The Singular Value Decomposition (SVD)

The matrix factorization behind PCA

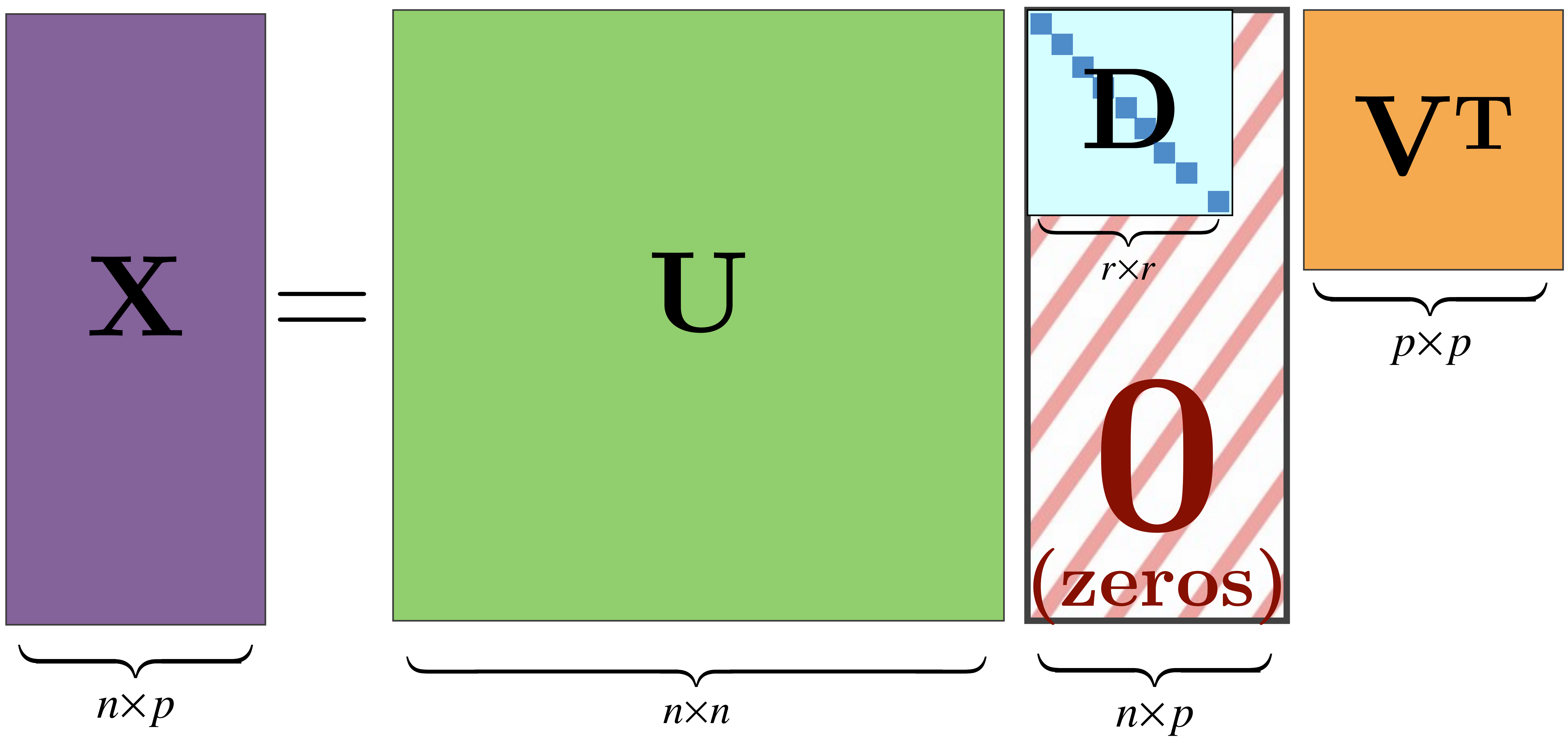
Singular Value Decomposition (SVD)

For any $n \times p$ matrix $\mathbf{X}$ with rank$=r$

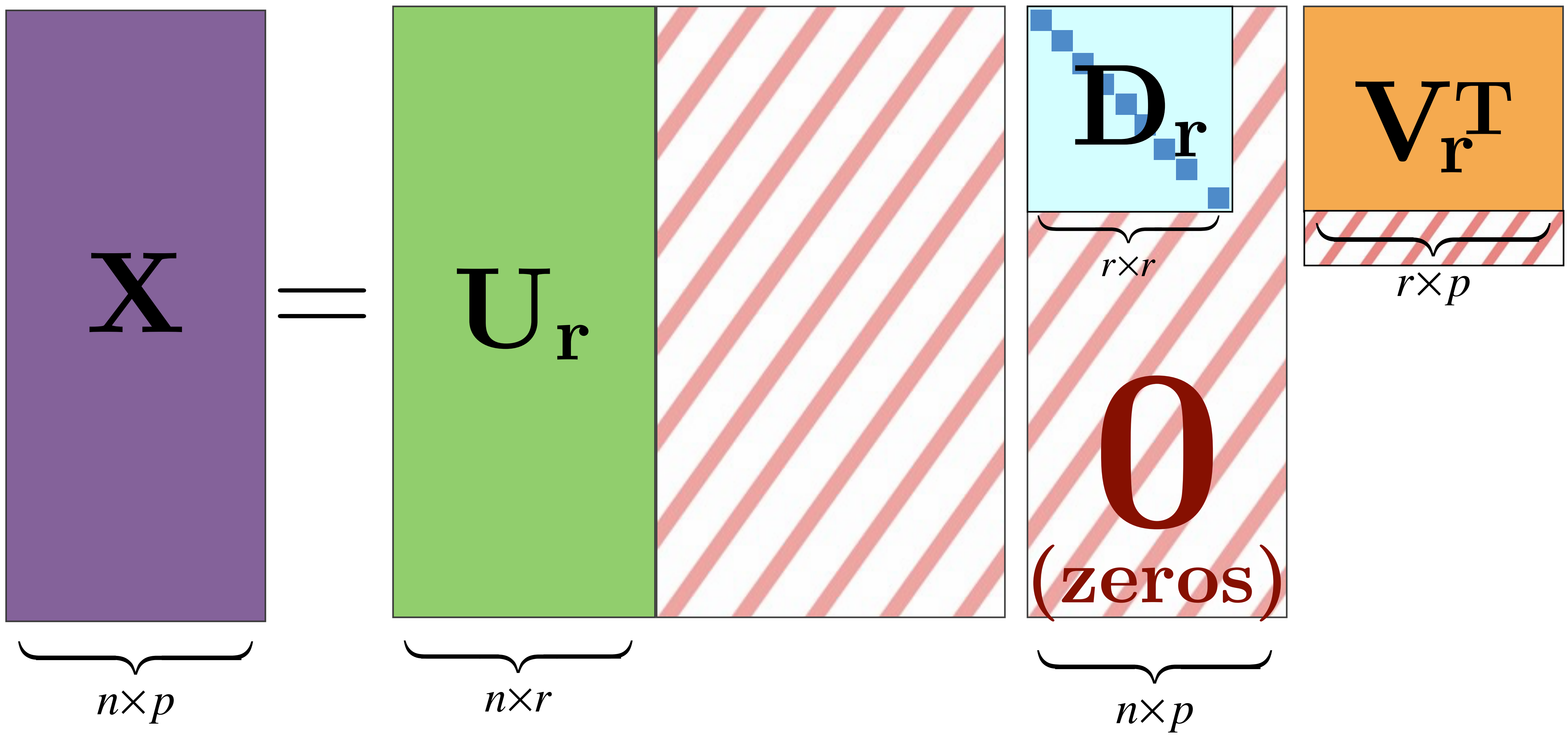
There exists orthogonal matrices $\mathbf{U}_{\mathrm{nxn}}$ and $\mathbf{V}_{\mathrm{pxp}}$
and a diagonal matrix $\mathbf{D}_{\mathrm{rxr}}=\mathrm{diag}(\sigma_1,\sigma_{2,\dots,}\sigma_{\mathrm{r}})$ such that:

$$\mathbf{X} = \mathbf{U}\underbrace{\begin{pmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}}_{n\times p}\mathbf{V}^T \quad \text{with} \quad \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r \geq 0$$

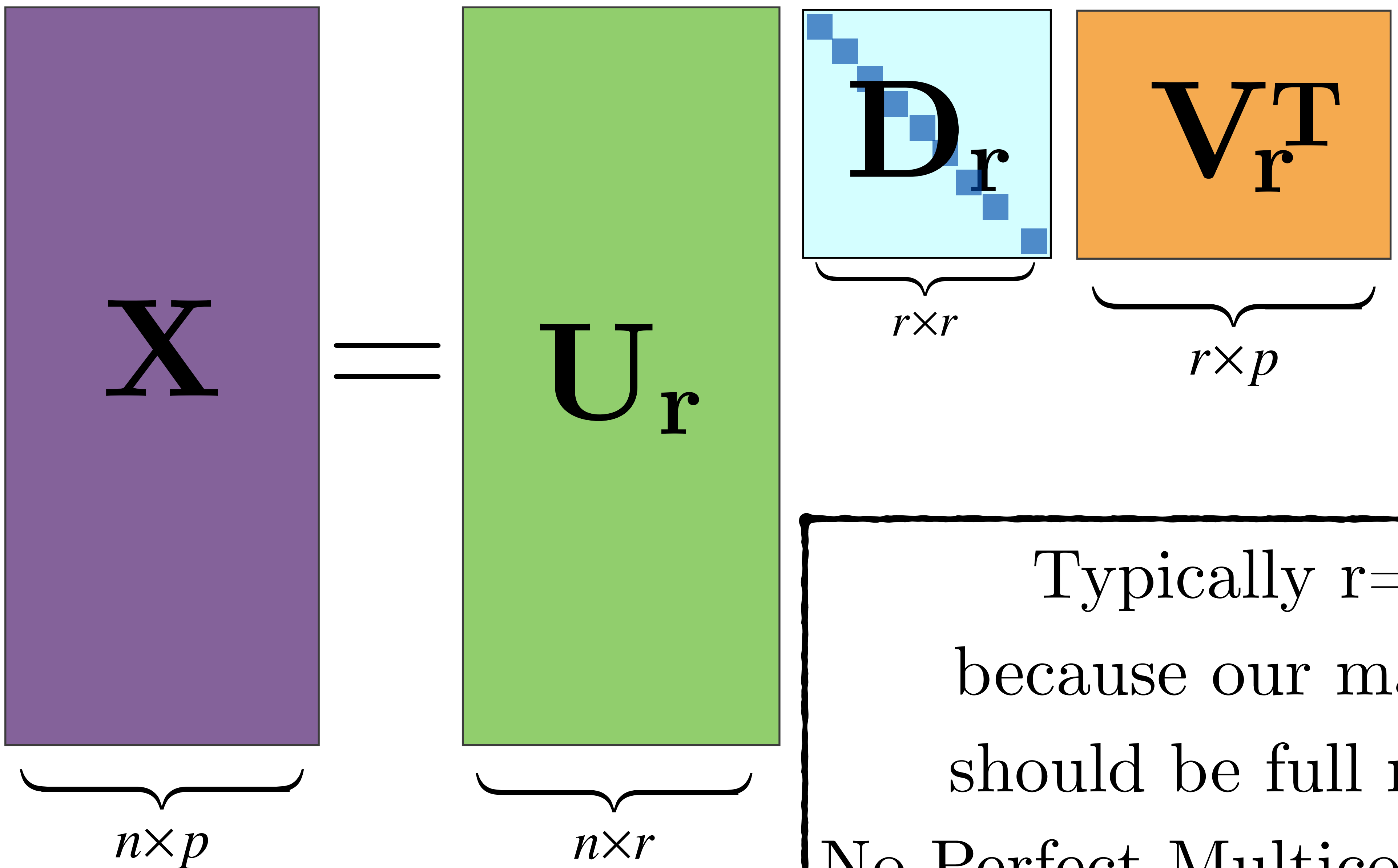
SVD, Illustrated



SVD, Illustrated

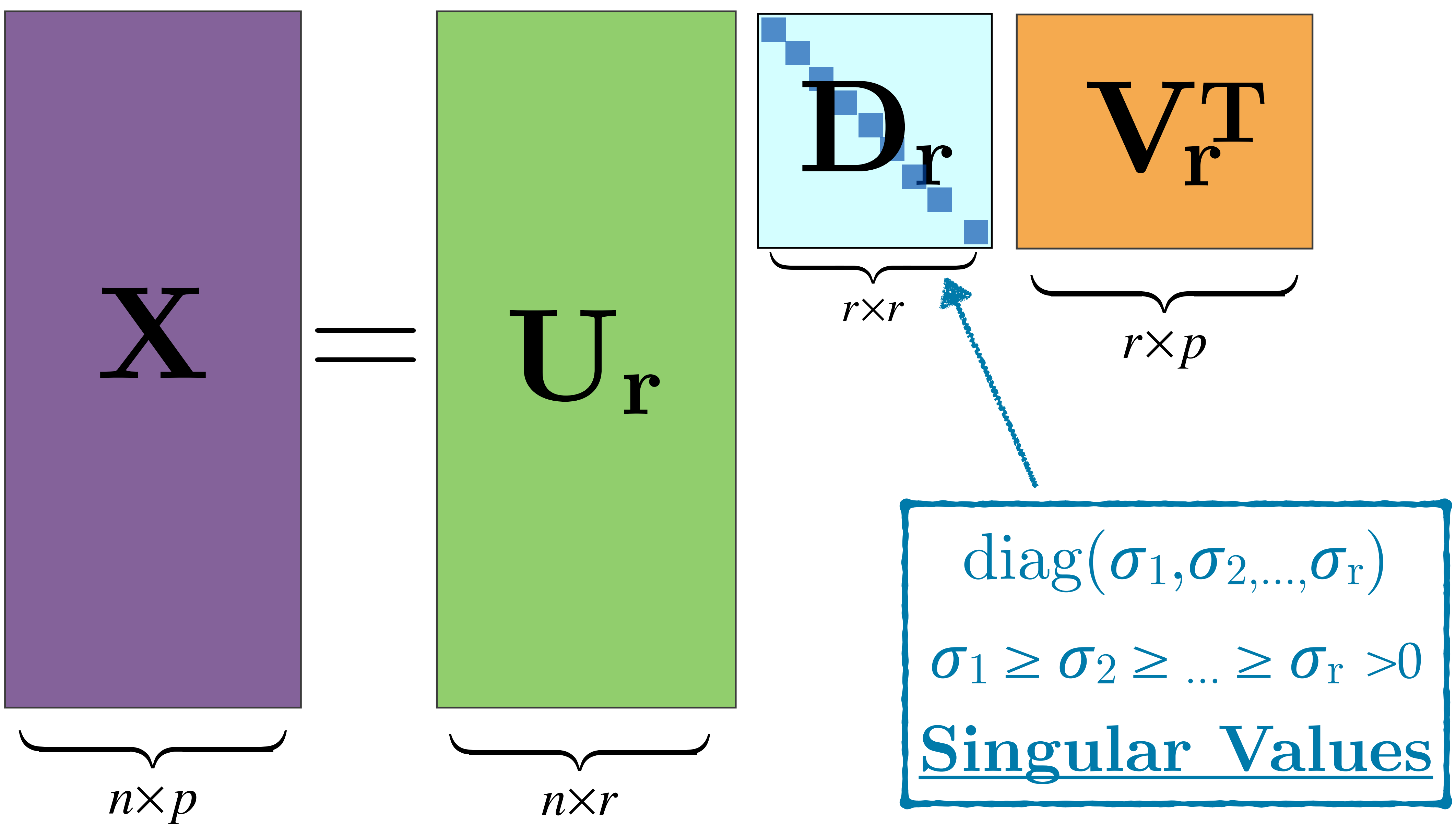


Skinny SVD

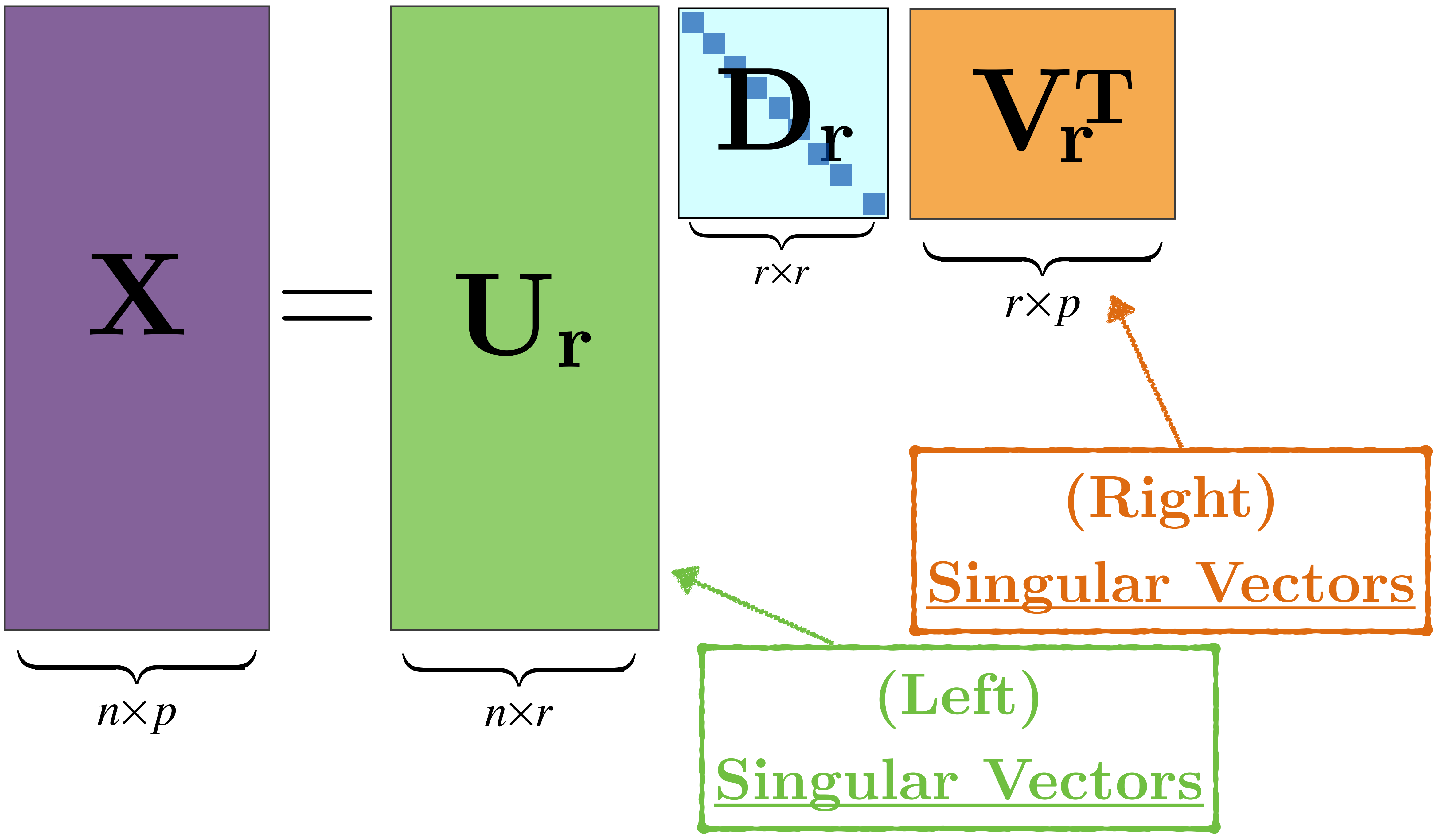


Typically r=p
because our matrix
should be full rank!
No Perfect Multicollinearity

Skinny SVD



Skinny SVD



X
=
U r
D r
V r T
r×r
r×p
n×p
n×r
(Right)
Singular Vectors
(Left)
Singular Vectors

SVD Fun Facts

- Right singular vectors (rows of $\mathbf{V}^{\mathrm{T}}$) are the (orthonormal) eigenvectors of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$
- Left singular vectors (columns of $\mathbf{U}$) are the (orthonormal) eigenvectors of $\mathbf{X}\mathbf{X}^{\mathrm{T}}$
- Singular values are the square roots of the eigenvalues. ($\mathbf{X}\mathbf{X}^{\mathrm{T}}$ and $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ have the same eigenvalues.)

SVD Fun Facts

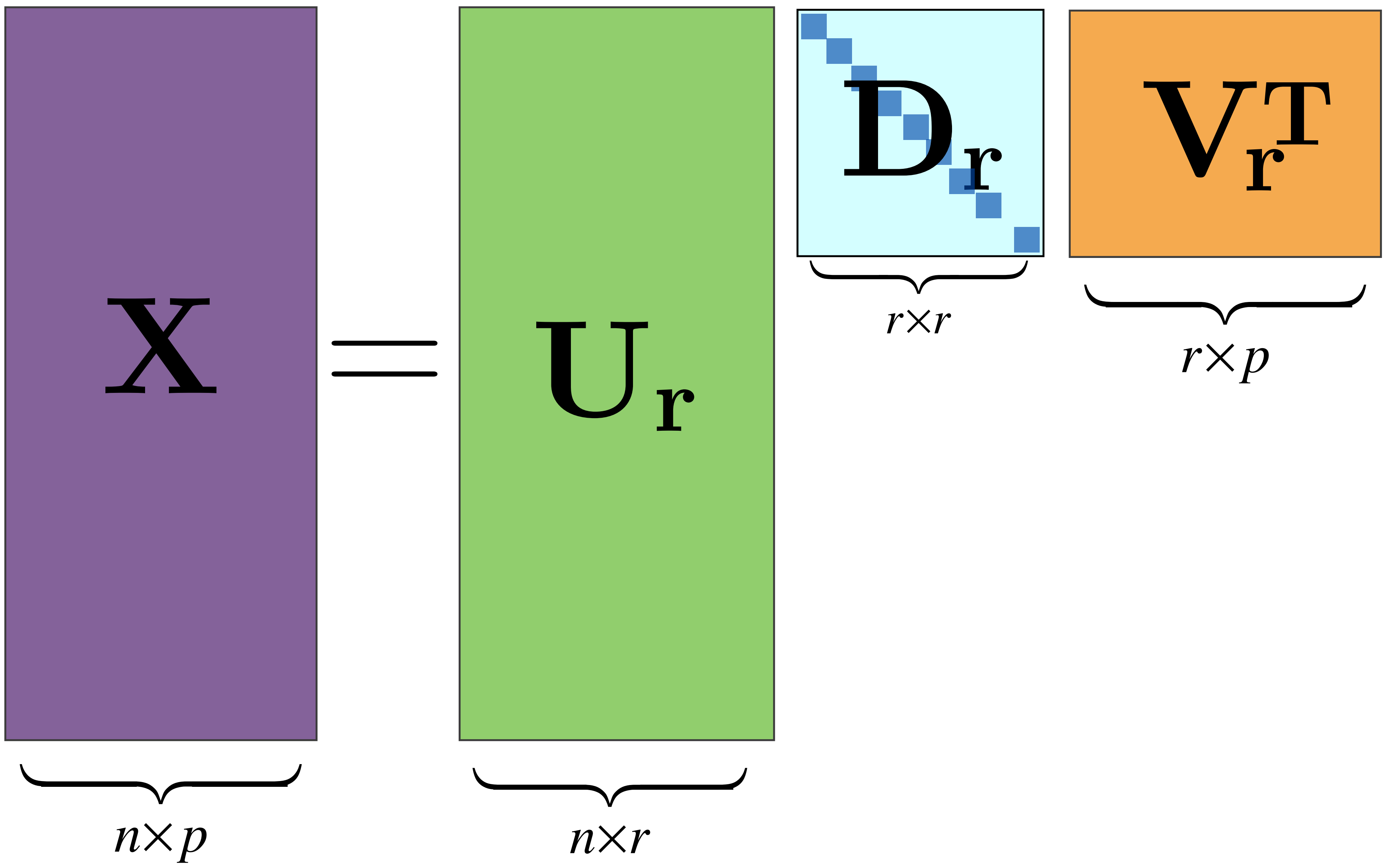
- Right singular vectors (rows of $\mathbf{V}^T$) are the (orthonormal) eigenvectors of $\mathbf{X}^T\mathbf{X}$

If $\mathbf{X}$ contains centered/standardized data then $\mathbf{X}^T\mathbf{X}$ is the covariance/correlation matrix and the singular vectors are principal components! It's PCA!

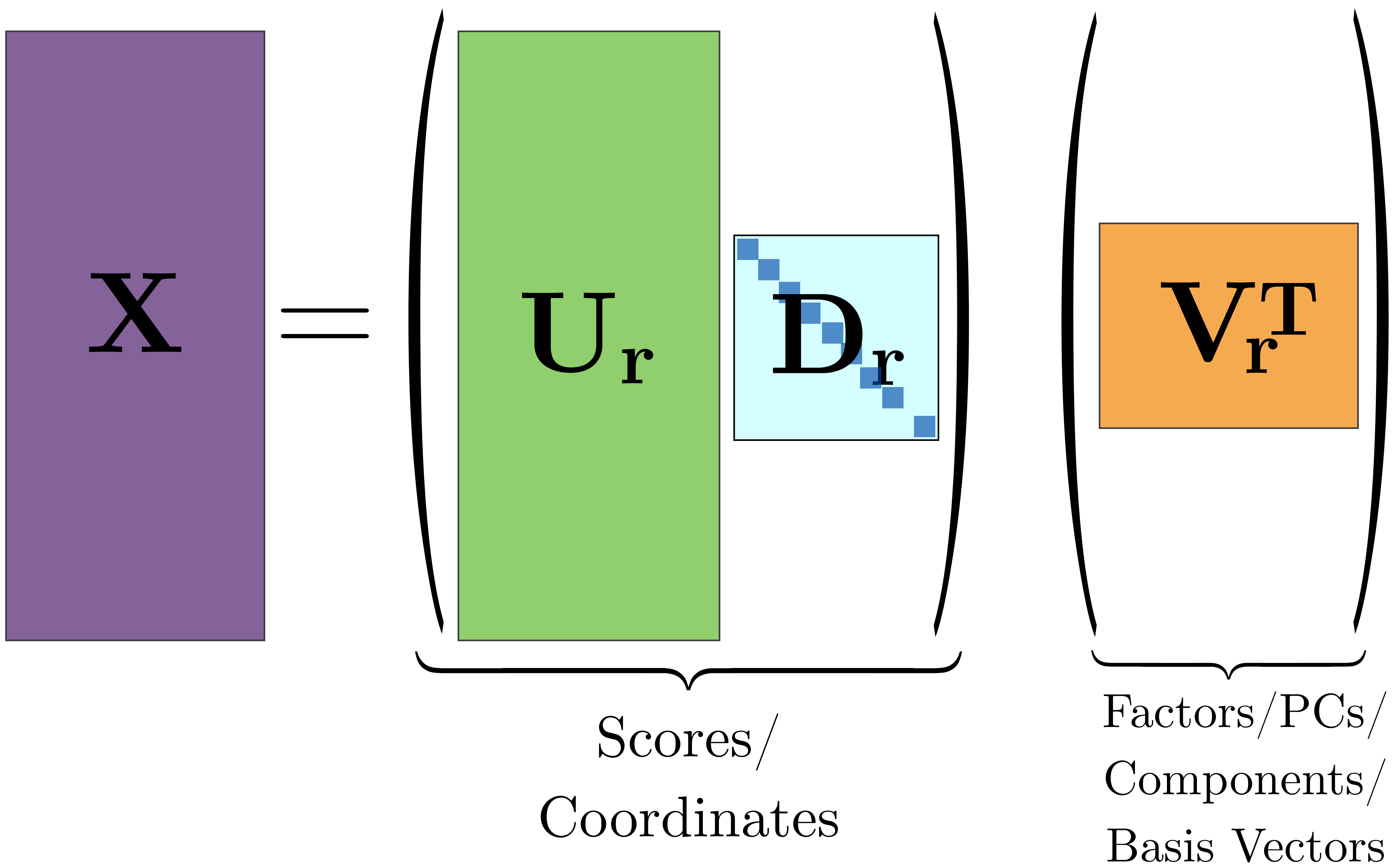
- Left singular vectors (columns of $\mathbf{U}$) are the (orthonormal) eigenvectors of $\mathbf{X}\mathbf{X}^T$

- Singular values are the square roots of the eigenvalues. ($\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T\mathbf{X}$ have the same eigenvalues.)

PCA from SVD



PCA from SVD

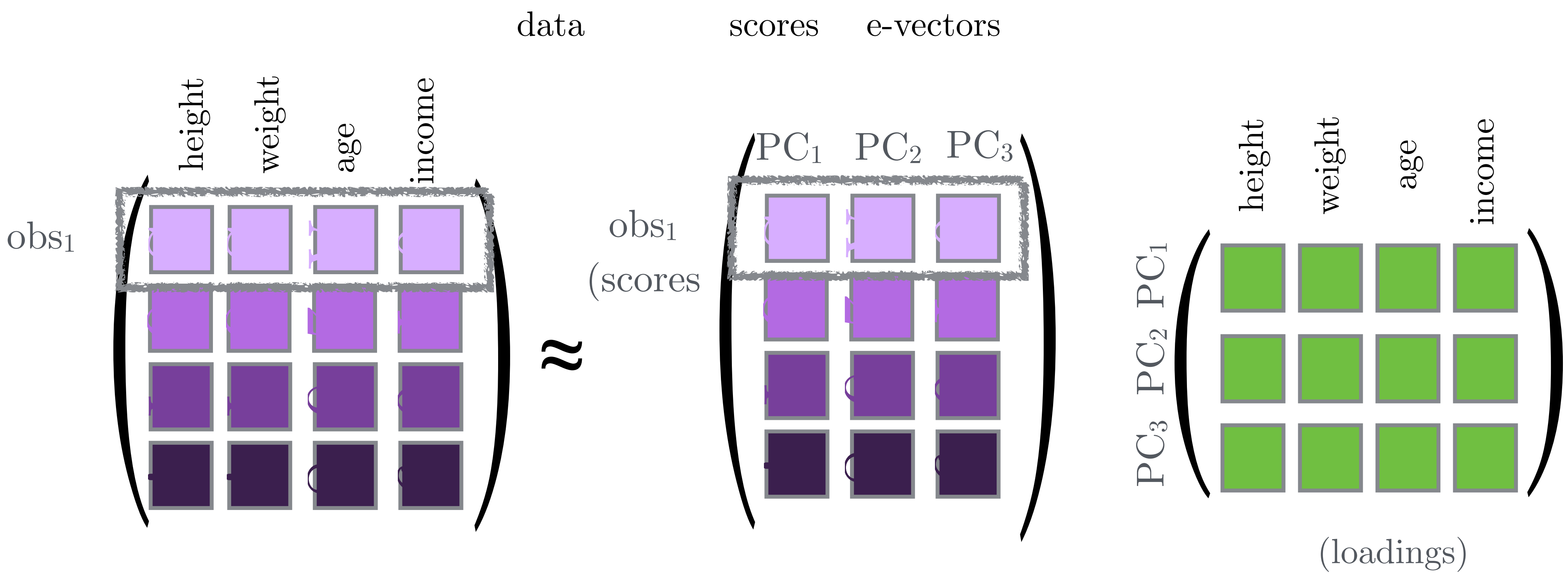


X
=
U_r
D_r
V_r^T
Scores/
Coordinates
Factors/PCs/
Components/
Basis Vectors

Slide Flashback

(Factor Analysis Lecture)

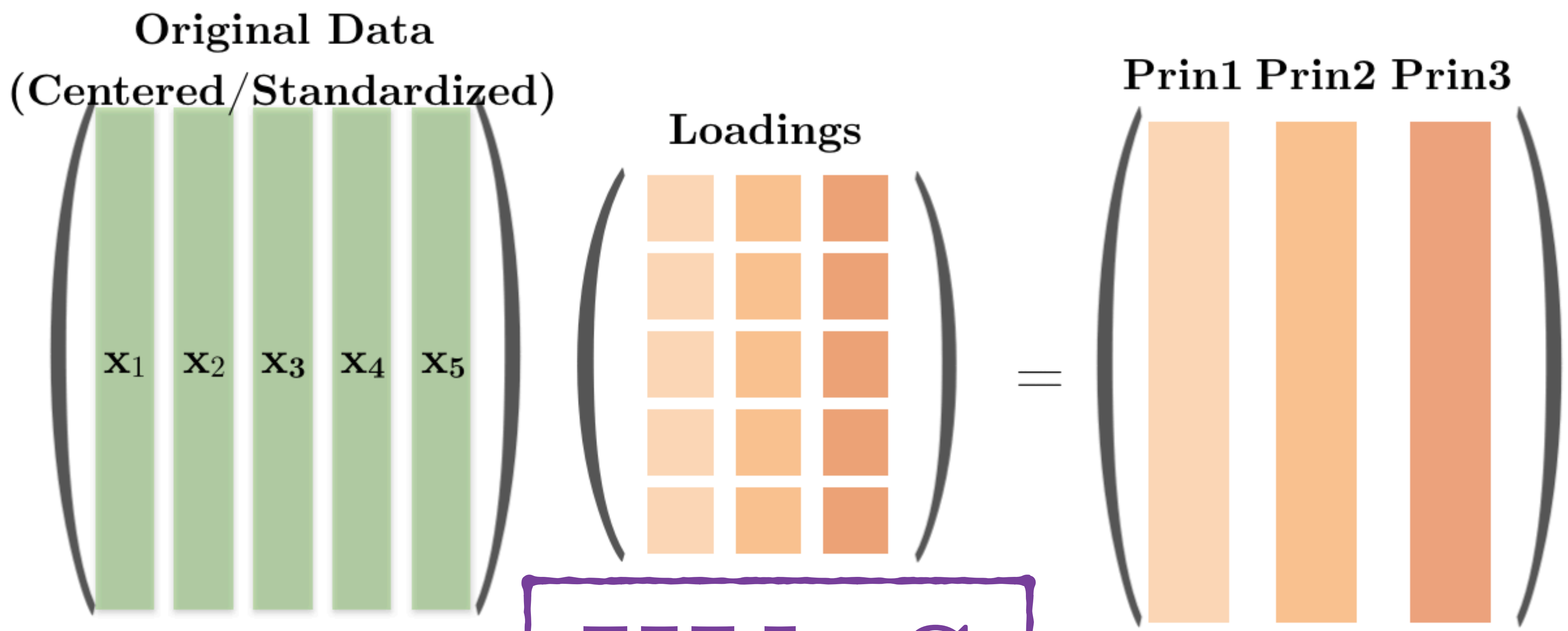
$$\mathbf{X} = \mathbf{S}\mathbf{V}^T$$



Slide Flashback

(PCA Lecture)

Coordinates in the New Basis



$$\mathbf{XV} = \mathbf{S}$$

Slide Flashback

(Orthogonality Lecture)

Orthogonal Matrix

- An orthogonal matrix is easy to maneuver inside matrix equations, since $\mathbf{V}^{-1} = \mathbf{V}^{T}$

- For example if U and V are orthogonal, the following equations are equivalent:

$$\mathbf{XV} = \mathbf{UD}$$

$$\mathbf{X} = \mathbf{UDV}^T$$

$$\mathbf{U}^T\mathbf{X} = \mathbf{DV}^T$$

$$\mathbf{U}^T\mathbf{XV} = \mathbf{D}$$

What's the point

- PCA **IS** the SVD on centered or standardized data.
- Sometimes, practitioners opt for the regular uncentered SVD rather than PCA.
  - True especially in genomics/text/image analysis

Dimension Reduction

$$\Longrightarrow$$

Noise Reduction

Resolving a Matrix into Components

Let $\mathbf{U}_r = [\mathbf{U}_1 \mid \mathbf{U}_2 \mid \ldots \mid \mathbf{U}_r]$ and $\mathbf{V}_r^T = \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \\ \vdots \\ \mathbf{V}_r^T \end{bmatrix}$

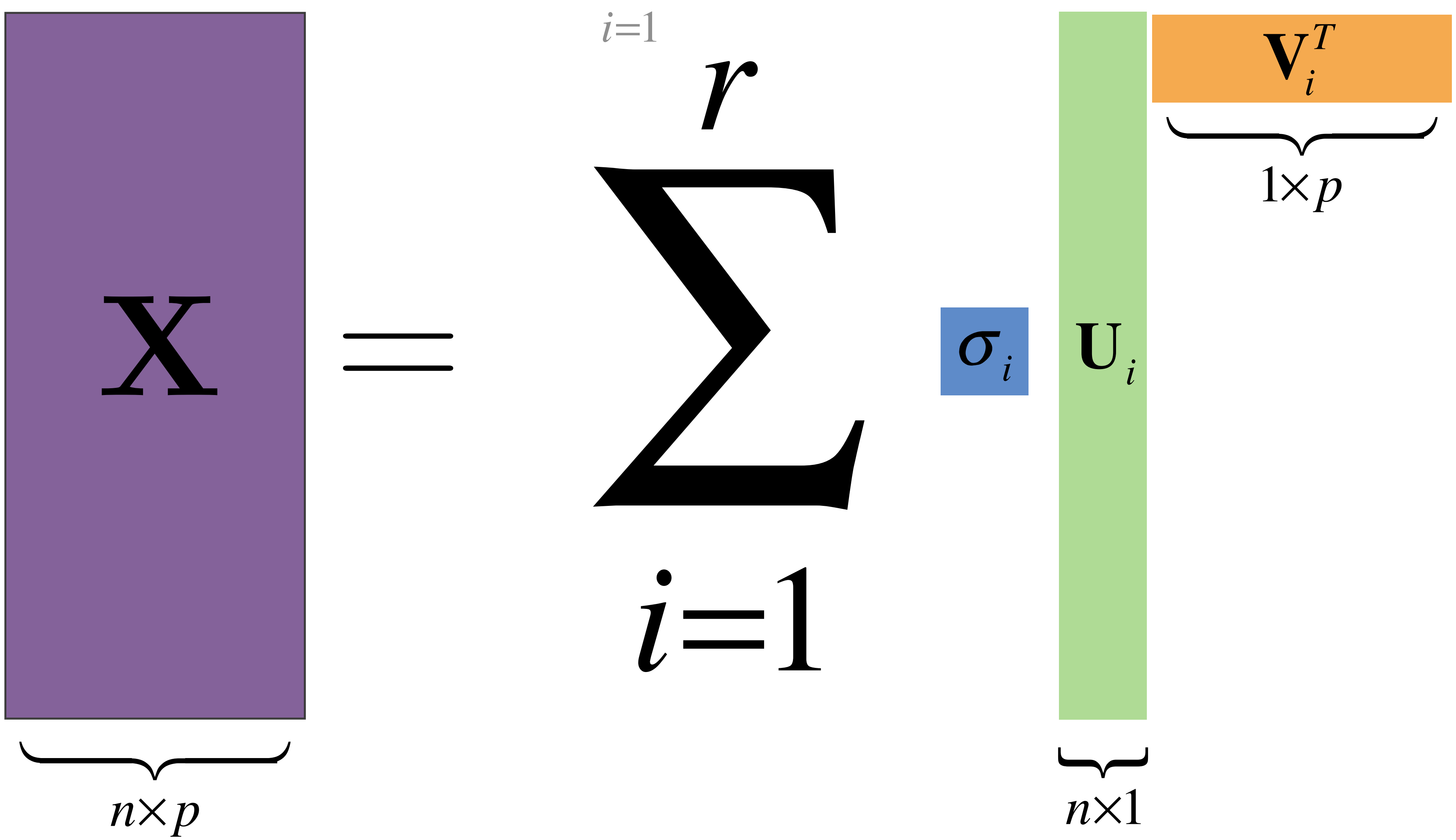
(the left and right singular vectors)

Then,
$$\mathbf{X} = \mathbf{U}_r \mathbf{D}_r \mathbf{V}_r^T = \sum_{i=1}^{r} \sigma_i \mathbf{U}_i \mathbf{V}_i^T$$
$$= \sigma_1 \mathbf{U}_1 \mathbf{V}_1^T + \sigma_2 \mathbf{U}_2 \mathbf{V}_2^T + \sigma_3 \mathbf{U}_3 \mathbf{V}_3^T + \ldots + \sigma_r \mathbf{U}_r \mathbf{V}_r^T$$

It's 'just' matrix multiplication - sum is visualized next slide

Resolving a Matrix into Components

$$\mathbf{X} == \sum_{i=1}^{r} \sigma_i \mathbf{U}_i \mathbf{V}_i^T$$



(sum of rank 1 matrices)

Signal-to-Noise Ratio

and

Noise Reduction

$$\mathbf{X} = \sigma_1 \mathbf{U}_1 \mathbf{V}_1^T + \sigma_2 \mathbf{U}_2 \mathbf{V}_2^T + \sigma_3 \mathbf{U}_3 \mathbf{V}_3^T + \ldots + \sigma_r \mathbf{U}_r \mathbf{V}_r^T$$

Think of these as
"unit basis directions"
for the matrix X.

Signal-to-Noise Ratio

and

Noise Reduction

$$\mathbf{X} = \sigma_1 \mathbf{U}_1 \mathbf{V}_1^T + \sigma_2 \mathbf{U}_2 \mathbf{V}_2^T + \sigma_3 \mathbf{U}_3 \mathbf{V}_3^T + \ldots + \sigma_r \mathbf{U}_r \mathbf{V}_r^T$$

Think of these as coordinates that say how much "signal" or information of the matrix **X** is directed along each basis direction.

The components are ordered by the magnitude of the signal.

$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r$

Signal-to-Noise Ratio

and

Noise Reduction

$$\mathbf{X} = \sigma_1 \mathbf{U}_1 \mathbf{V}_1^T + \sigma_2 \mathbf{U}_2 \mathbf{V}_2^T + \sigma_3 \mathbf{U}_3 \mathbf{V}_3^T + \ldots + \sigma_r \mathbf{U}_r \mathbf{V}_r^T$$

Anytime we have signal, we inevitably have some noise.

Our data is typically an imperfect depiction of reality.

Signal-to-Noise Ratio

and

Noise Reduction

$$\mathbf{X} = \sigma_1 \mathbf{U}_1 \mathbf{V}_1^T + \sigma_2 \mathbf{U}_2 \mathbf{V}_2^T + \sigma_3 \mathbf{U}_3 \mathbf{V}_3^T + \ldots + \sigma_r \mathbf{U}_r \mathbf{V}_r^T$$

If we assume there is no pattern to the noise -
That it is uniformly distributed "in every direction"

Then amount of noise in each of the terms in this sum is the same!

Signal-to-Noise Ratio

and

Noise Reduction

$$\mathbf{X} = \sigma_1 \mathbf{U}_1 \mathbf{V}_1^T + \sigma_2 \mathbf{U}_2 \mathbf{V}_2^T + \sigma_3 \mathbf{U}_3 \mathbf{V}_3^T + \ldots + \sigma_r \mathbf{U}_r \mathbf{V}_r^T$$

The amount of signal in each of the terms in this sum is decreasing →

The amount of noise in each of the terms in this sum is the same.

⇘ ⇙

The signal-to-noise ratio is higher in first terms. Last terms could be mostly noise

Signal-to-Noise Ratio

and

Noise Reduction

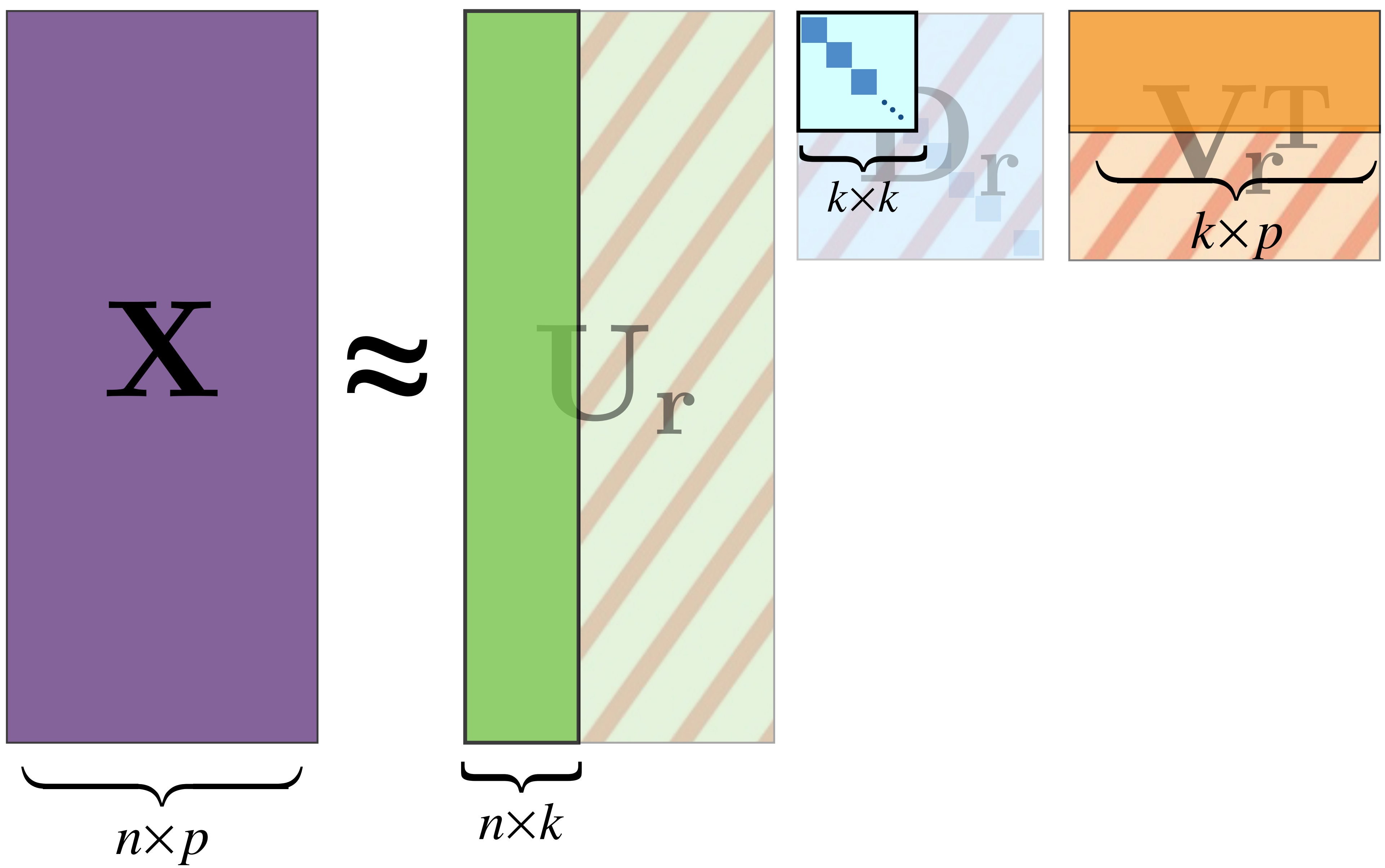
$$\mathbf{X} = \sigma_1 \mathbf{U}_1 \mathbf{V}_1^T + \sigma_2 \mathbf{U}_2 \mathbf{V}_2^T + \sigma_3 \mathbf{U}_3 \mathbf{V}_3^T + \ldots + \sigma_r \mathbf{U}_r \mathbf{V}_r^T$$

If the last terms have more noise, then we won't lose much information by omitting them, AND we may actually lose a good bit of noise. That's a perk of dimension reduction.

Truncated SVD

$$\mathbf{X} = \sigma_1 \mathbf{U}_1 \mathbf{V}_1^T + \sigma_2 \mathbf{U}_2 \mathbf{V}_2^T + \sigma_3 \mathbf{U}_3 \mathbf{V}_3^T + \ldots + \sigma_k \mathbf{U}_k \mathbf{V}_k^T$$

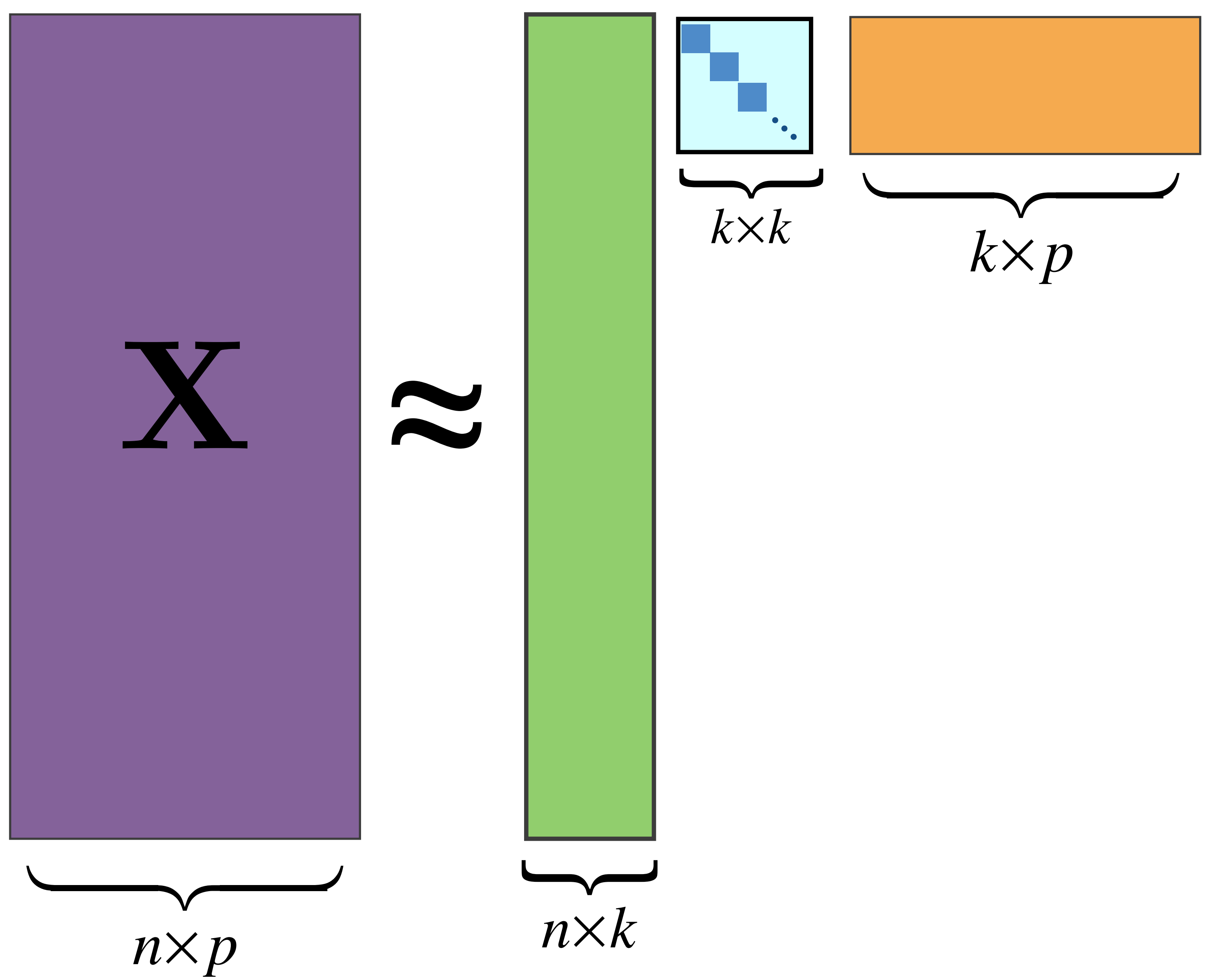
k≤r



Truncated SVD

$$\mathbf{X} = \sigma_1 \mathbf{U}_1 \mathbf{V}_1^T + \sigma_2 \mathbf{U}_2 \mathbf{V}_2^T + \sigma_3 \mathbf{U}_3 \mathbf{V}_3^T + \ldots + \sigma_k \mathbf{U}_k \mathbf{V}_k^T$$

k≤r



The SVD of Dr. Rappa

Our Fearless Leader

Let's start in B&W

Imagine this is your data matrix.

Let's start in B&W

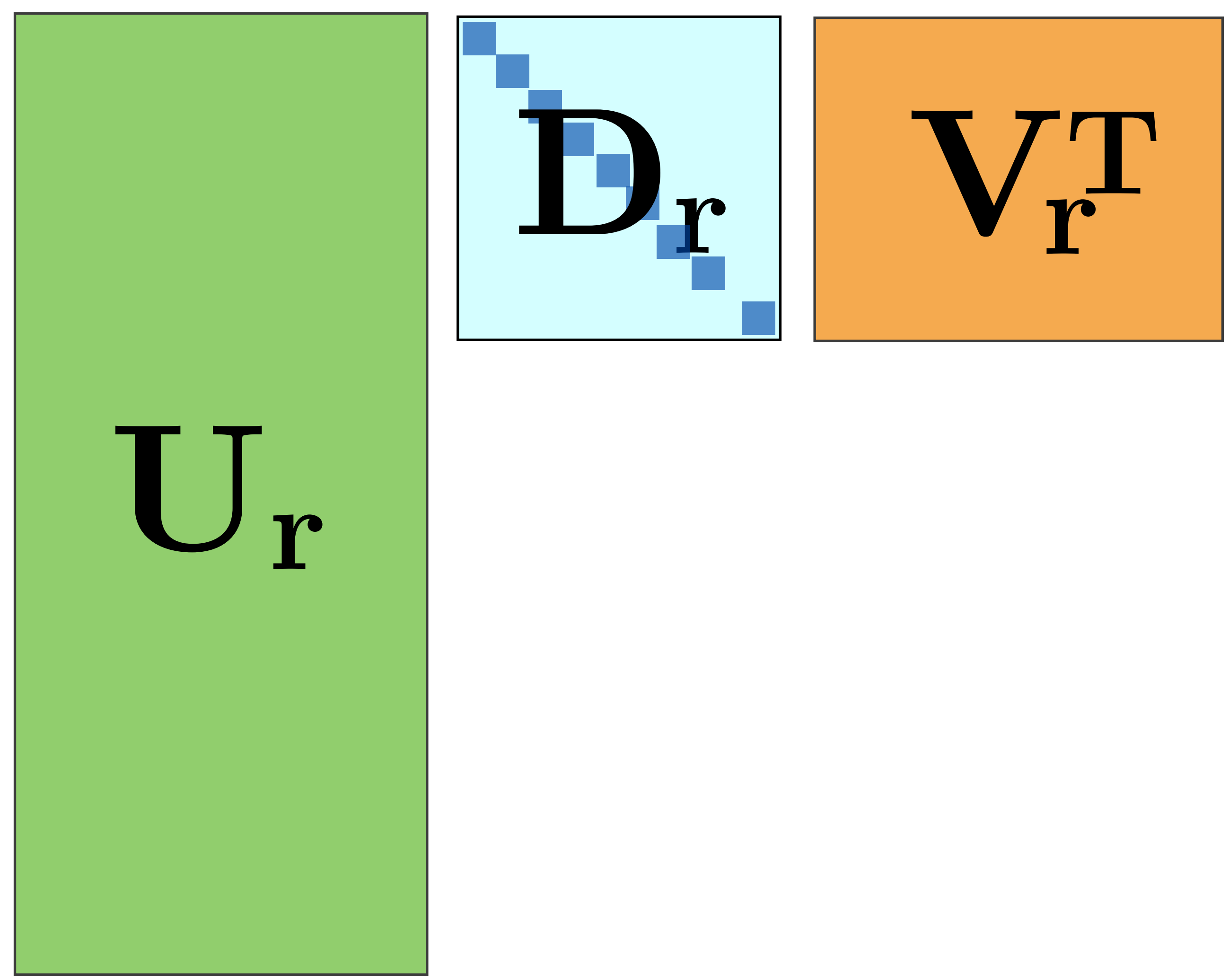
Each pixel represents a number between 0 and 1.
0=black   1=white

The matrix is 160 x 250, and is called **rappa.grey**

Take the SVD of the matrix

```r
rappasvd=svd(rappa.grey)
U=rappasvd$u
d=rappasvd$d
Vt=t(rappasvd$v)
```

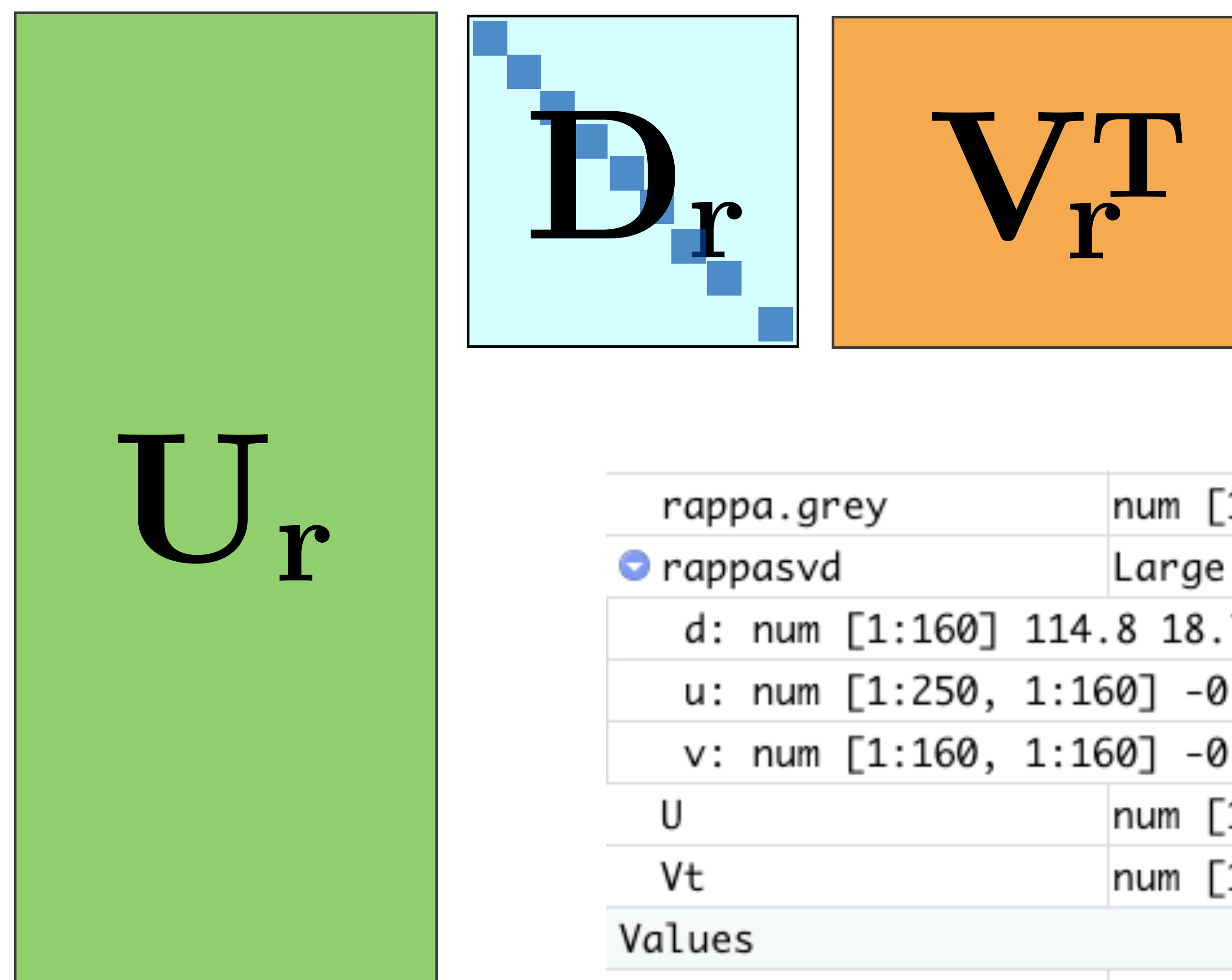
=



Take the SVD of the matrix

```
rappasvd=svd(rappa.grey)
U=rappasvd$u
d=rappasvd$d
Vt=t(rappasvd$v)
```

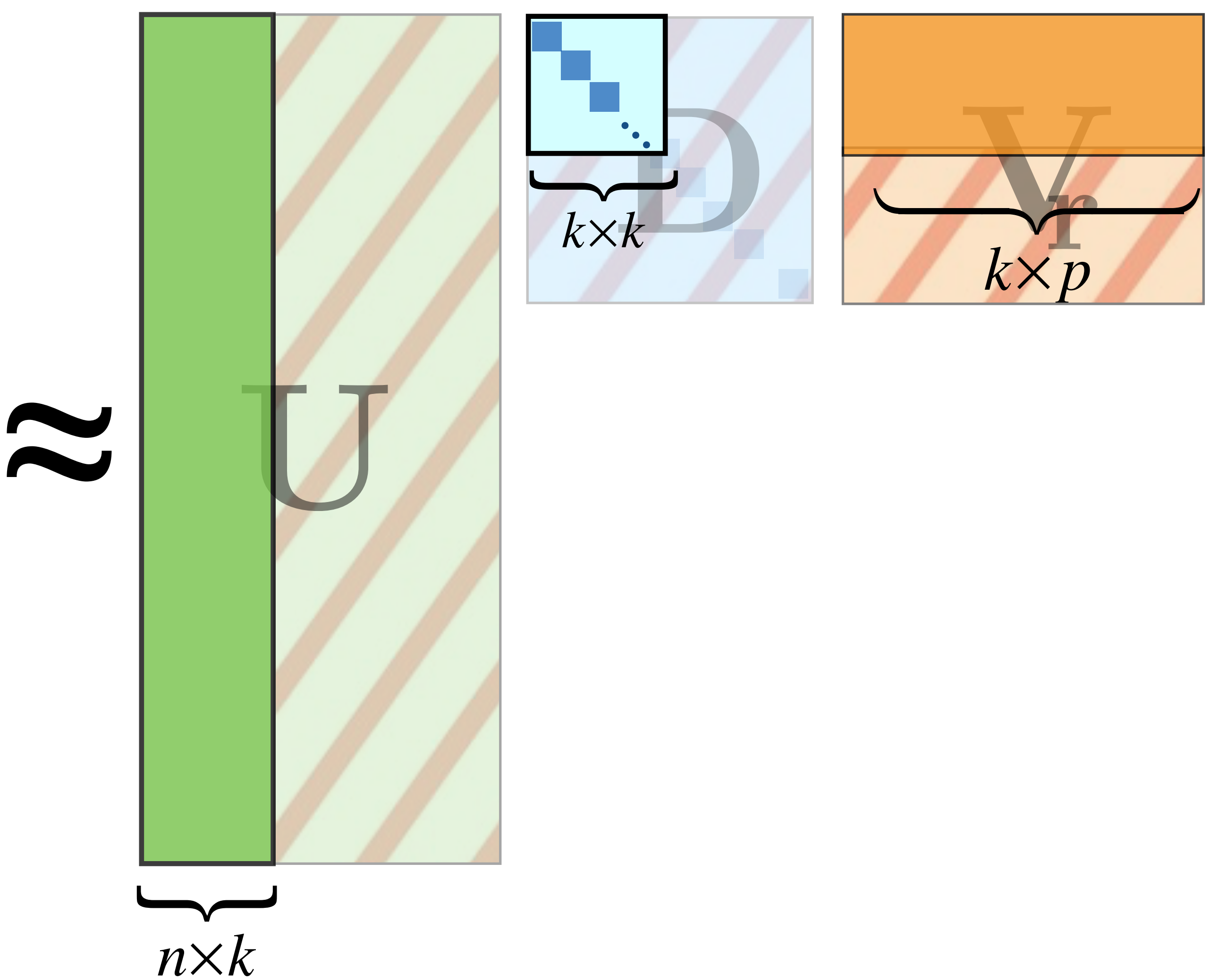
=



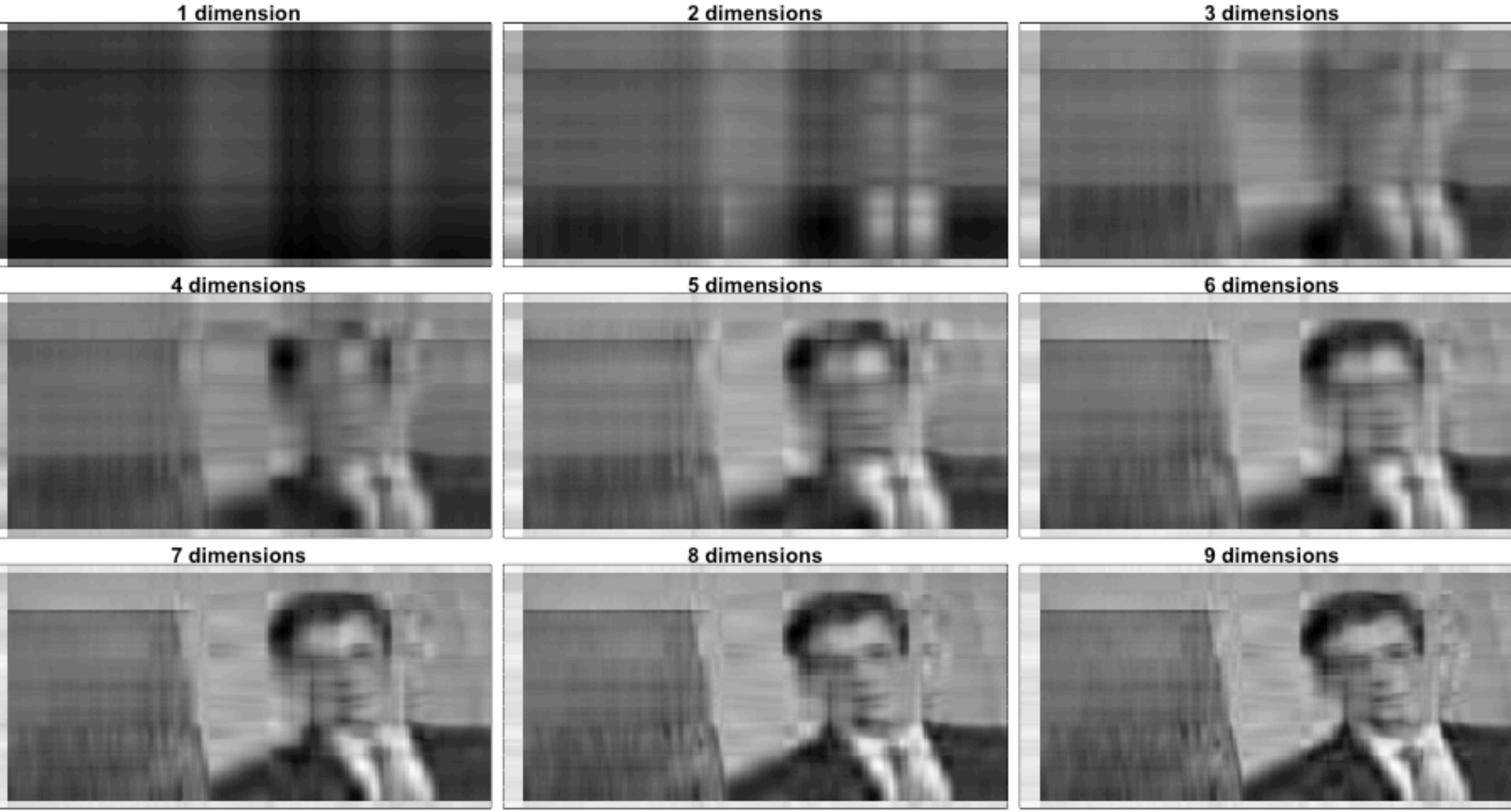
rappa.grey	num [1:250, 1
rappasvd	Large list (3
d: num [1:160] 114.8 18.7 14.1 1	
u: num [1:250, 1:160] -0.107 -0.	
v: num [1:160, 1:160] -0.135 -0.	
U	num [1:250, 1
Vt	num [1:160, 1
Values	
d	num [1:160] 1

rank k approximations

```
RappaRank_k =  U[,1:k] %*% diag(d[1:k]) %*% Vt[1:k,]
image(RappaRank_k,
      col=grey((0:1000)/1000),
      main=paste(k,"dimensions"),
      xaxt = 'n',
      yaxt = 'n')
```



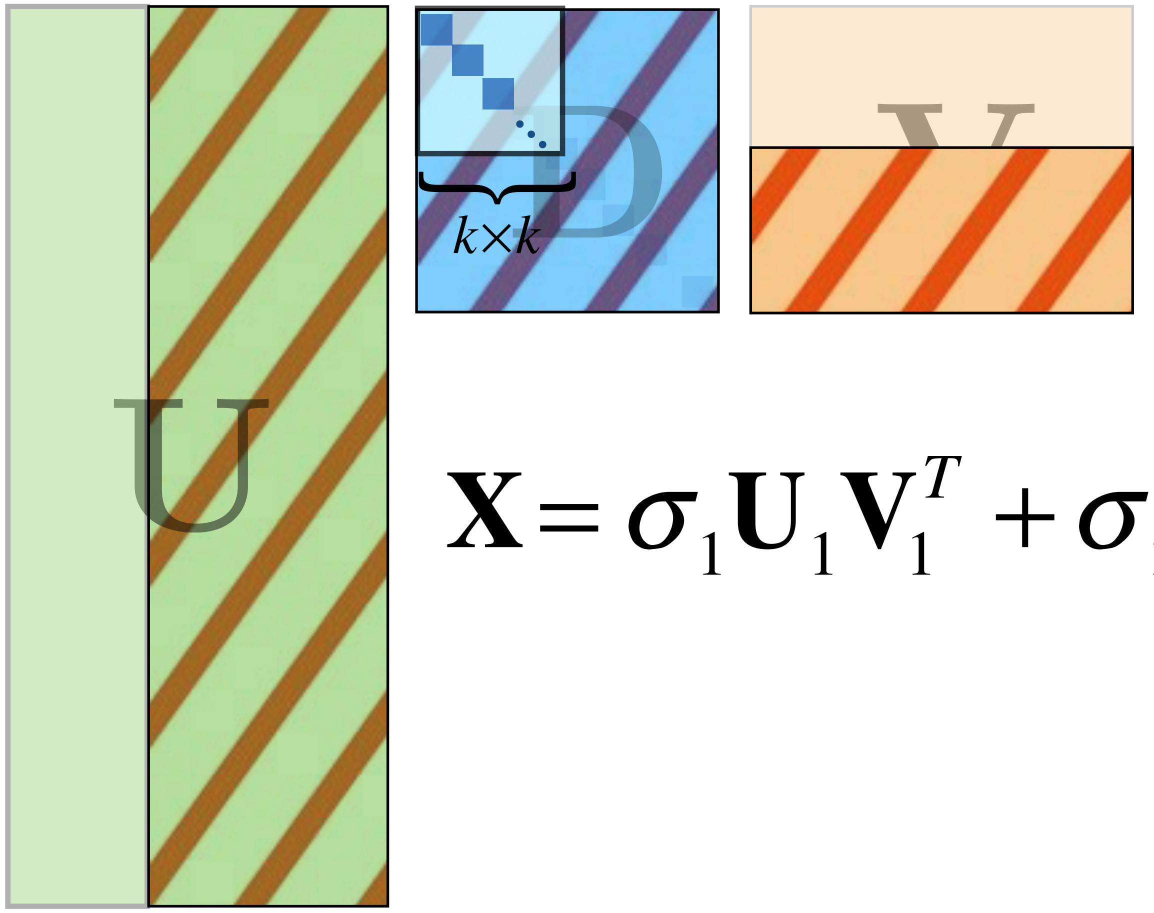
rank k approximations



rank k approximations

What about the dropped components?

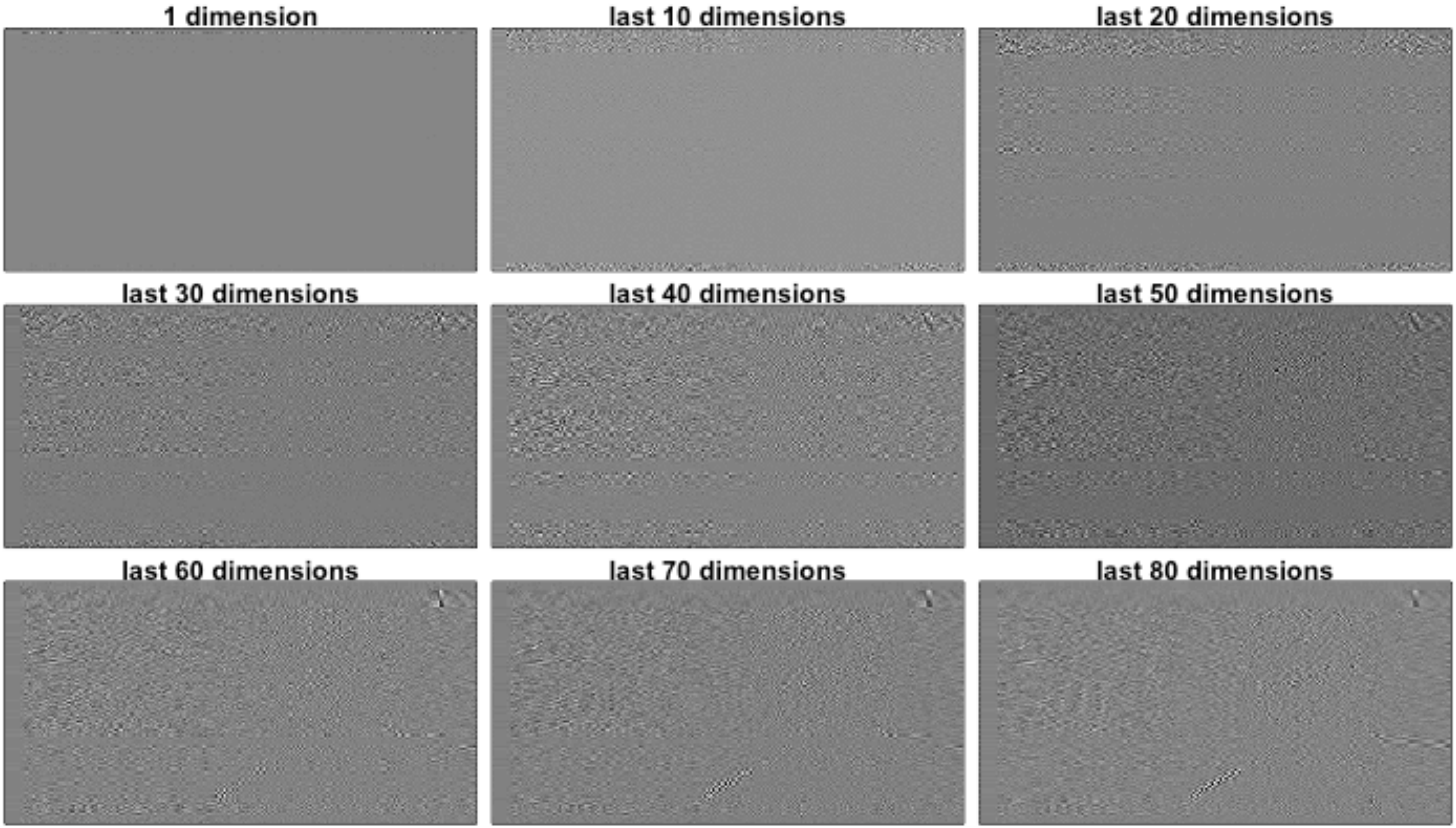
```
RappaRank_n =  U[,n:160] %*% diag(d[n:160]) %*% Vt[n:160,]
image(RappaRank_n,
      col=grey((0:1000)/1000),
      main=paste("last", (160-n),"dimensions"),
      xaxt = 'n',
      yaxt = 'n')
```



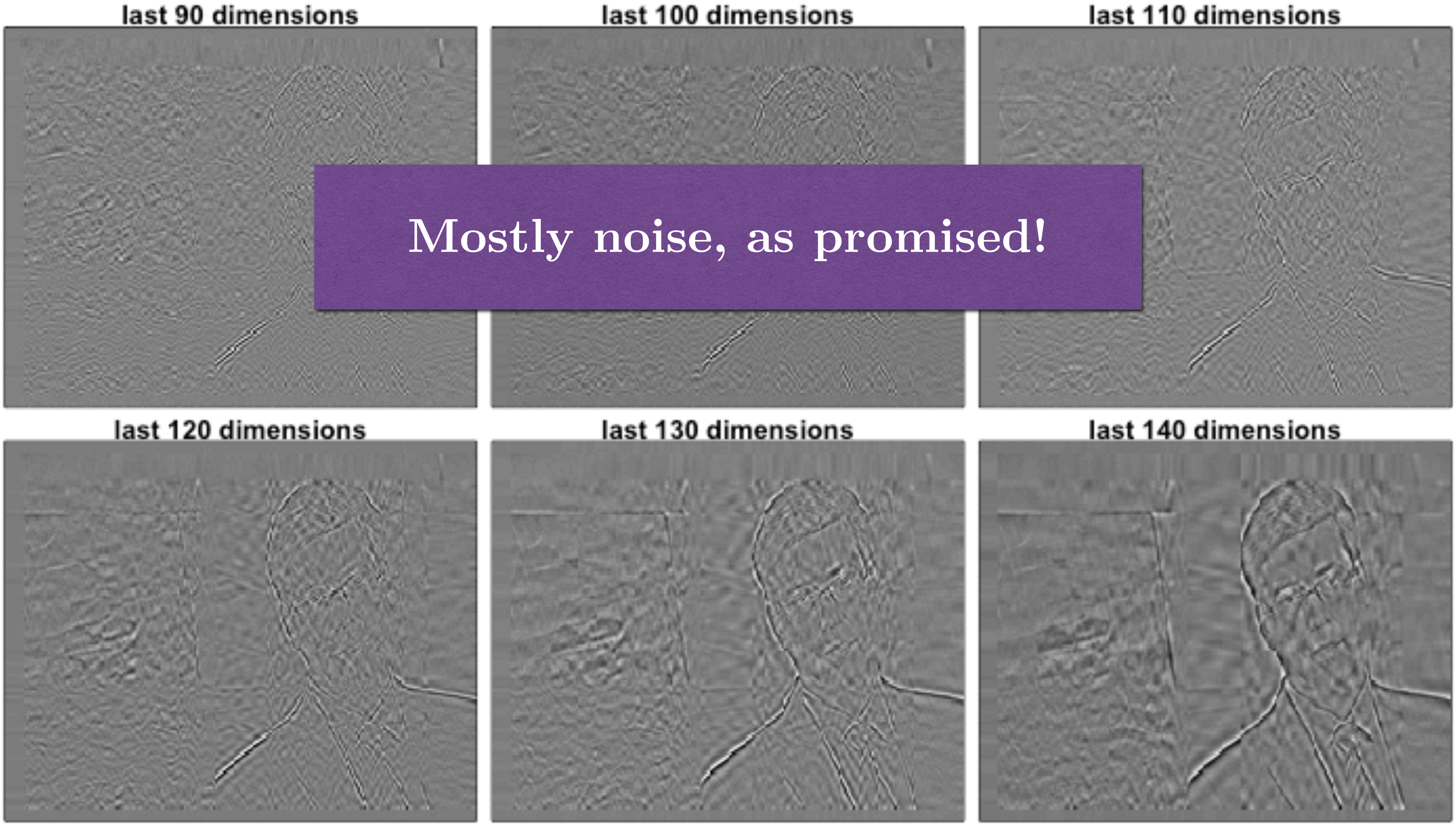
$$\mathbf{X} = \sigma_1 \mathbf{U}_1 \mathbf{V}_1^T + \sigma_2 \mathbf{U}_2 \mathbf{V}_2^T + \sigma_3 \mathbf{U}_3 \mathbf{V}_3^T + \ldots + \sigma_r \mathbf{U}_r \mathbf{V}_r^T$$

what did we **lose**?

What about the dropped components?



What about the dropped components?



Mostly noise, as promised!

What's the point?

- Orthogonal projections built on the theory of maximal variance don't tend to lie in the story they tell in the first few dimensions.
- Additional components can certainly help *resolve* the story - adding detail and clarity - but the theme remains the same.
- When you're reducing dimensionality of datasets, use the visual of 9-dimensional Dr. Rappa as an analogy to what you're seeing in the projection.