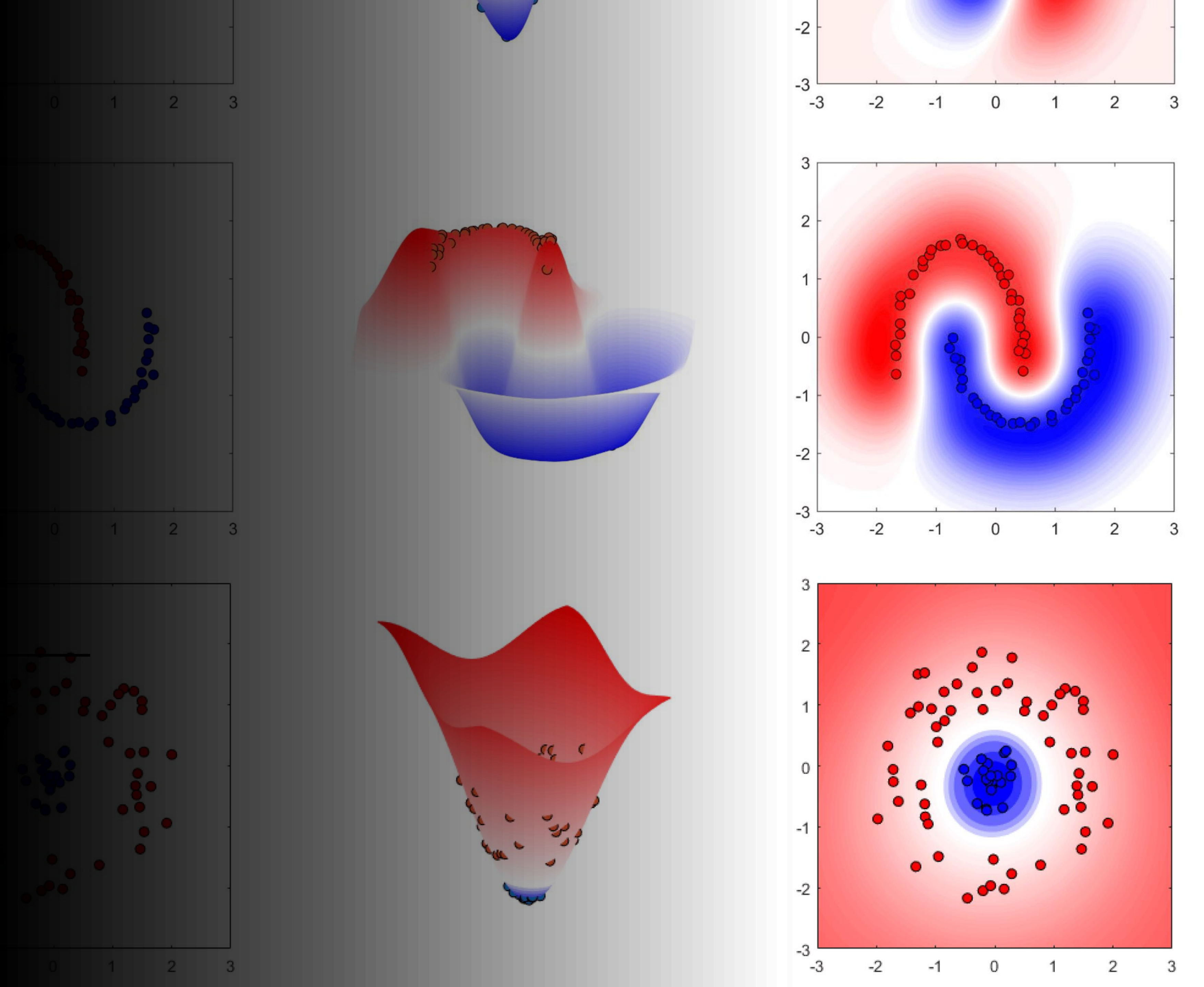
Revisiting The "Kernel Trick"



Revisiting rationale for the kernel trick

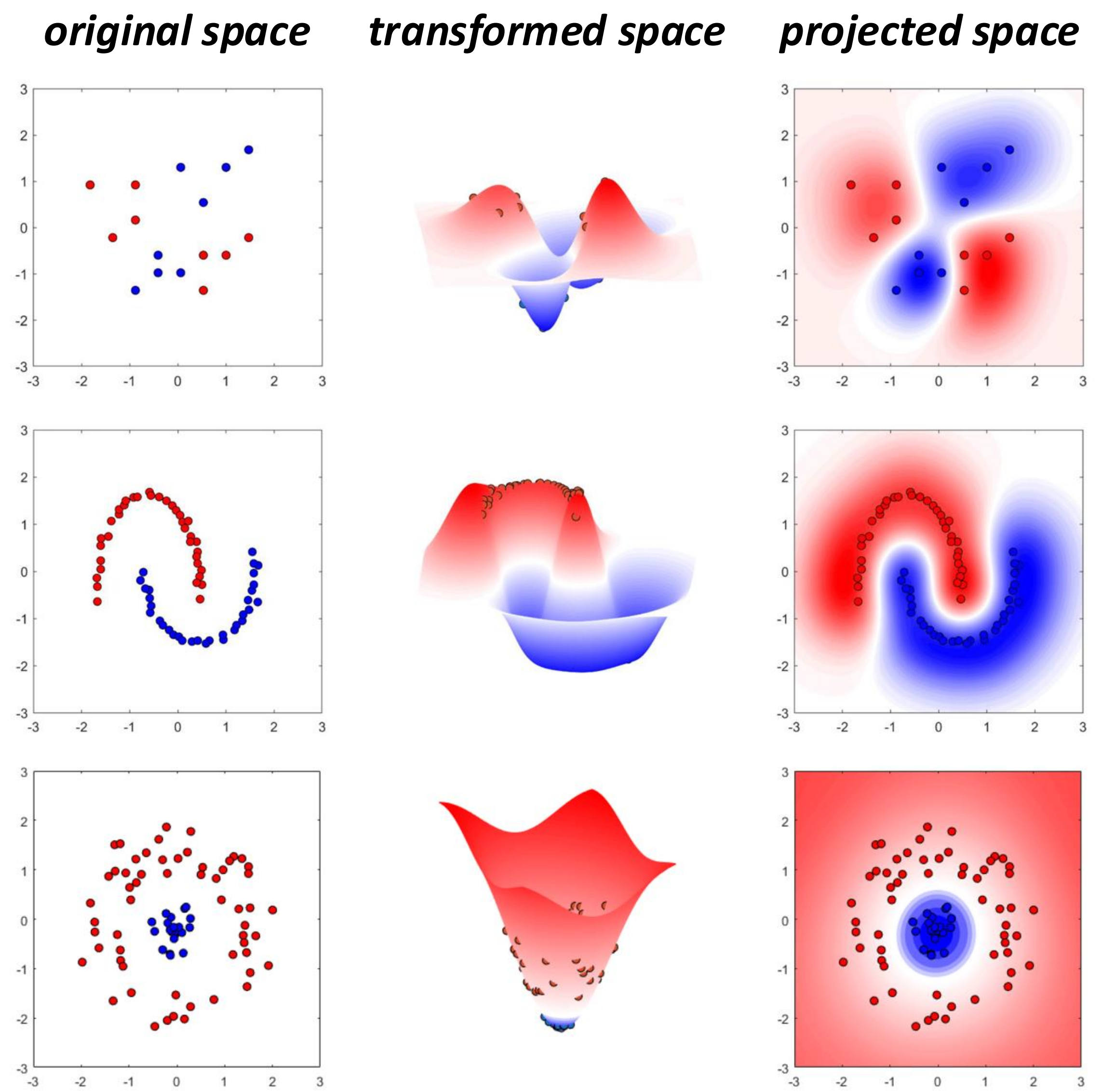
Rather than explicitly mapping the data into a higher-dimensional space (computationally expensive?), **the kernel trick** allows the algorithm to compute the **dot product of the data points** in this higher-dimensional space <u>*without ever performing the mapping*</u>.

This is achieved through a **kernel function**, which computes the inner product between two data points as if they were mapped into the higher-dimensional space.

Key benefits of the kernel trick:

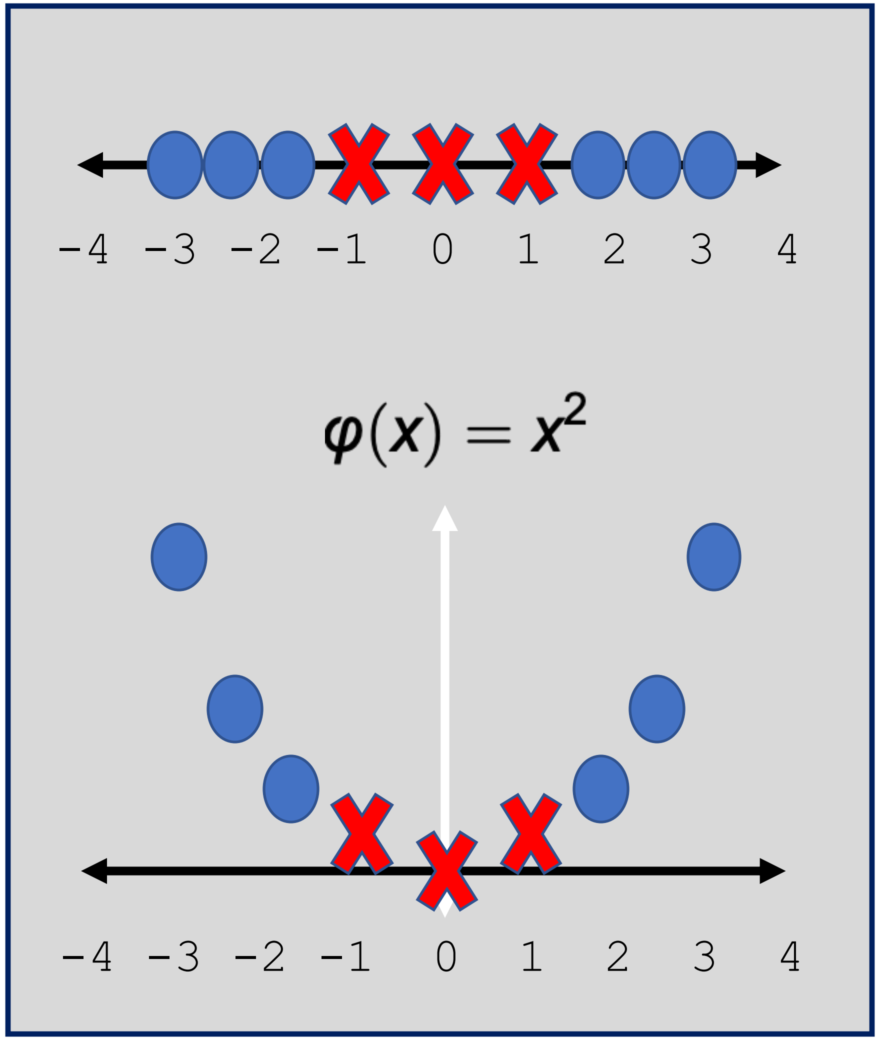
•**Non-linear separability**: It enables linear classifiers to handle non-linearly separable data *by implicitly transforming* the data into a higher-dimensional space.

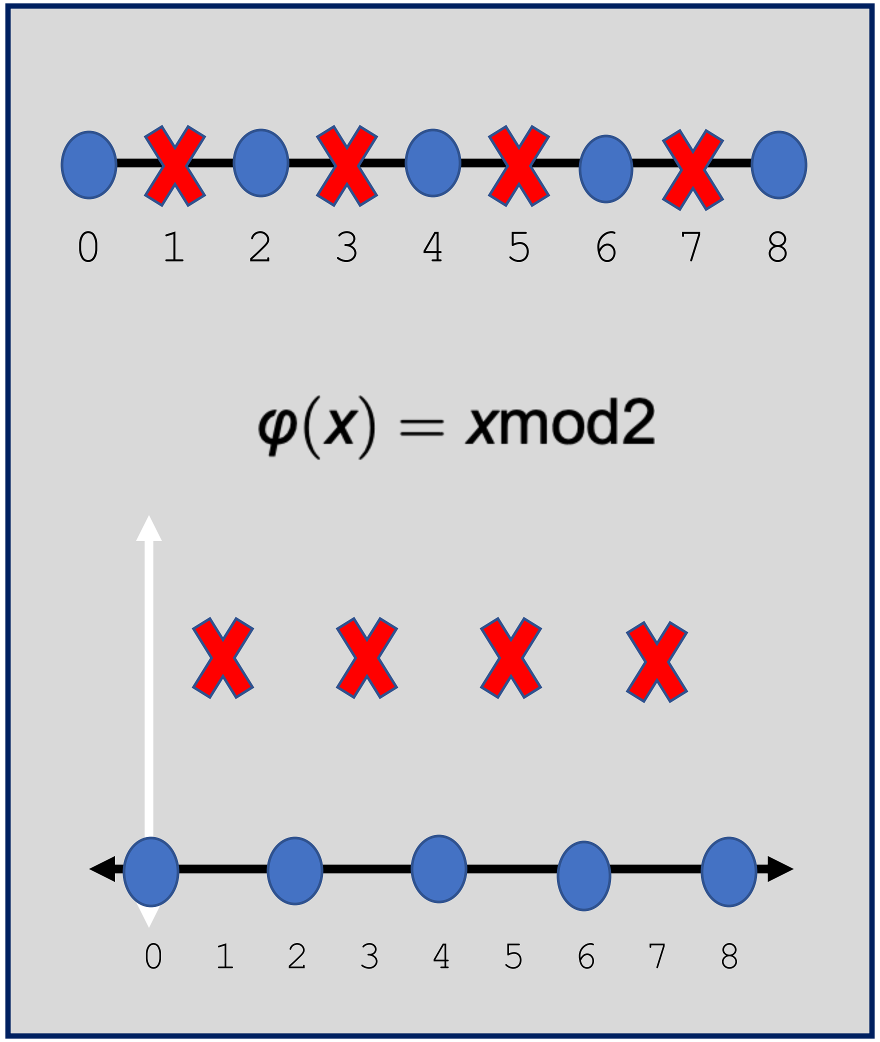
•**Computational efficiency**: It avoids the need to compute the transformation explicitly → faster and more efficient.



Utility for classification depends on the kernel

What transformations can make this data linearly separable?





wait...what is a kernel?

A kernel function computes the inner product of some set of vectors *in the transformed space*

vectors	$\mathbf{x}_i \in V$
transformation	$\mathbf{v} = \boldsymbol{\varphi}(\mathbf{x}_i) \in \mathbb{R}^n$
kernel	$k(\mathbf{x}_1, \mathbf{x}_2) = \langle \boldsymbol{\varphi}(\mathbf{x}_1), \boldsymbol{\varphi}(\mathbf{x}_2) \rangle$

Demonstrative Example

Imagine we have two 2D vectors

$$\mathbf{x}_1 = (1, 2)$$
$$\mathbf{x}_2 = (2, 3)$$

Suppose we apply a quadratic transformation to 3D:

$$\varphi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

Computing the transformed vectors in the 3D space:

$$\varphi(\mathbf{x}_1) = (1^2, \sqrt{2} \cdot 1 \cdot 2, 2^2) = (1, 2.828, 4)$$
$$\varphi(\mathbf{x}_2) = (1^2, \sqrt{2} \cdot 2 \cdot 3, 3^2) = (4, 8.485, 9)$$

Computing the dot product of these transformed vectors:

$$\varphi(\mathbf{x}_1) \cdot \varphi(\mathbf{x}_2) \approx (1 \times 4) + (2.828 \times 8.485) + (4 \times 9) = 4 + 23.995 + 36 = 63.995$$

Demonstrative Example

Imagine we have two 2D vectors

$$\mathbf{x}_1 = (1, 2)$$
$$\mathbf{x}_2 = (2, 3)$$

Instead, using a polynomial kernel of degree 2:

$$k(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1 \cdot \mathbf{x}_2)^2$$

Applied to the given vectors:

$$\begin{aligned} k((1,2)),(2,3)) &= (2+6)^2 \\ &= 8^2 = 64 \end{aligned}$$