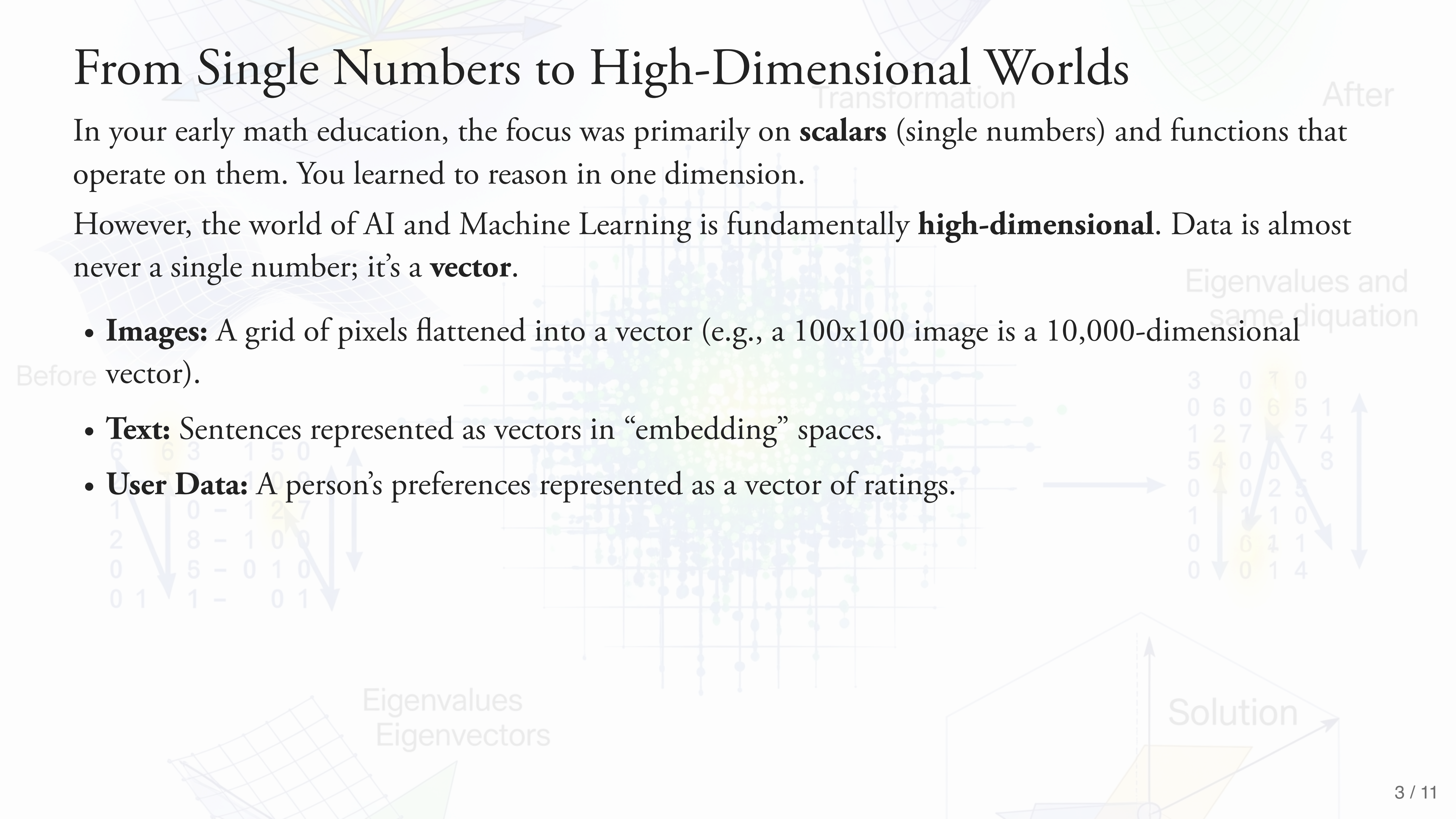
Linear Algebra: The Foundation of High-Dimensional Transformations and Modern AI

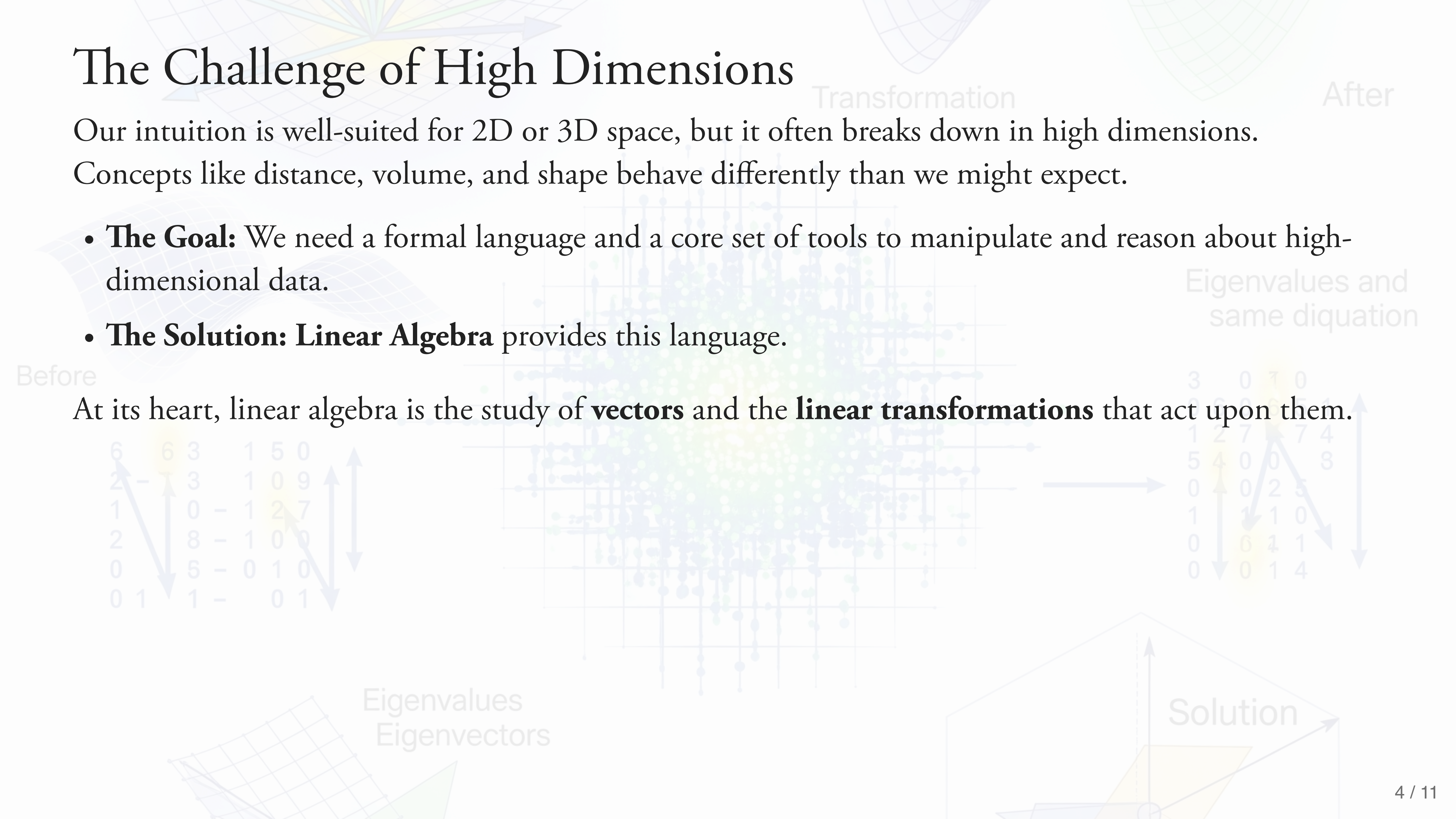
From Single Numbers to High-Dimensional Worlds

In your early math education, the focus was primarily on **scalars** (single numbers) and functions that operate on them. You learned to reason in one dimension.

However, the world of AI and Machine Learning is fundamentally **high-dimensional**. Data is almost never a single number; it's a **vector**.

- **Images:** A grid of pixels flattened into a vector (e.g., a 100x100 image is a 10,000-dimensional vector).
- **Text:** Sentences represented as vectors in "embedding" spaces.
- **User Data:** A person's preferences represented as a vector of ratings.





The Challenge of High Dimensions

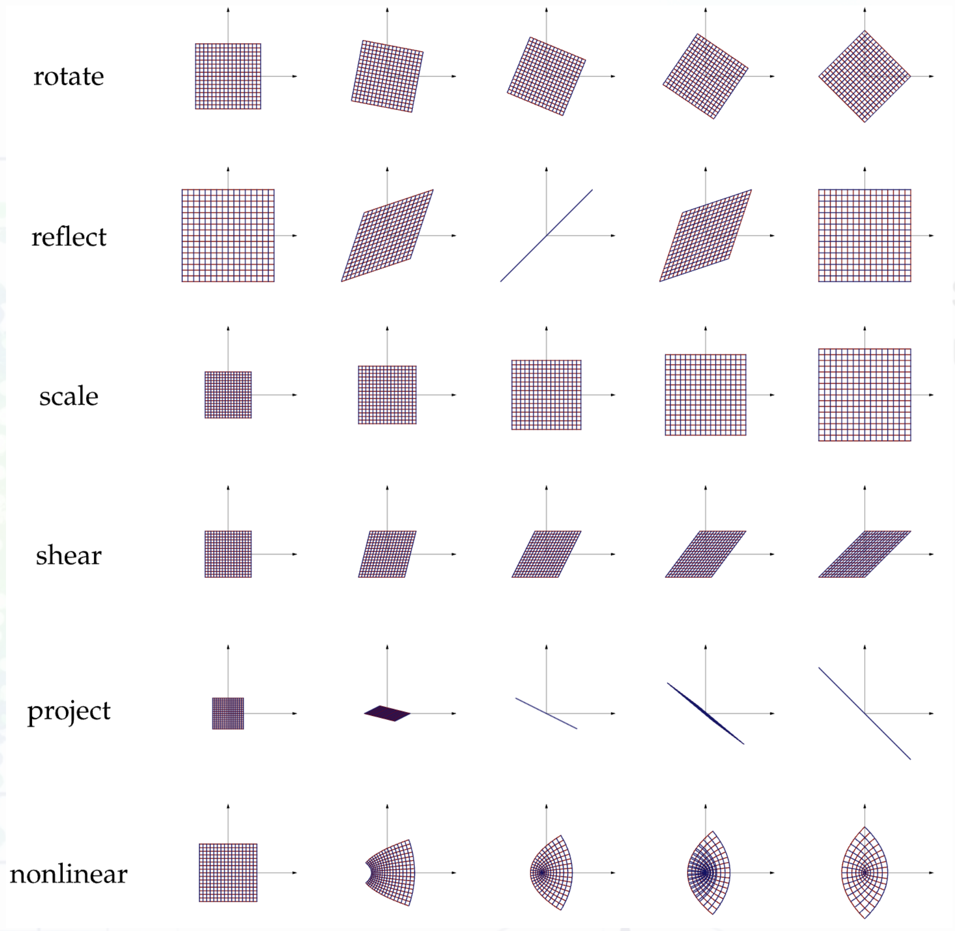
Our intuition is well-suited for 2D or 3D space, but it often breaks down in high dimensions. Concepts like distance, volume, and shape behave differently than we might expect.

- **The Goal:** We need a formal language and a core set of tools to manipulate and reason about high-dimensional data.
- **The Solution: Linear Algebra** provides this language.

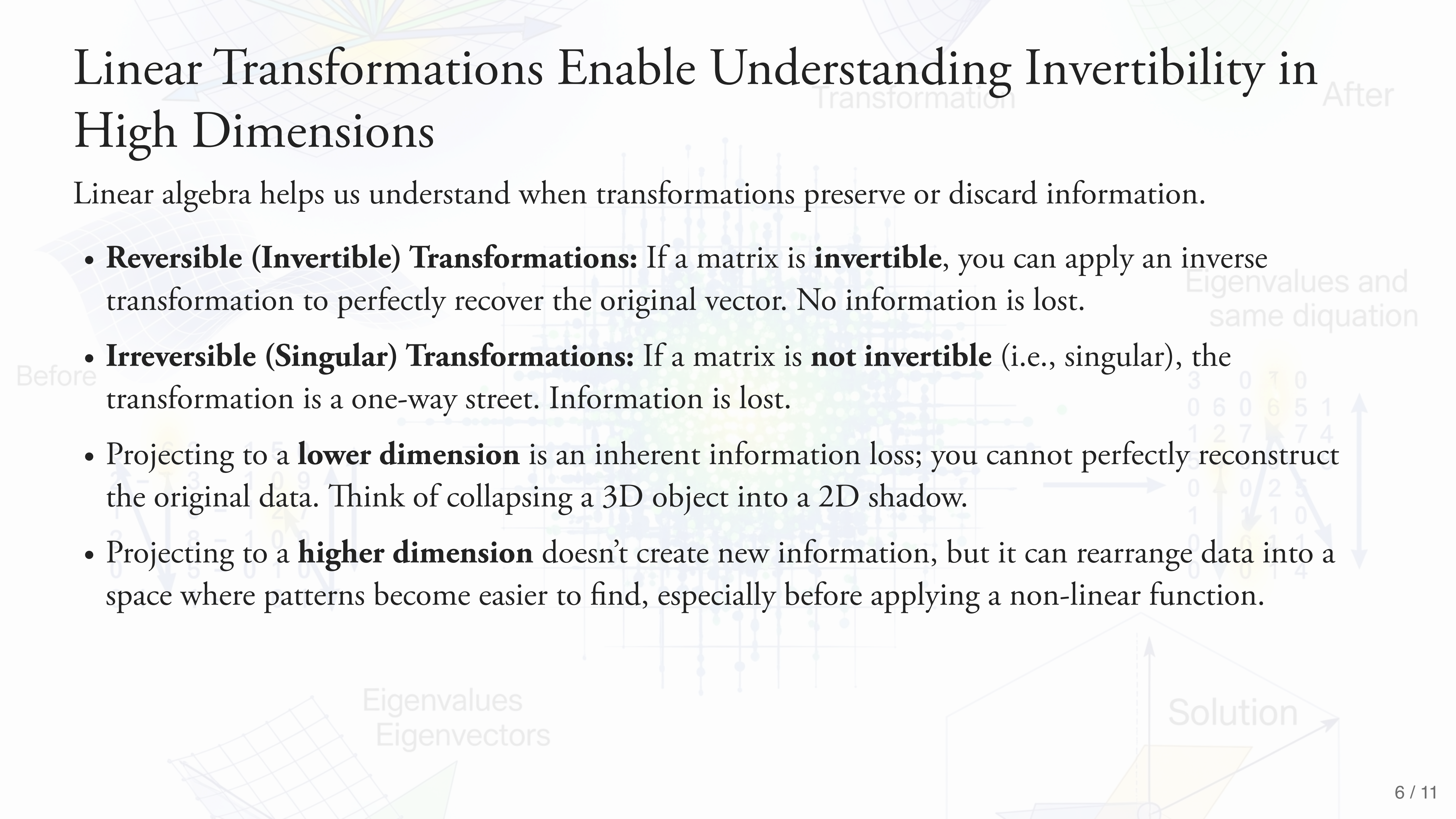
At its heart, linear algebra is the study of **vectors** and the **linear transformations** that act upon them.

What is a Matrix Multiplication, Really?

- The fundamental operation in linear algebra is **matrix multiplication**.
- It's not just a mechanical process; it's a way to apply a **linear transformation** to data. 💪
- A matrix A is an operator that transforms an input vector x into an output vector y via the equation $y = Ax$.
- These transformations include operations like **rotating, scaling, and shearing** the data space itself.
- While seemingly simple, these linear operators are the fundamental building blocks for creating far more complex transformations in high dimensions.



Linear operators (e.g., matrix multiplication) can scale, rotate, and shear the original space and are the simple building block for complex non-linear transformations (bottom).



Linear Transformations Enable Understanding Invertibility in High Dimensions

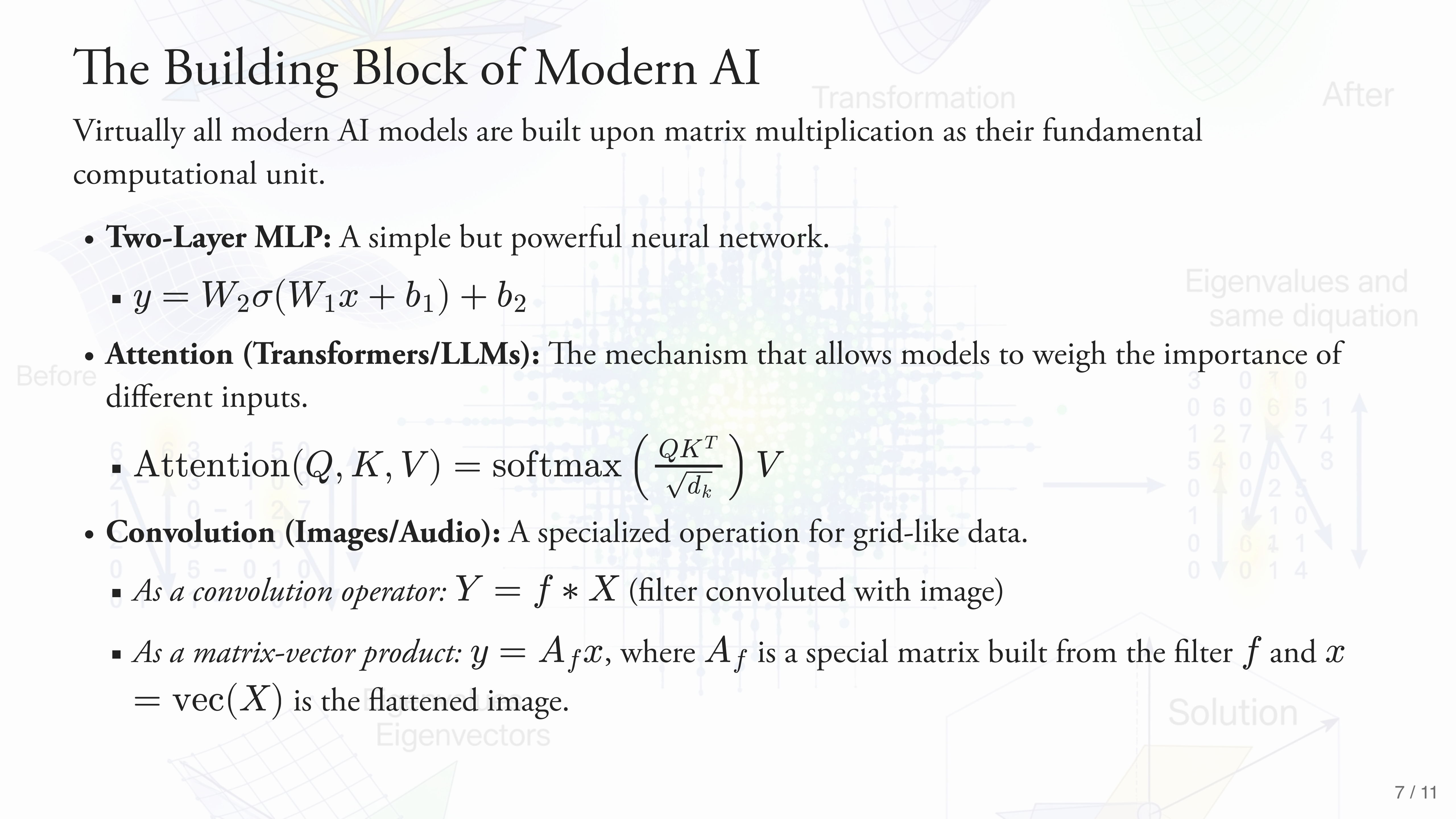
Linear algebra helps us understand when transformations preserve or discard information.

- **Reversible (Invertible) Transformations:** If a matrix is **invertible**, you can apply an inverse transformation to perfectly recover the original vector. No information is lost.
- **Irreversible (Singular) Transformations:** If a matrix is **not invertible** (i.e., singular), the transformation is a one-way street. Information is lost.
- Projecting to a **lower dimension** is an inherent information loss; you cannot perfectly reconstruct the original data. Think of collapsing a 3D object into a 2D shadow.
- Projecting to a **higher dimension** doesn't create new information, but it can rearrange data into a space where patterns become easier to find, especially before applying a non-linear function.

The Building Block of Modern AI

Virtually all modern AI models are built upon matrix multiplication as their fundamental computational unit.

- **Two-Layer MLP:** A simple but powerful neural network.
  - $y = W_2\sigma(W_1x + b_1) + b_2$
- **Attention (Transformers/LLMs):** The mechanism that allows models to weigh the importance of different inputs.
  - $\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$
- **Convolution (Images/Audio):** A specialized operation for grid-like data.
  - *As a convolution operator:* $Y = f * X$ (filter convoluted with image)
  - *As a matrix-vector product:* $y = A_f x$, where A_f is a special matrix built from the filter f and $x = \text{vec}(X)$ is the flattened image.



From Theory to Practice: Visualizing the "Code of Life"

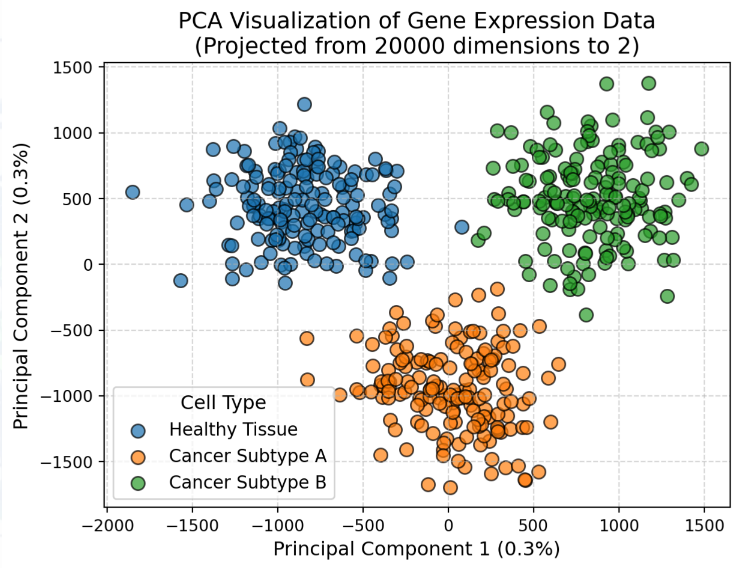
Imagine you are analyzing genetic data to distinguish between healthy cells and cancer cells.

The High-Dimensional Reality: You sequence 500 cells. For *each* cell, you measure the activity of **20,000 genes**.

The Critical Hurdle: You suspect there are distinct clusters (cell types), but **you cannot visualize 20,000 dimensions**. You cannot plot this on a screen to show a doctor. The data is invisible to human intuition.

The Solution: We need to project these 20,000 dimensions down to just **2** so we can create a scatter plot.

▶ Code



The Task: Dimensionality Reduction

This is the standard **Dimensionality Reduction** task: compressing information to gain insight.

- **The Input:** A vector of size d (20,000 genes in our example).
- **The Output:** A vector of size k (2 for visualization in our example).
- **The Goal:** Find a set of $k \ll d$ dimensions that compress the data in a way that loses the the least amount of information (the 2D plot that is as faithful to the 20,000 dimensional reality as possible).

Principal Component Analysis (PCA) is the linear method to solve dimensionality reduction. It finds the best linear projection of the high-dimensional points.

The Math: PCA Minimizes Reconstruction Error

To solve this mathematically, we don't just "squash" the data. We ask for a linear projection that minimizes the difference between the original data and the compressed version:

$$\min_{Z\in\mathbb{R}^{n\times k},W\in\mathbb{R}^{d\times k}:W^TW=I} \|X_c - ZW^T\|_F^2$$

Why we need a Linear Algebra Review: To actually understand and solve this equation, you need more than high-school math. You need to understand:

1. **Matrix Norms:** How do we measure "error" when the variable is a whole matrix?
2. **Orthogonality Constraints:** How do we force the new axes to be independent and perpendicular?
3. **Eigen-decomposition and Singular Value Decomposition (SVD):** The analytic solutions are related to these foundational linear algebra decompositions.

Summary of Linear Alegbra Introduction

We are reviewing Linear Algebra because it provides the mathematical thinking for higher dimensional spaces and can be directly used to solve certain problems in AI

1. **Data is Vectorized:** Whether it's genes, pixels, or words, real-world data lives in high-dimensional vector spaces.
2. **Linear Transformations are the Building Blocks of All High-Dimensional Transformations:** We manipulate this data using matrix multiplication to rotate, scale, and compress it.
3. **Example Application: Linear Unsupervised Dimensionality Reduction:** We may need to reduce dimensions ($k \ll d$) to visualize data or make it computationally manageable.