ADVANCES IN
COMPUTER
VISION

MODULE 1:
Geometry

REPRESENTATION THEORY FOR COMPUTER VISION

Prof. Vincent Sitzmann

Last time on 6.8300

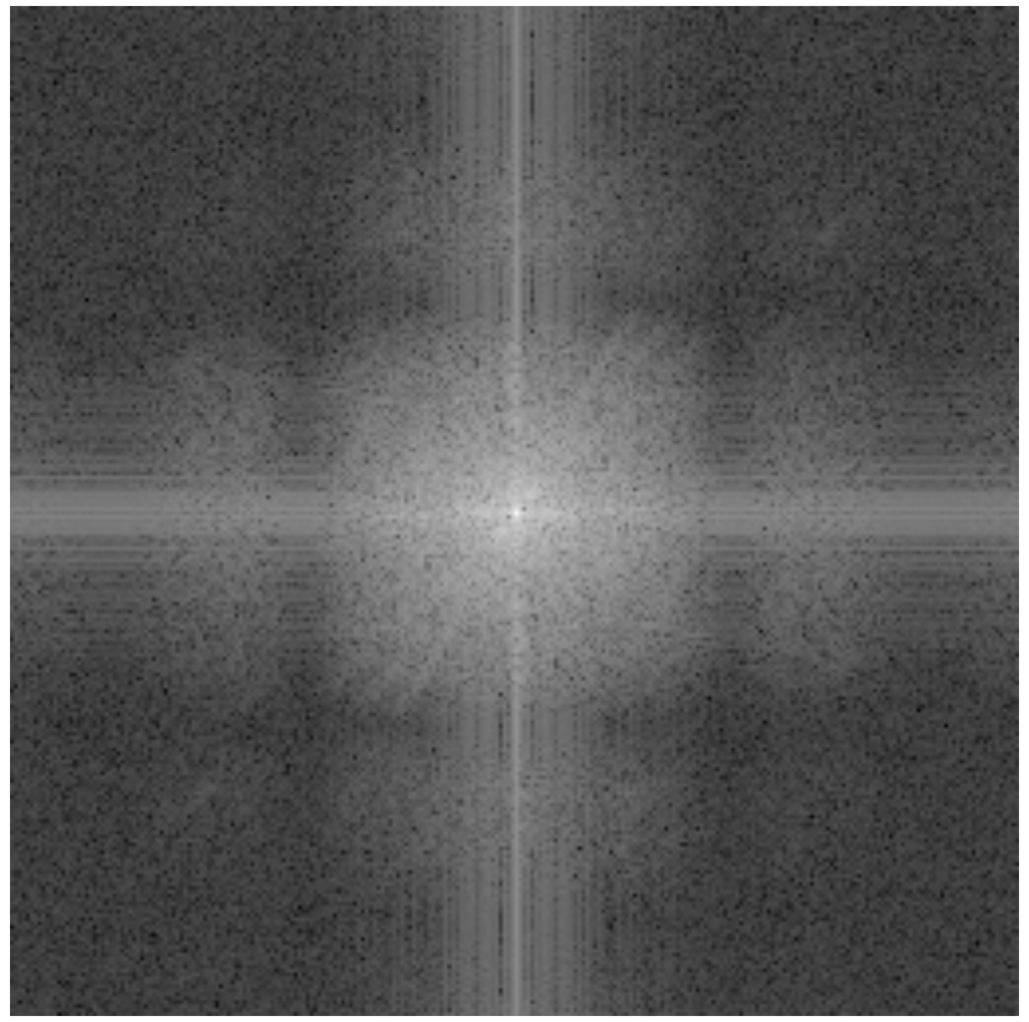
Linear Image Processing: A Fundamental Toolbox for Image Editing & Understanding

What is this arrow?

Edge Detection

Blurring

Frequency Analysis



Scale Spaces

The Discrete Fourier Transform

$$a_{2k} = \frac{2}{N}\sum_{n=0}^{N-1} f(n)\cos\left(\frac{2\pi kn}{N}\right), \qquad a_{2k+1} = \frac{2}{N}\sum_{n=0}^{N-1} f(n)\sin\left(\frac{2\pi kn}{N}\right),$$

$$\mathbf{F} = \qquad\qquad \mathbf{f}$$

The Discrete Fourier Transform

$$a_{2k} = \frac{2}{N}\sum_{n=0}^{N-1} f(n)\cos\left(\frac{2\pi kn}{N}\right), \qquad a_{2k+1} = \frac{2}{N}\sum_{n=0}^{N-1} f(n)\sin\left(\frac{2\pi kn}{N}\right),$$

- For each n we compute the dot-product between the discrete signal x and the cosine and sine.

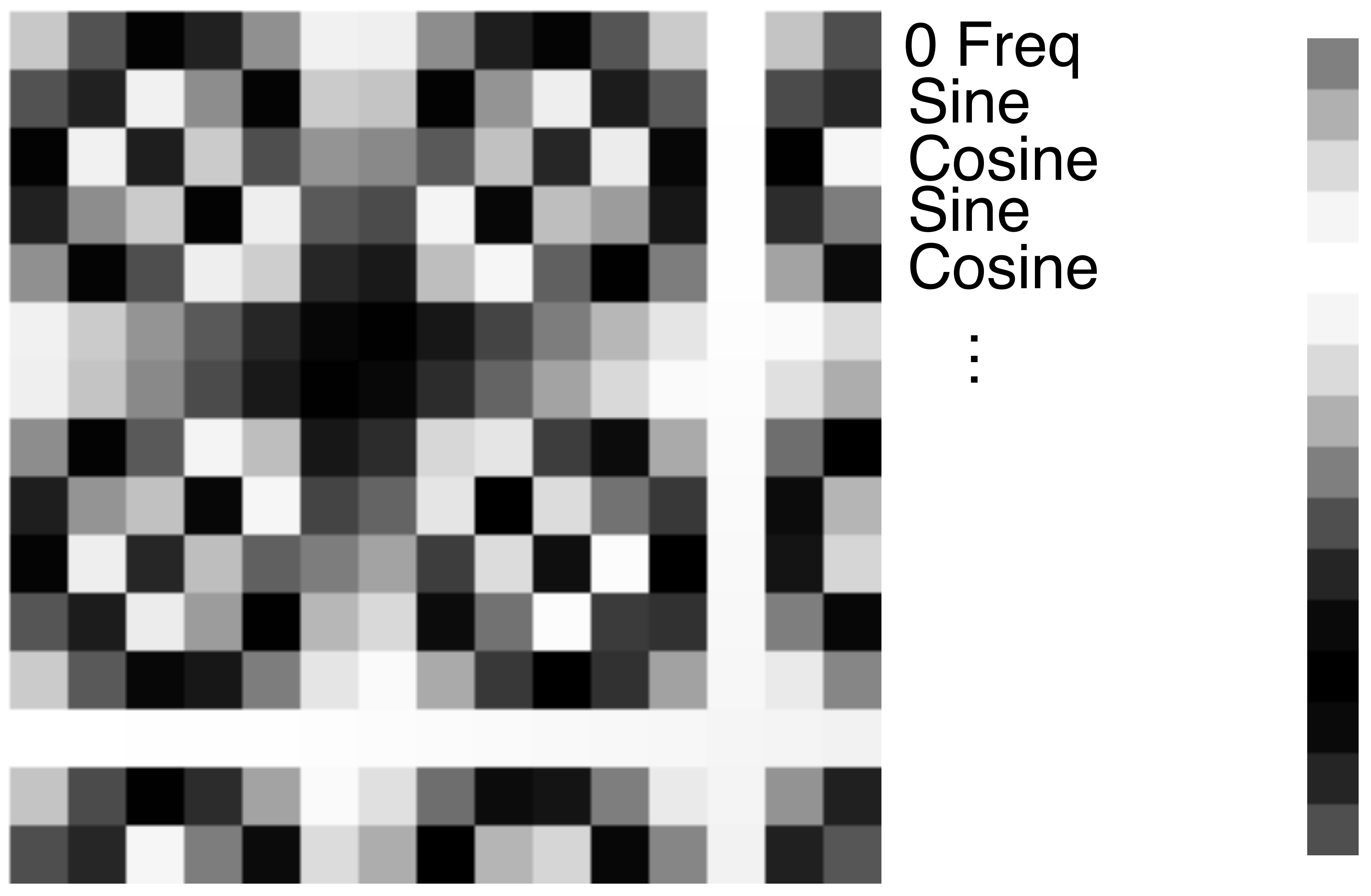
$$\mathbf{F} = \qquad\qquad \mathbf{f}$$

The Discrete Fourier Transform

$$a_{2k} = \frac{2}{N} \sum_{n=0}^{N-1} f(n) \cos\left(\frac{2\pi kn}{N}\right), \qquad a_{2k+1} = \frac{2}{N} \sum_{n=0}^{N-1} f(n) \sin\left(\frac{2\pi kn}{N}\right),$$

- For each n we compute the dot-product between the discrete signal x and the cosine and sine.
- This is just a matrix-vector multiplication!

$\mathbf{F} =$



$\mathbf{f}$

Why do we care about the basis of our image space?

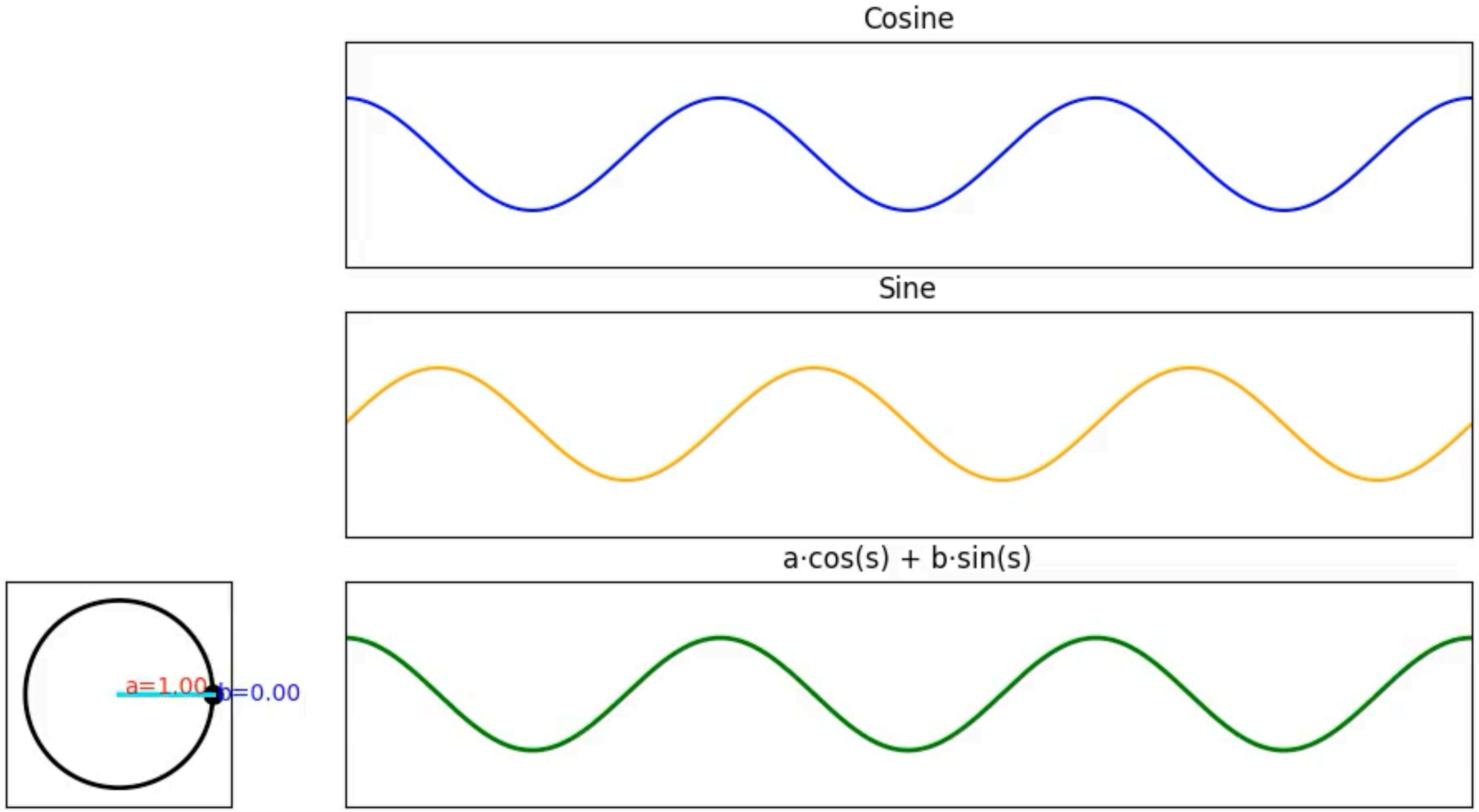
The coefficients of an image for different bases are a simple form of *image representation,* making some information about the image easily accessible (locality, spatial frequency, …)

Some operations are simpler in a different basis, computationally more efficient, or easier to interpret.

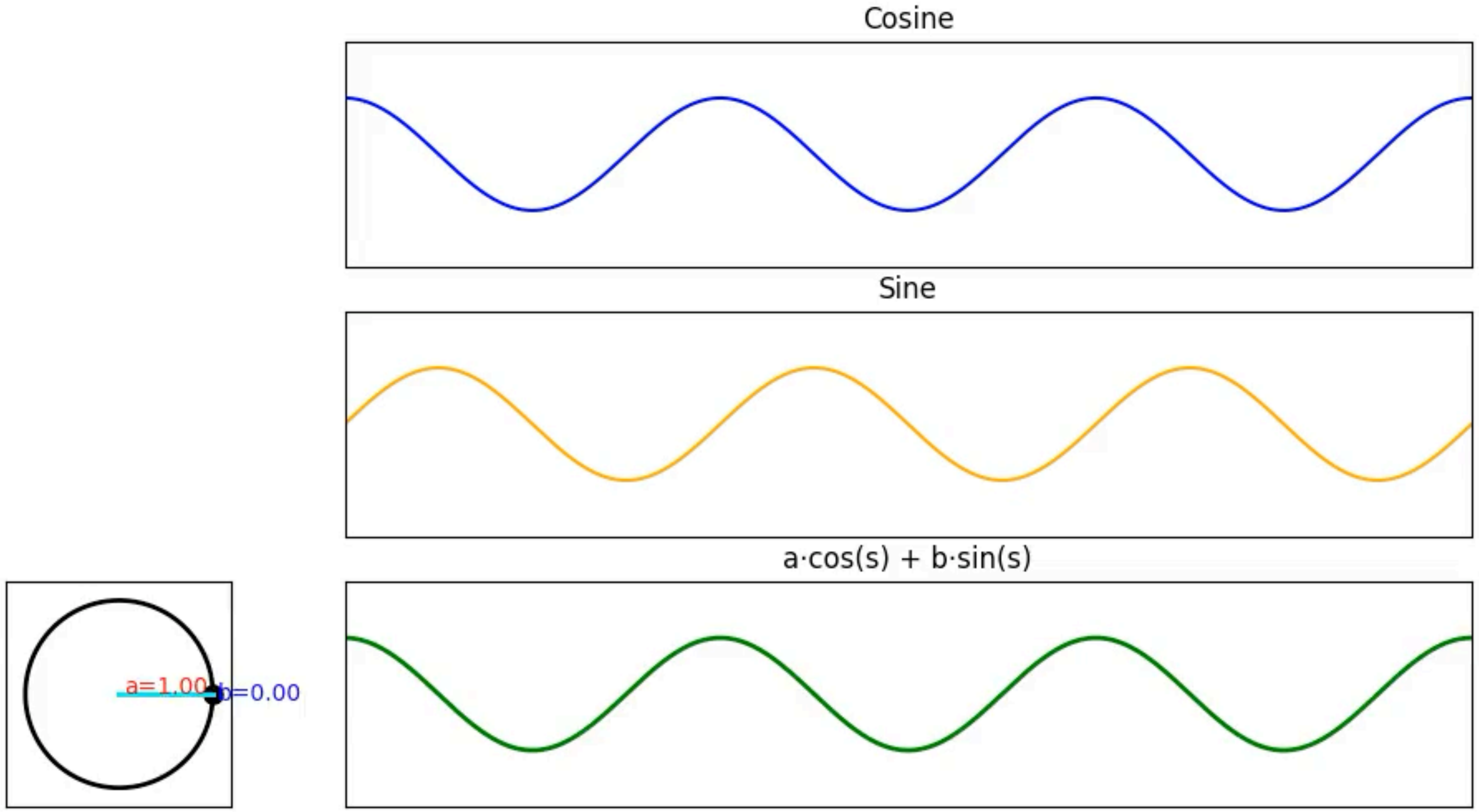
Let's look at some linear operators and discuss how they work in the pixel basis and the Fourier basis!

Motivation

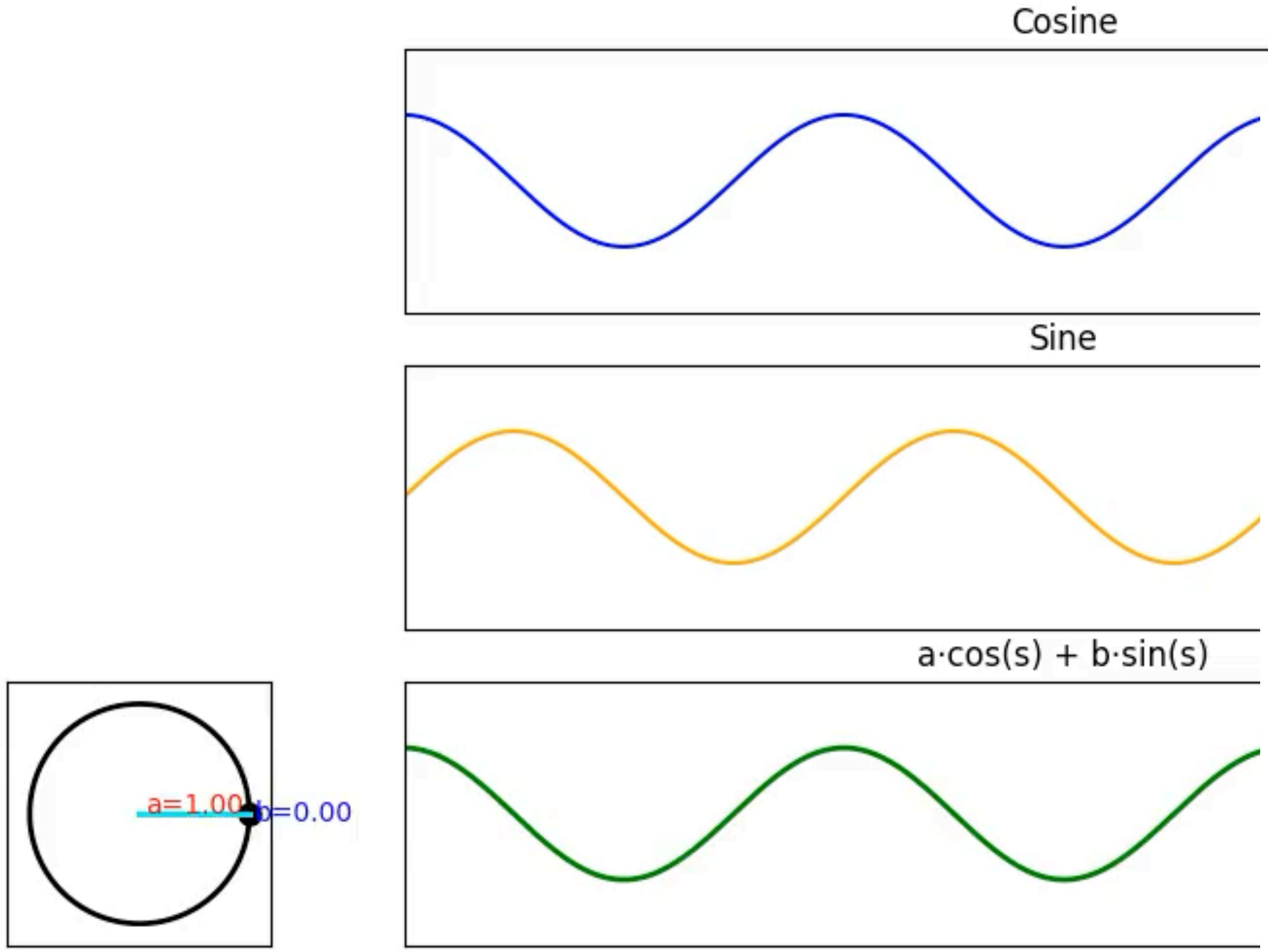
Something is up with shifts & the FT…

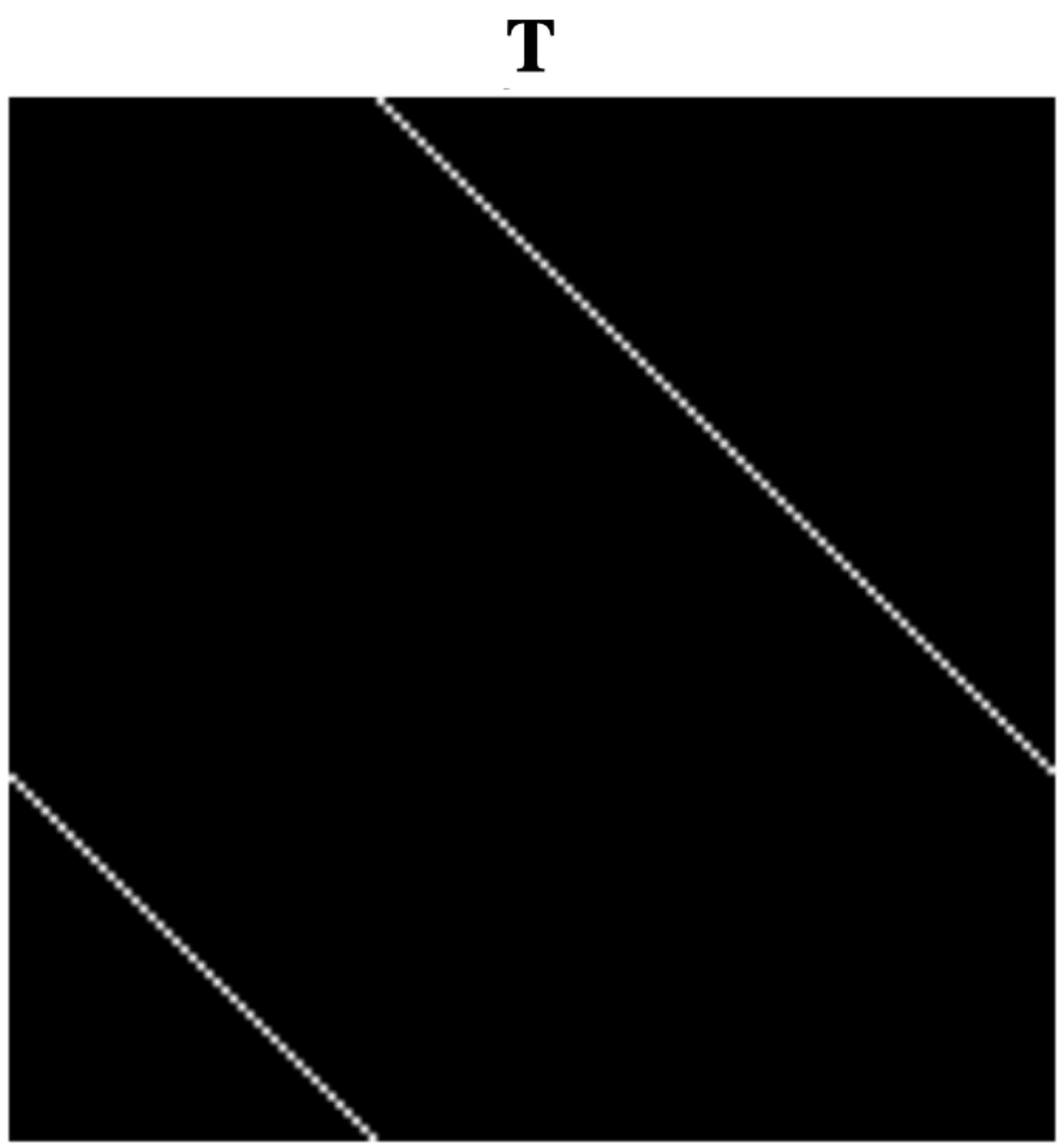


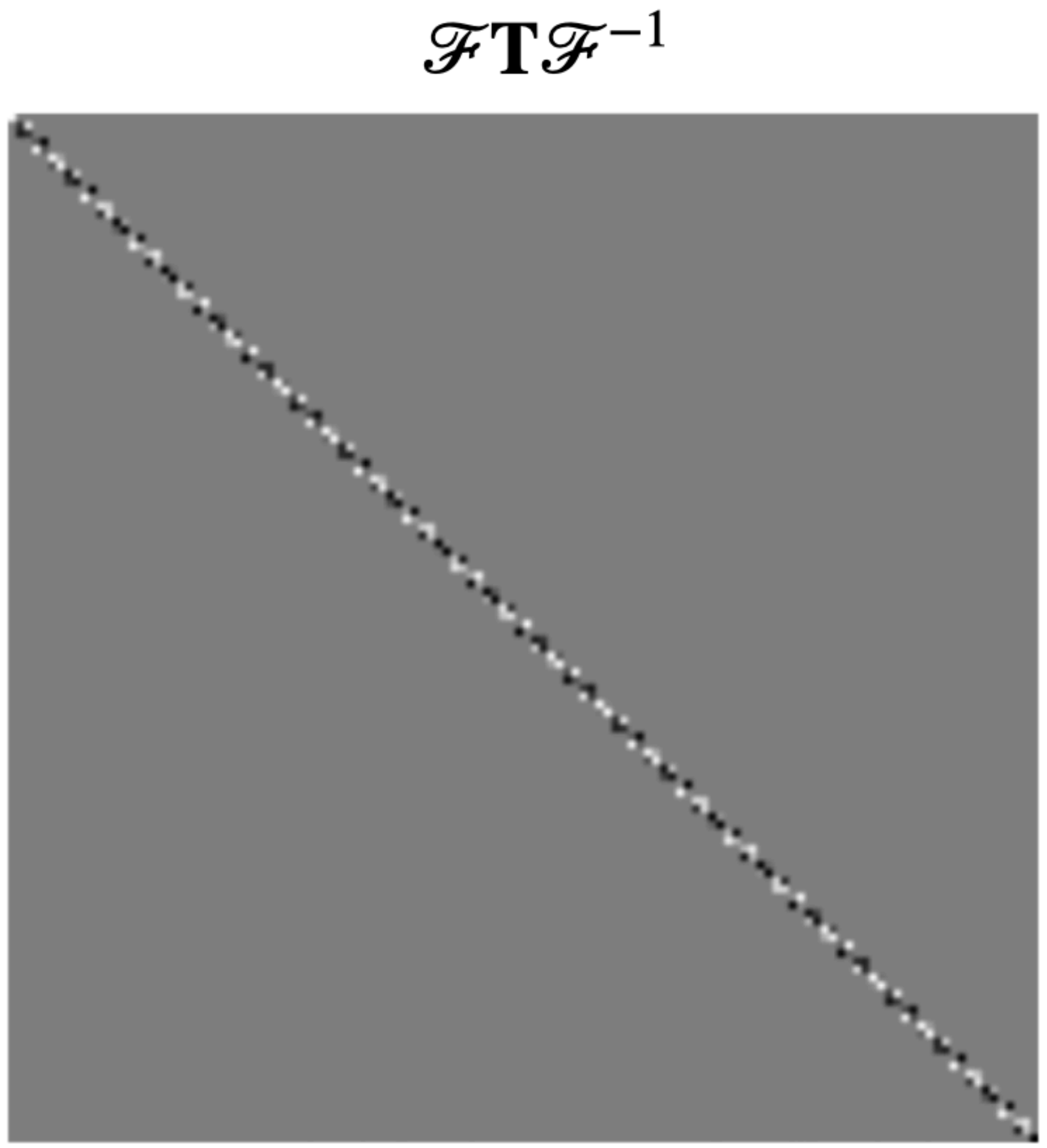
Something is up with shifts & the FT…



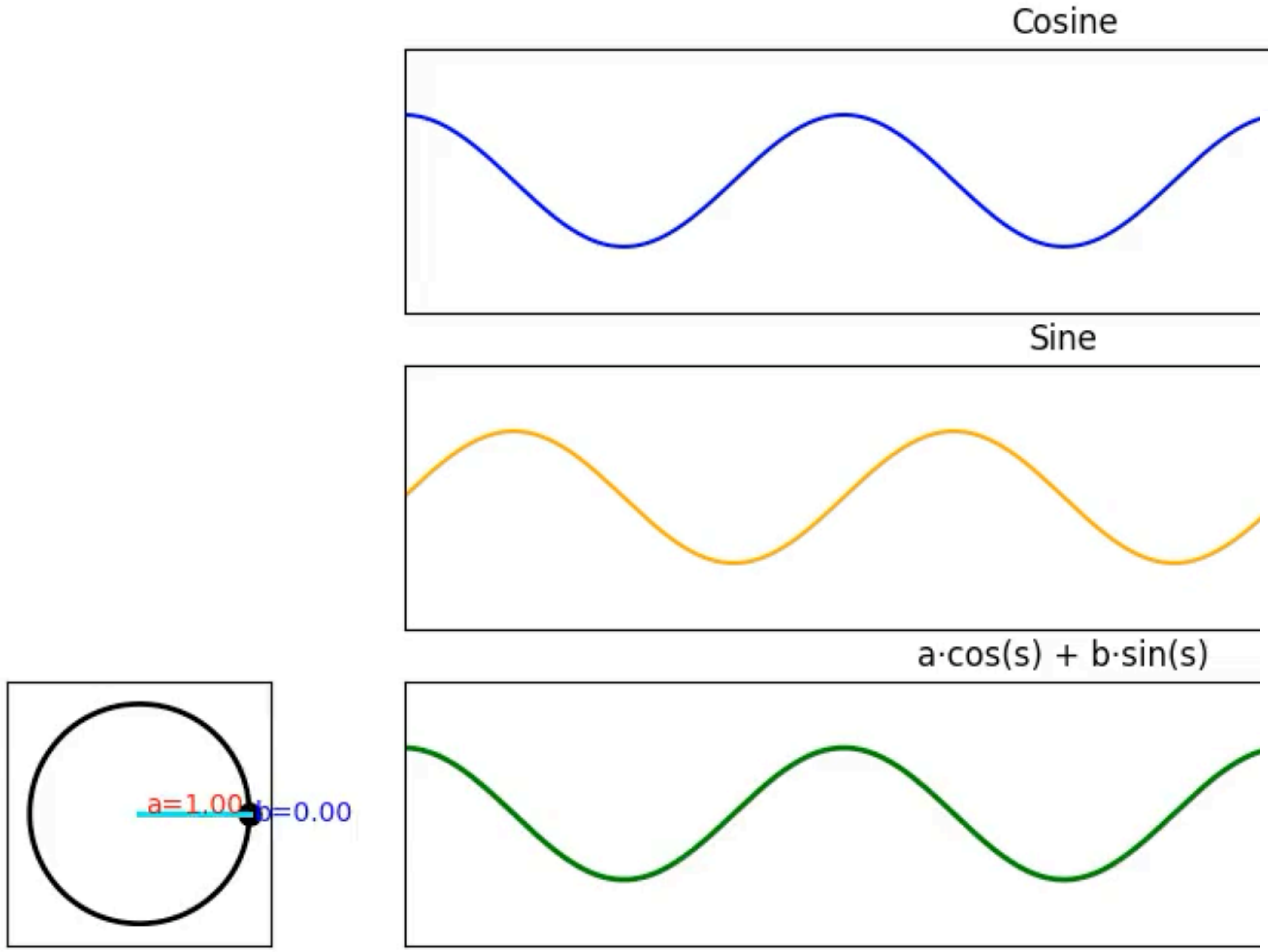
Something is up with shifts & the FT…

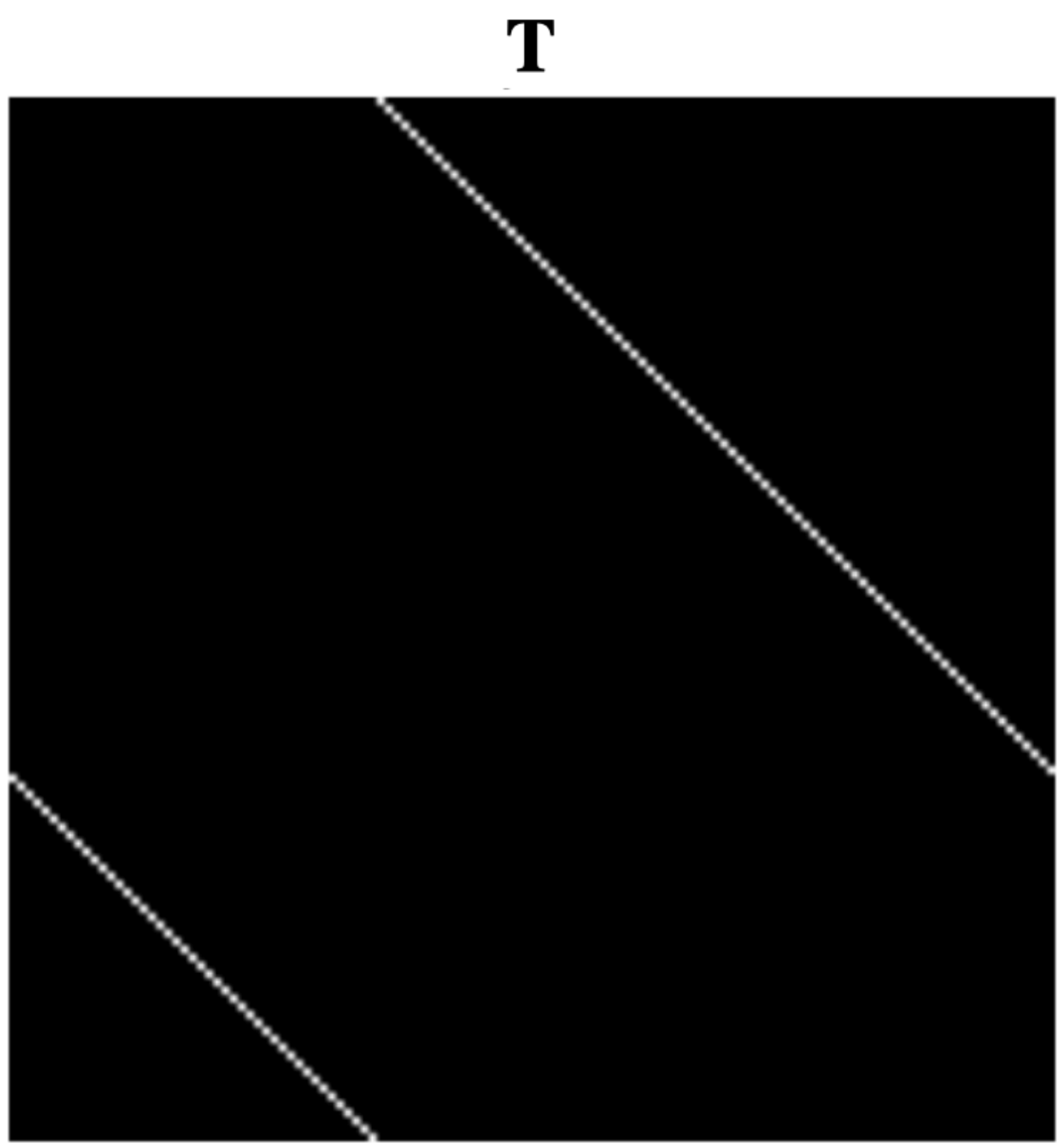


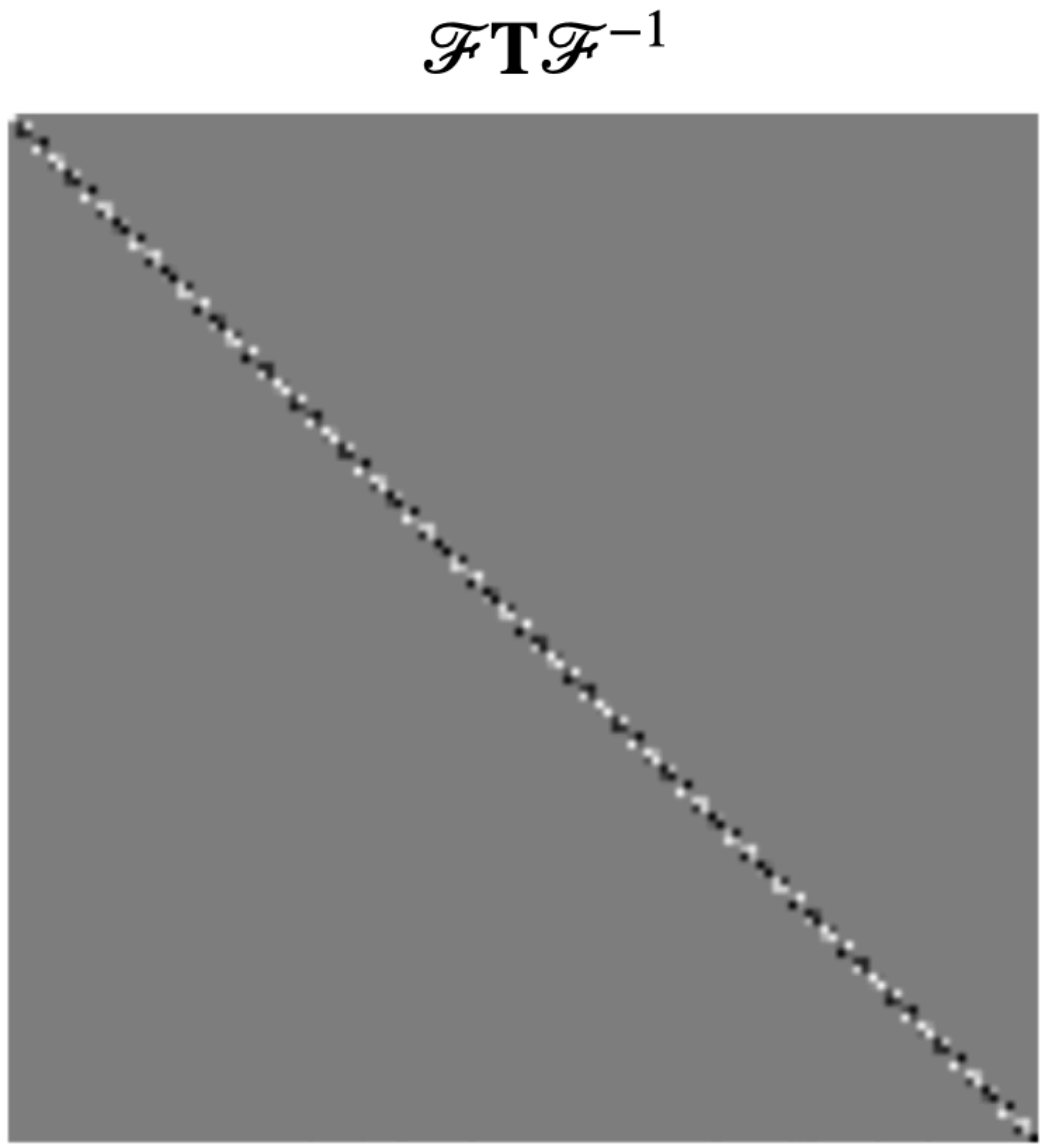




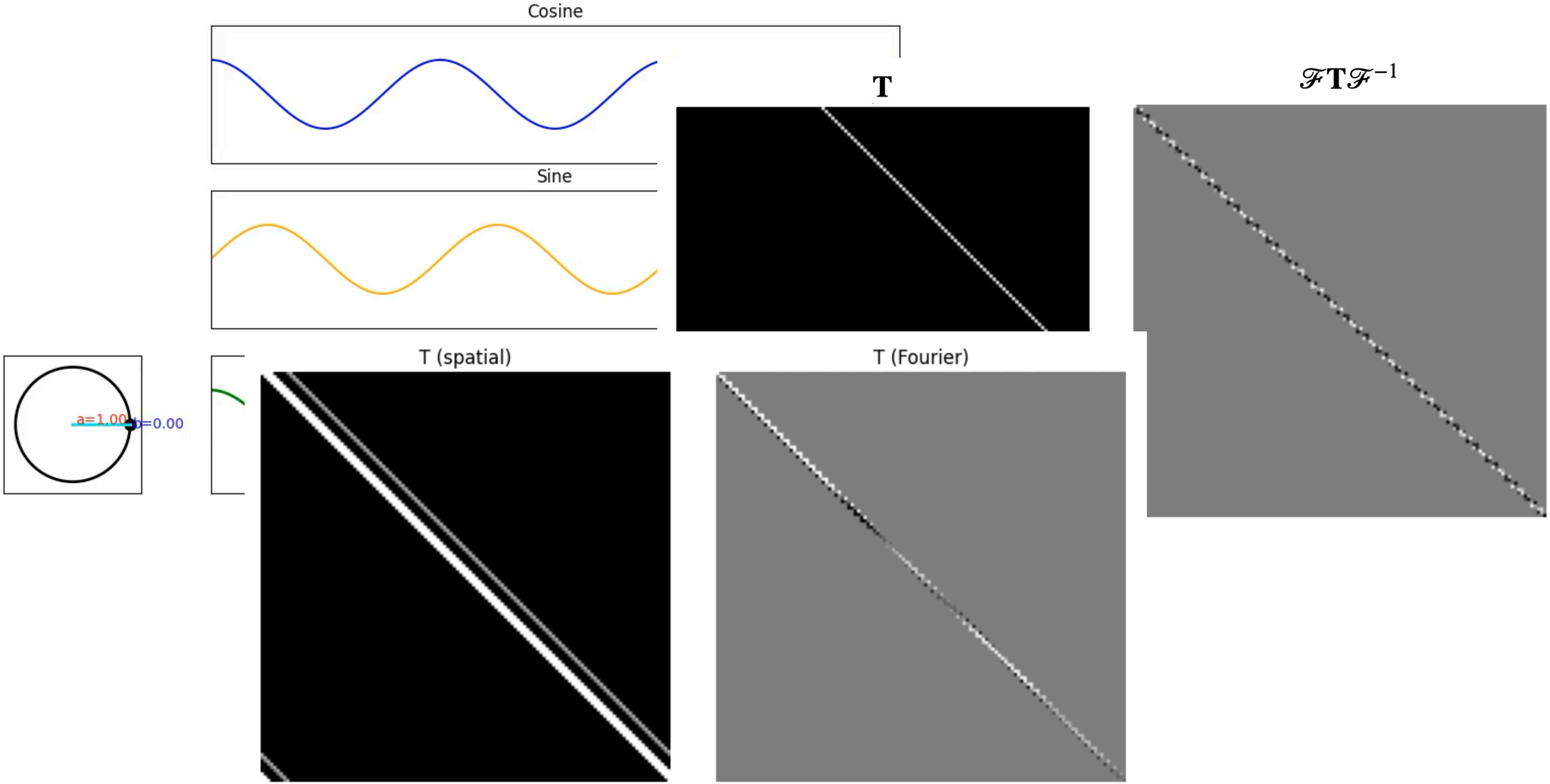
Something is up with shifts & the FT…



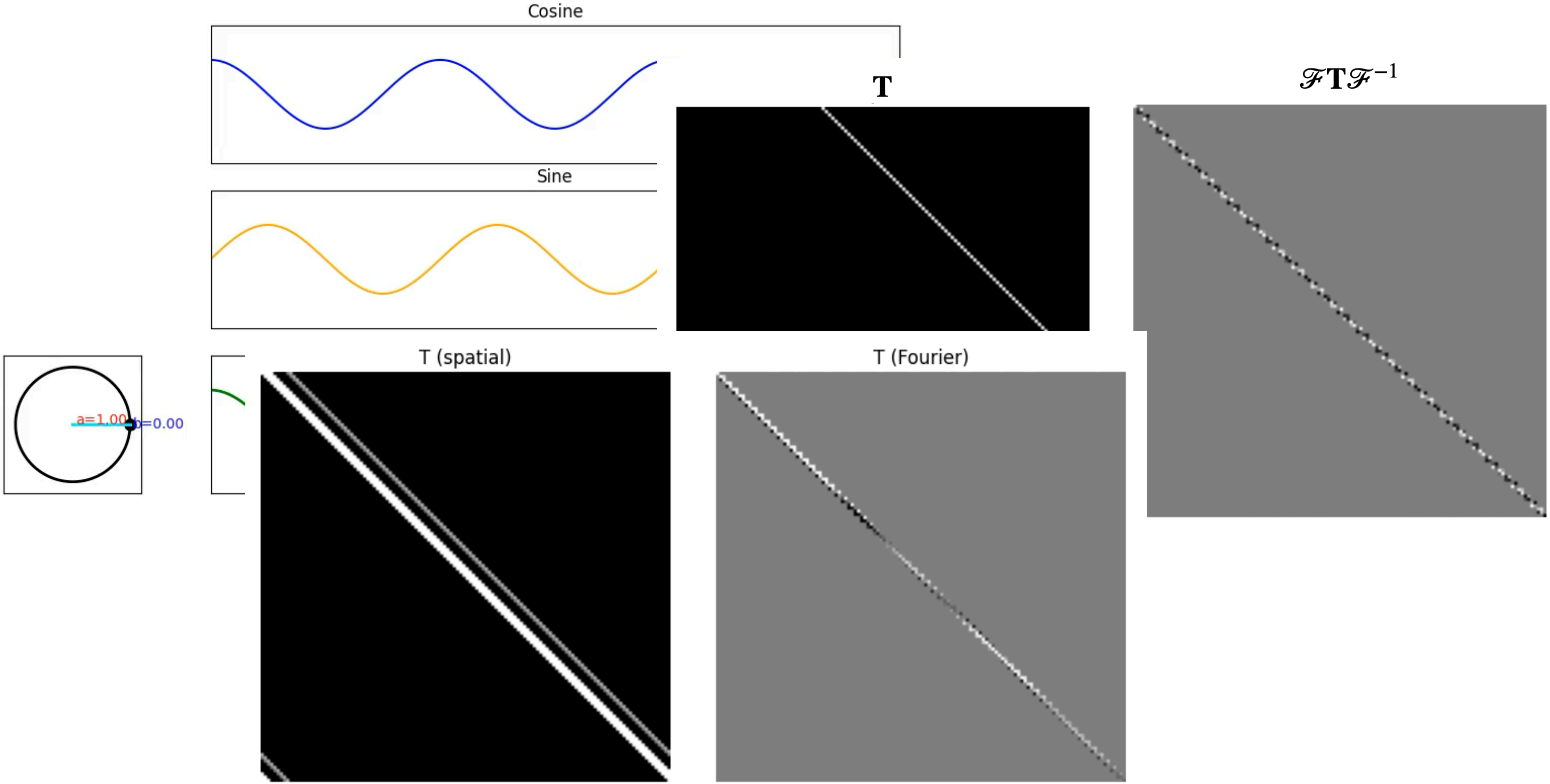




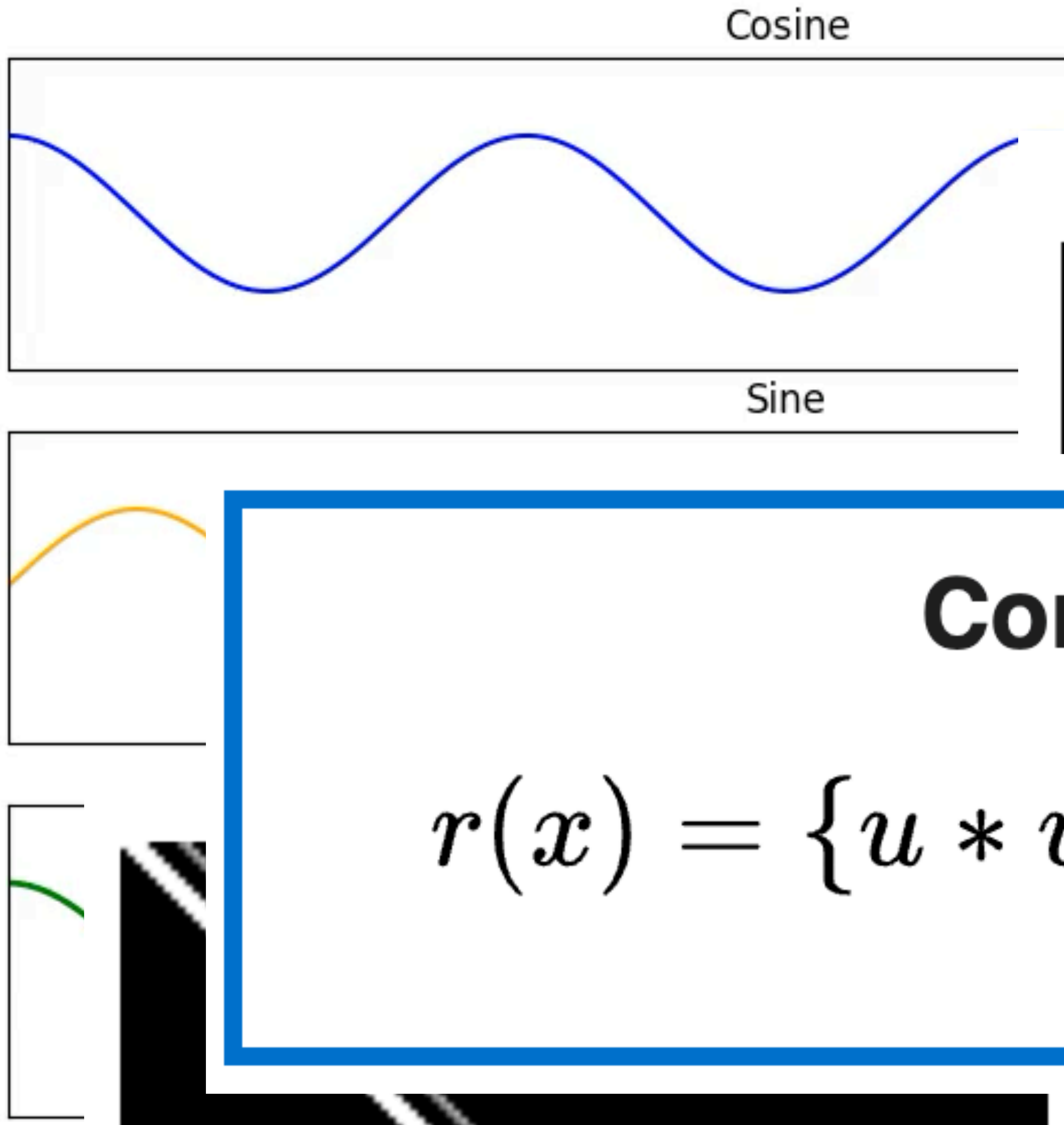
Something is up with shifts & the FT…

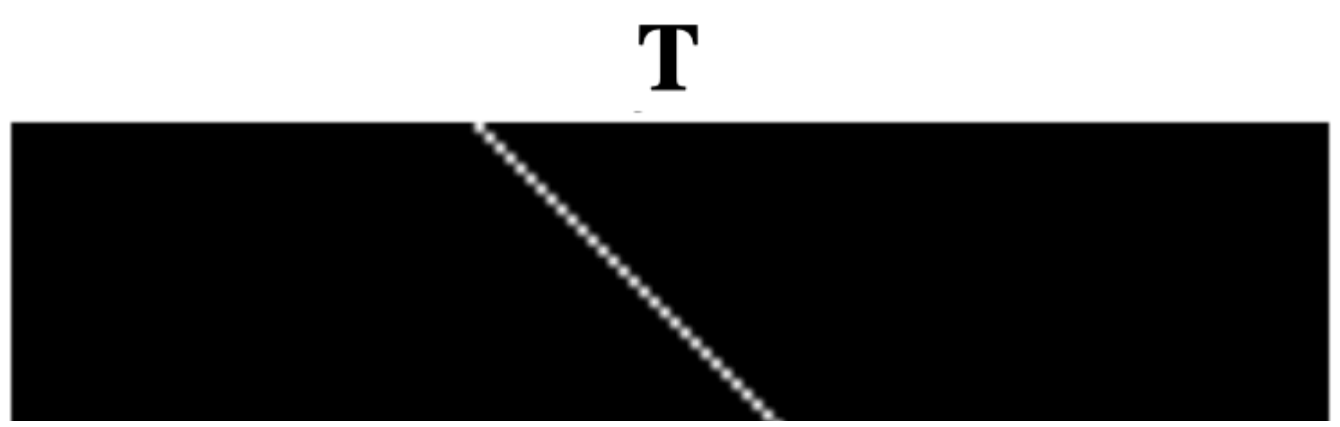


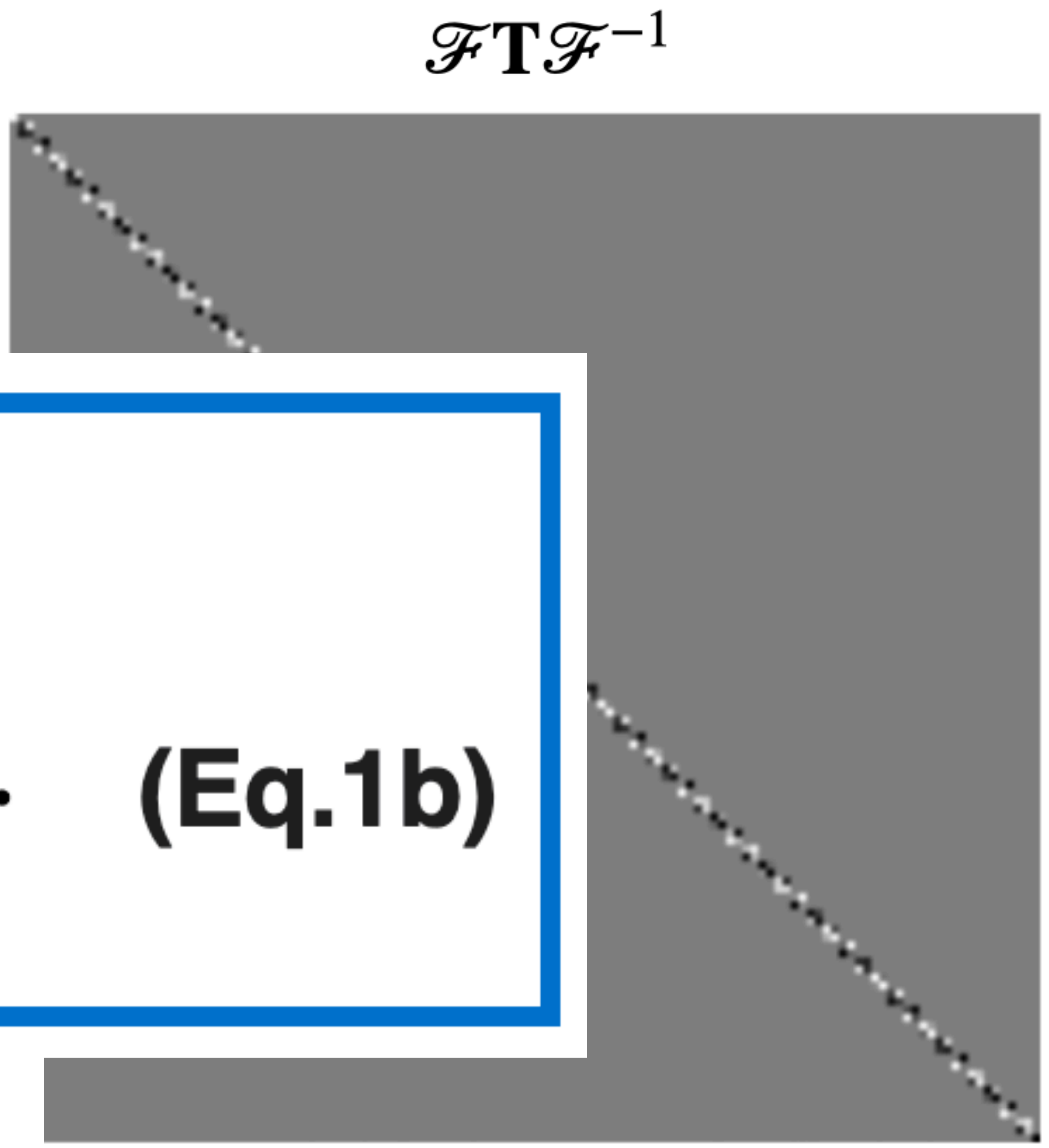
Something is up with shifts & the FT…



Something is up with shifts & the FT…

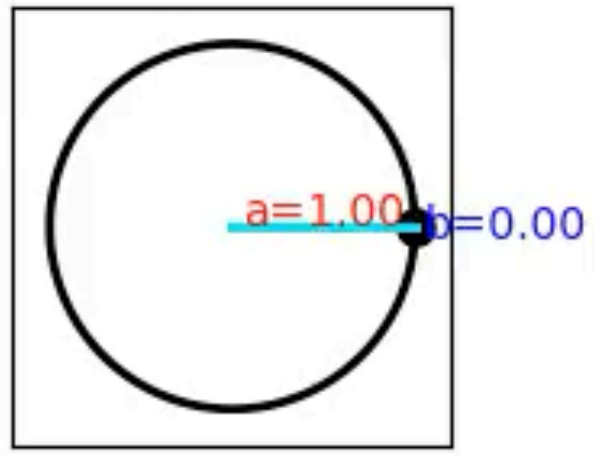


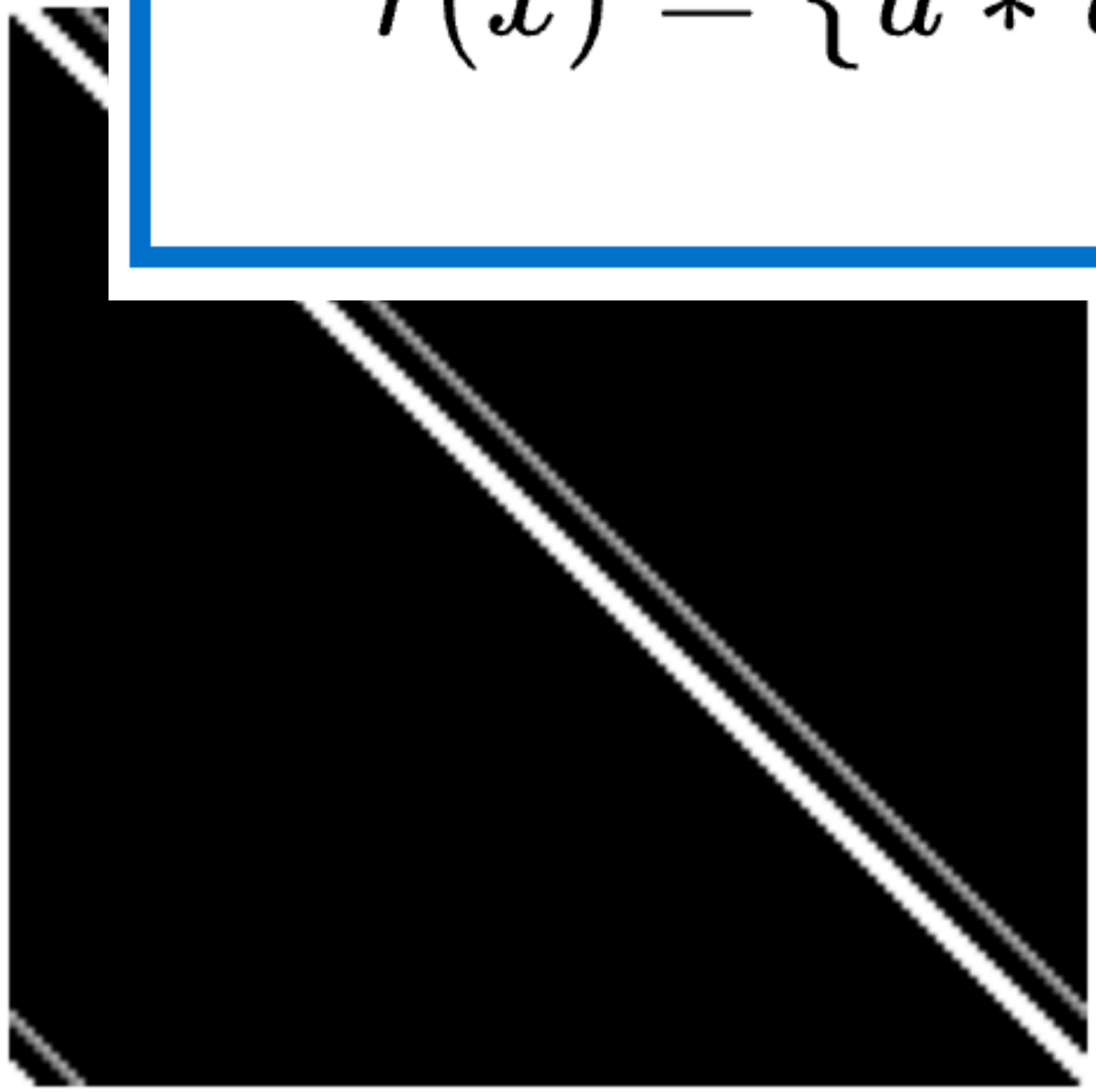




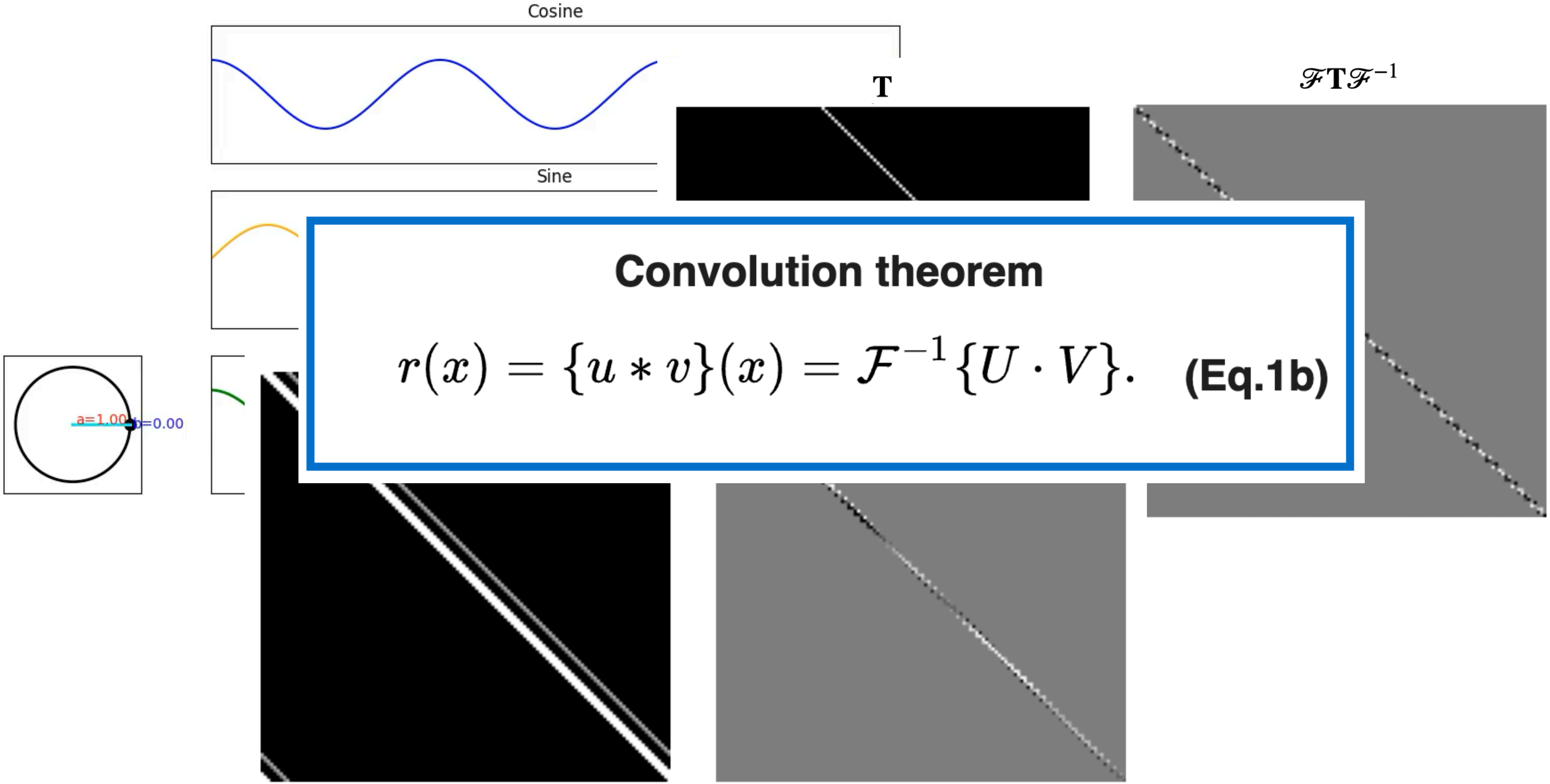
Convolution theorem

$$r(x) = \{u * v\}(x) = \mathcal{F}^{-1}\{U \cdot V\}. \quad \textbf{(Eq.1b)}$$





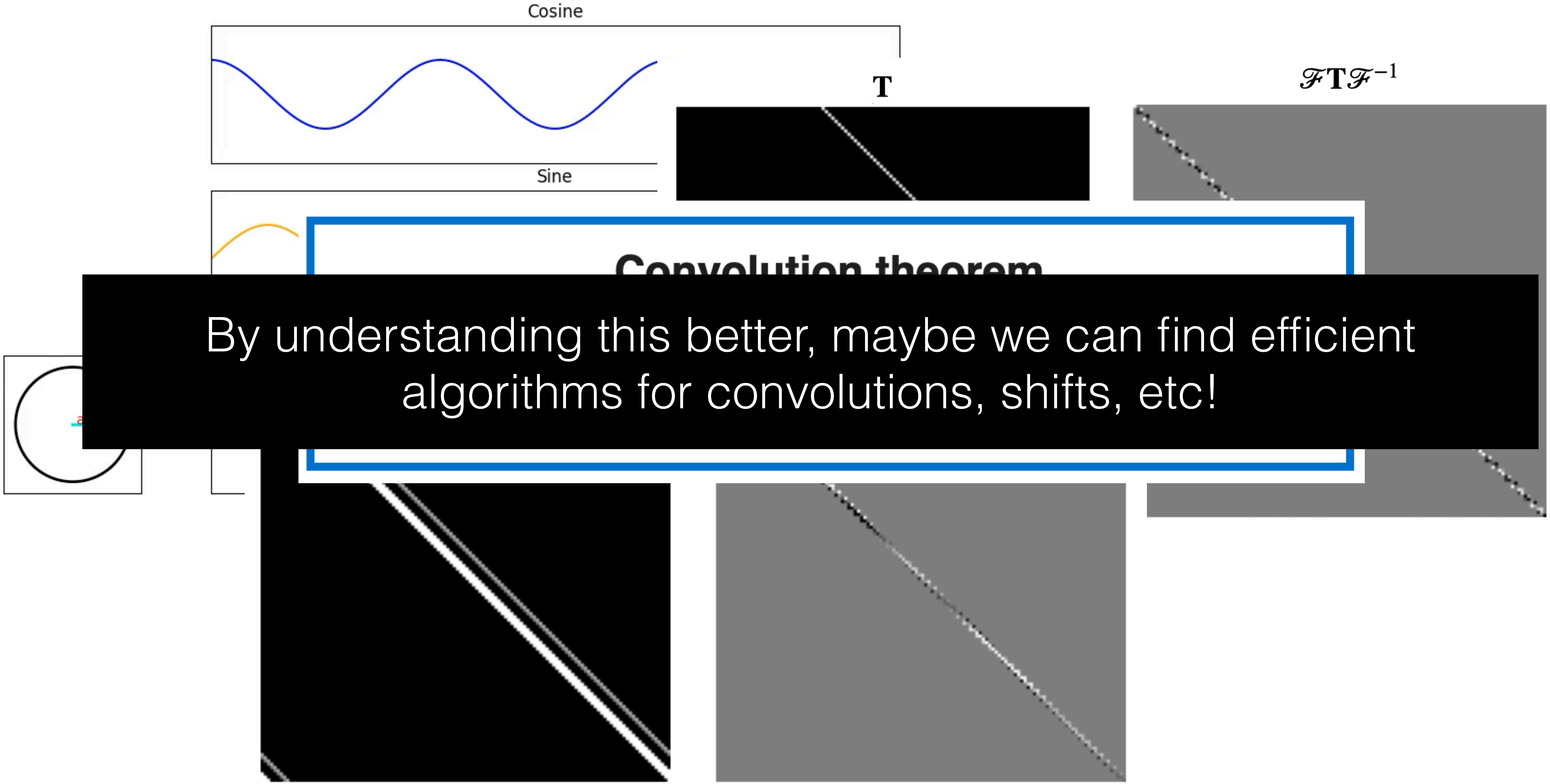
Something is up with shifts & the FT…



Convolution theorem

$$r(x) = \{u * v\}(x) = \mathcal{F}^{-1}\{U \cdot V\}. \quad \textbf{(Eq.1b)}$$

Something is up with shifts & the FT…



By understanding this better, maybe we can find efficient algorithms for convolutions, shifts, etc!

Geometric guarantees (equivariance)

CNNs are translation equivariant

Via convolutions

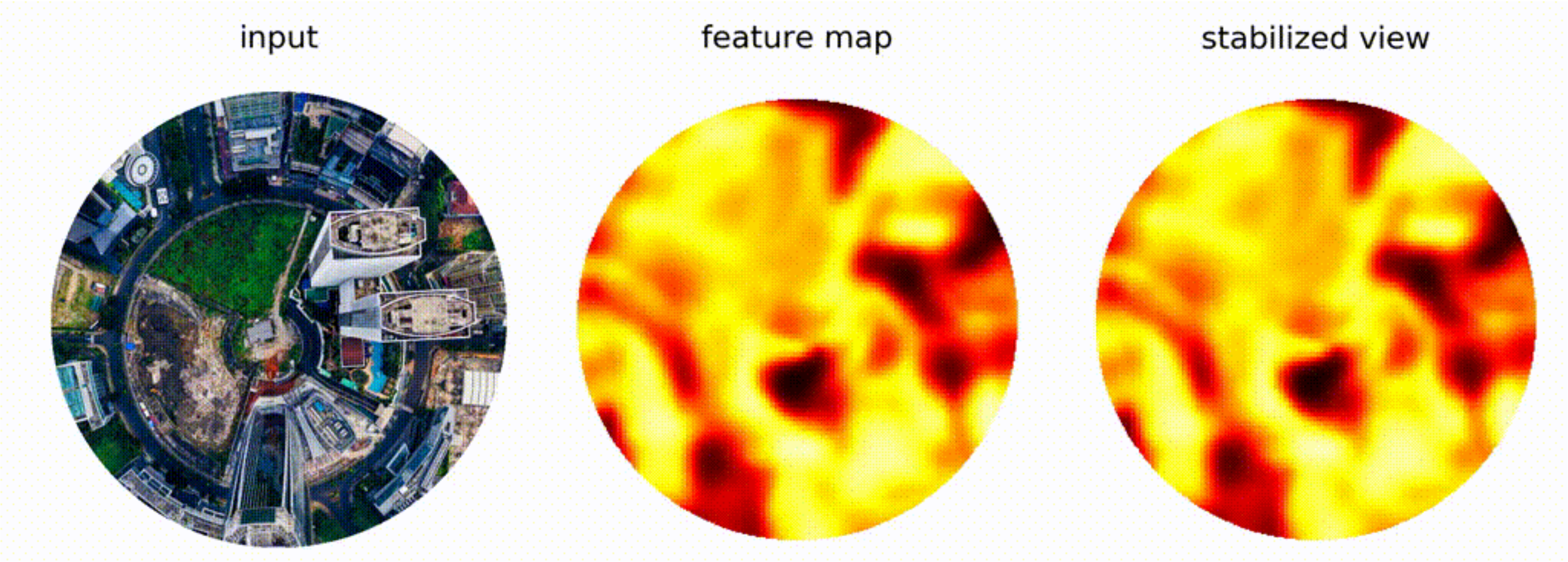
Geometric guarantees (equivariance)

CNNs are translation equivariant

Via convolutions

Geometric guarantees (equivariance)

CNN

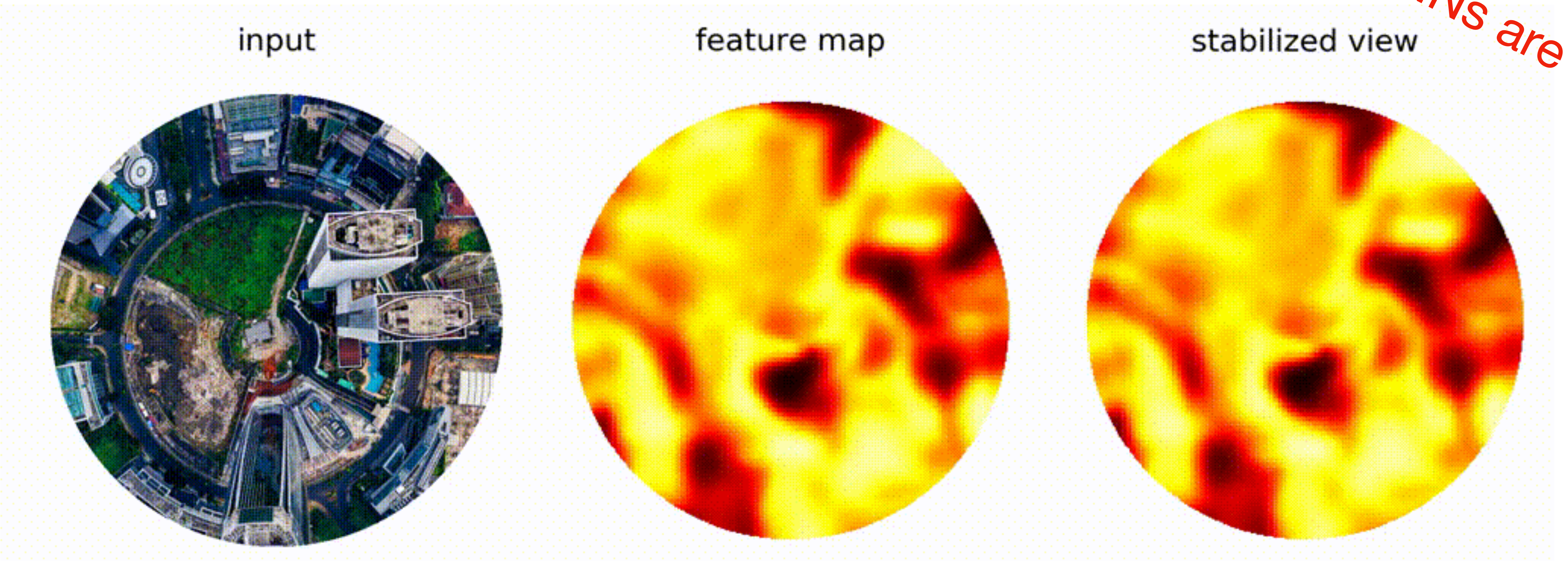


Figures source:
https://github.com/QUVA-Lab/e2cnn

Slide curtesy of Erik Bekkers from UVA Deep Learning II Course

Geometric guarantees (equivariance)

CNN



CNNs are not rotation equivariant

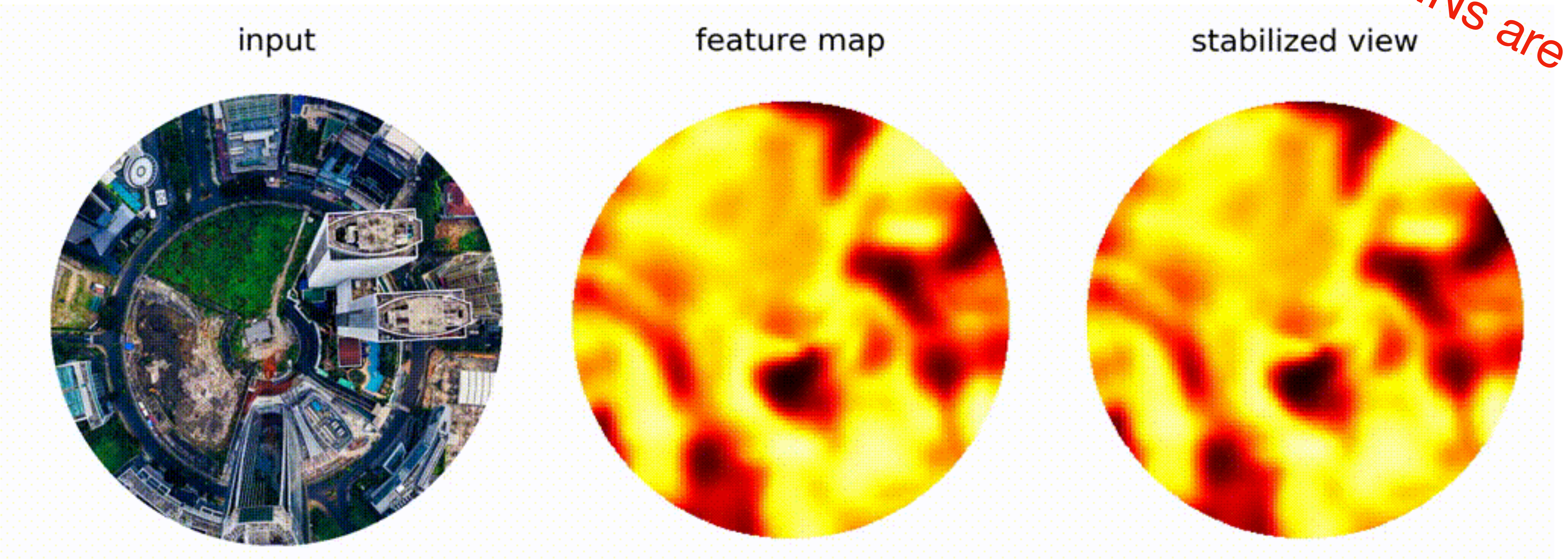
Figures source:
https://github.com/QUVA-Lab/e2cnn

Slide curtesy of Erik Bekkers from UVA Deep Learning II Course

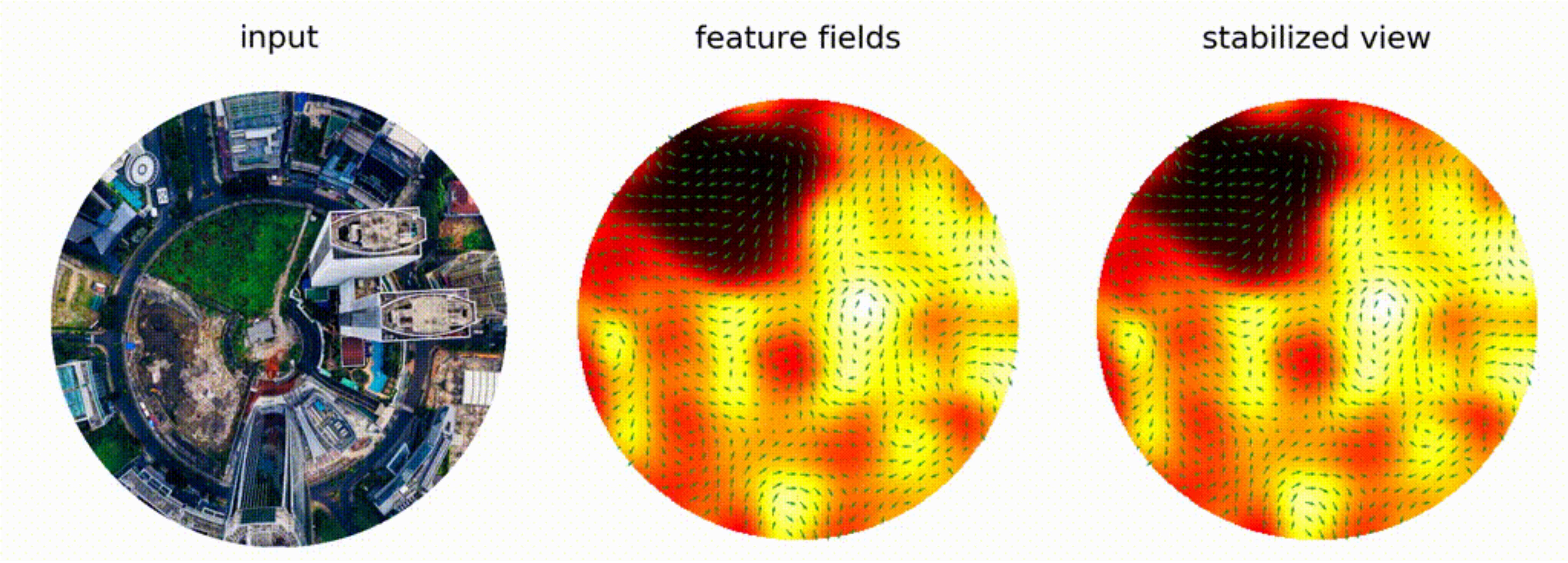
Geometric guarantees (equivariance)

CNNs are not rotation equivariant

CNN



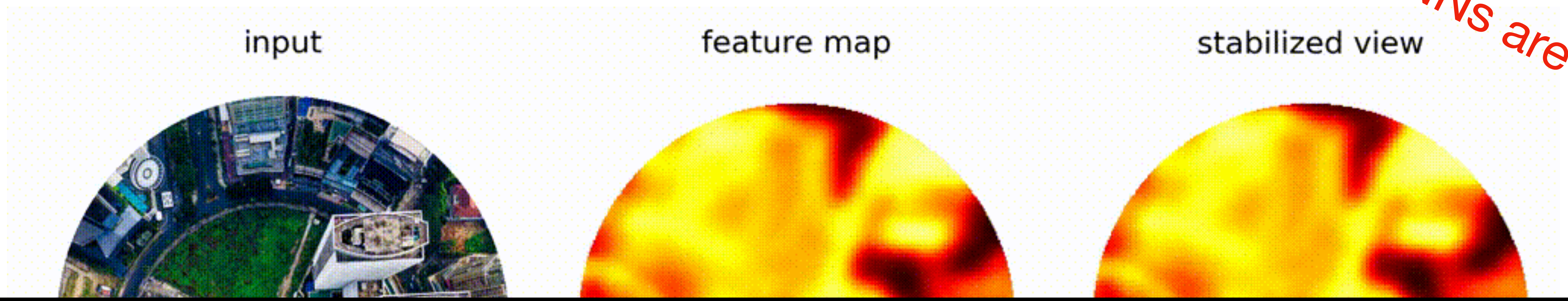
Equivariant NN



Figures source:
https://github.com/QUVA-Lab/e2cnn

Geometric guarantees (equivariance)

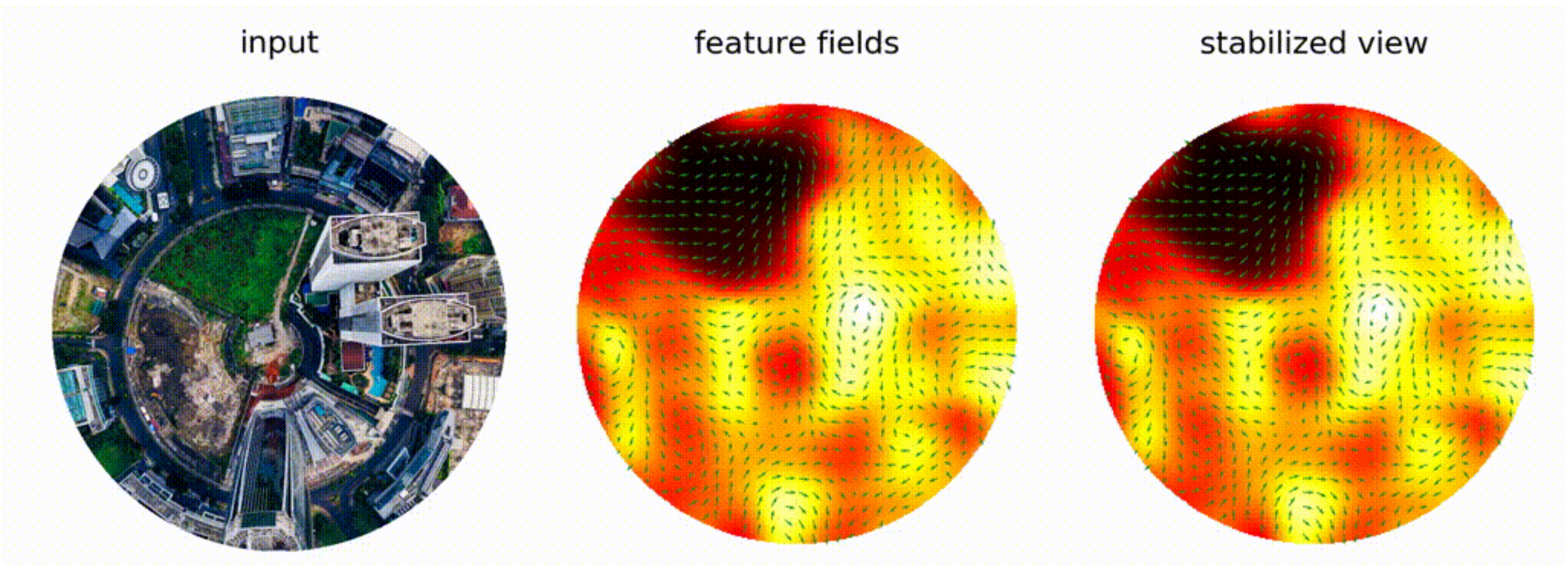
CNN



CNNs are not rotation equivariant

We need these tools to build NNs that generalize!

Equivariant NN



Figures source:
https://github.com/QUVA-Lab/e2cnn

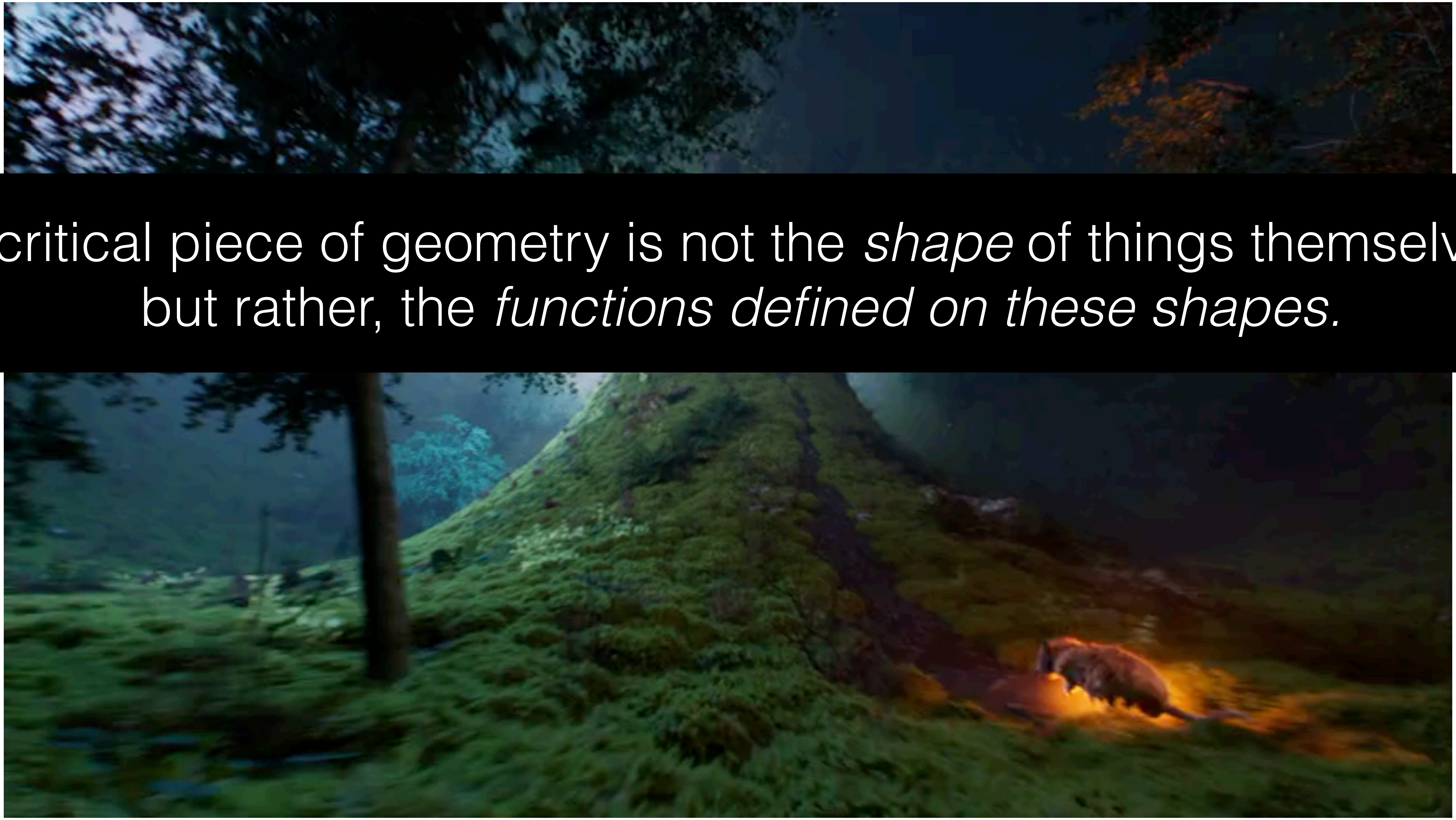
Geometry is an integral part of human vision

Credit: Helmet and the Norwegian University of Science and Technology's Kavli Institute for Systems Neuroscience

Geometry is an integral part of human vision

Credit: Helmet and the Norwegian University of Science and Technology's Kavli Institute for Systems Neuroscience

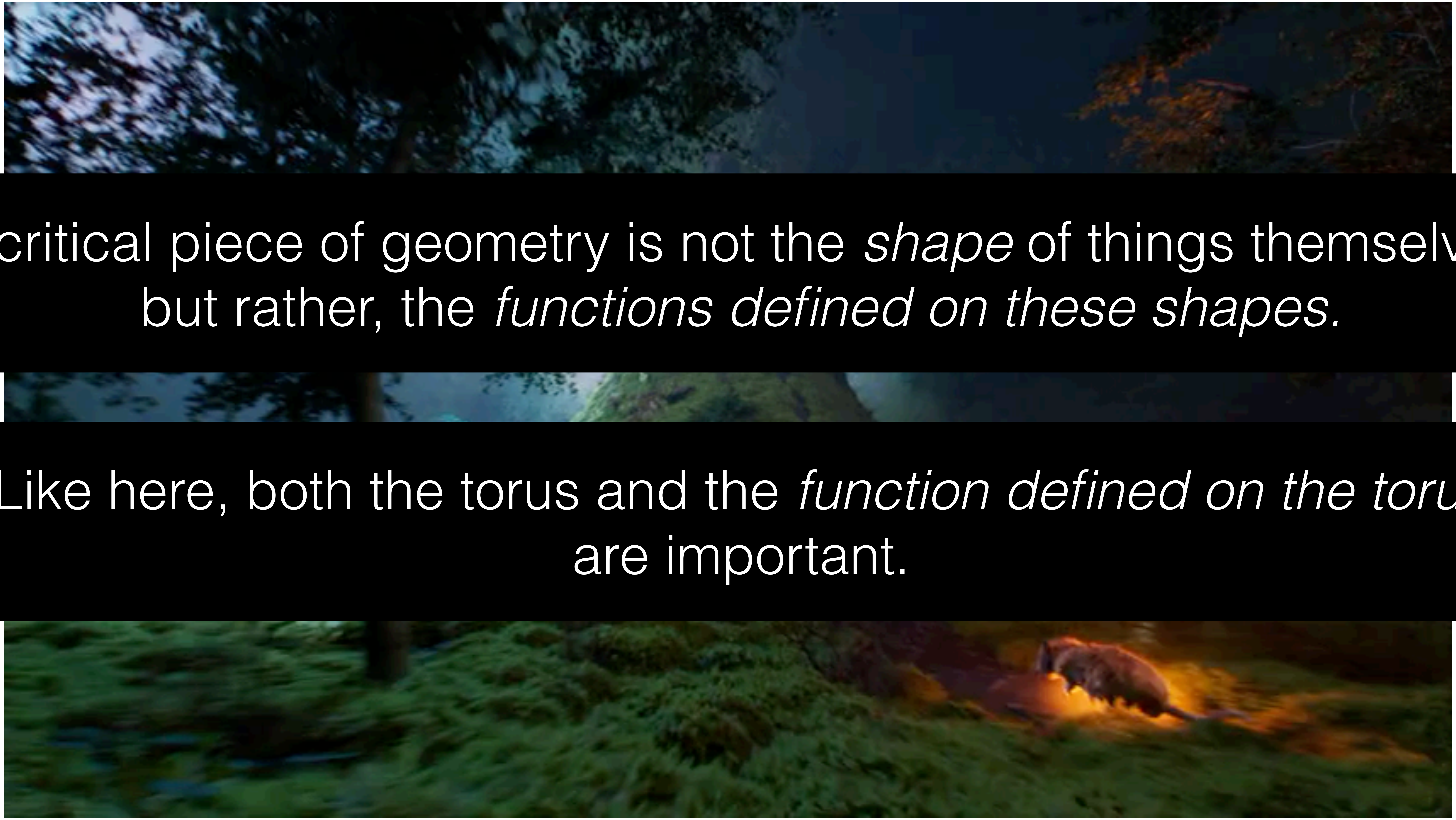
Geometry is an integral part of human vision



A critical piece of geometry is not the *shape* of things themselves, but rather, the *functions defined on these shapes.*

Credit: Helmet and the Norwegian University of Science and Technology's Kavli Institute for Systems Neuroscience

Geometry is an integral part of human vision



A critical piece of geometry is not the *shape* of things themselves, but rather, the *functions defined on these shapes.*

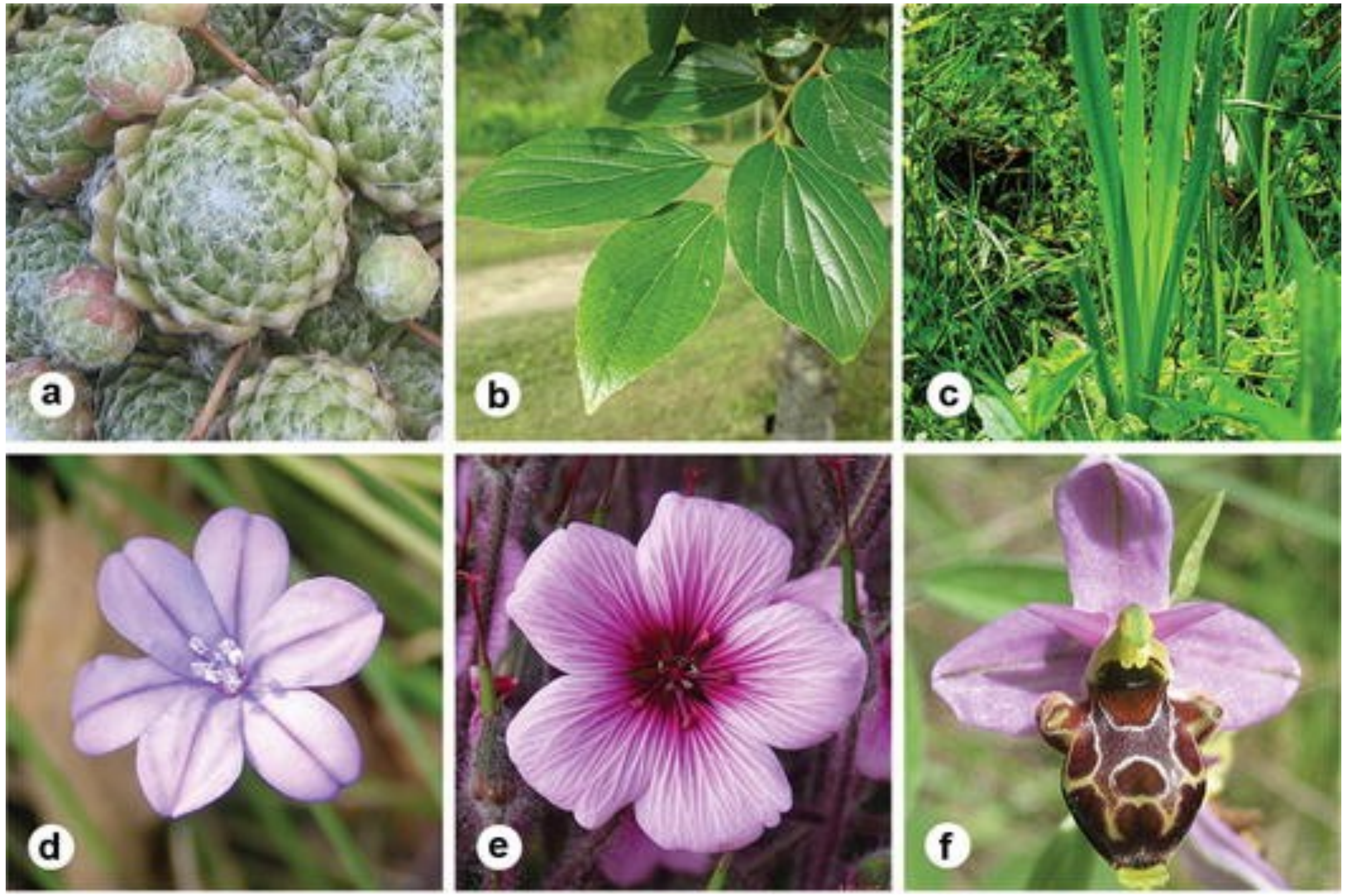
Like here, both the torus and the *function defined on the torus* are important.

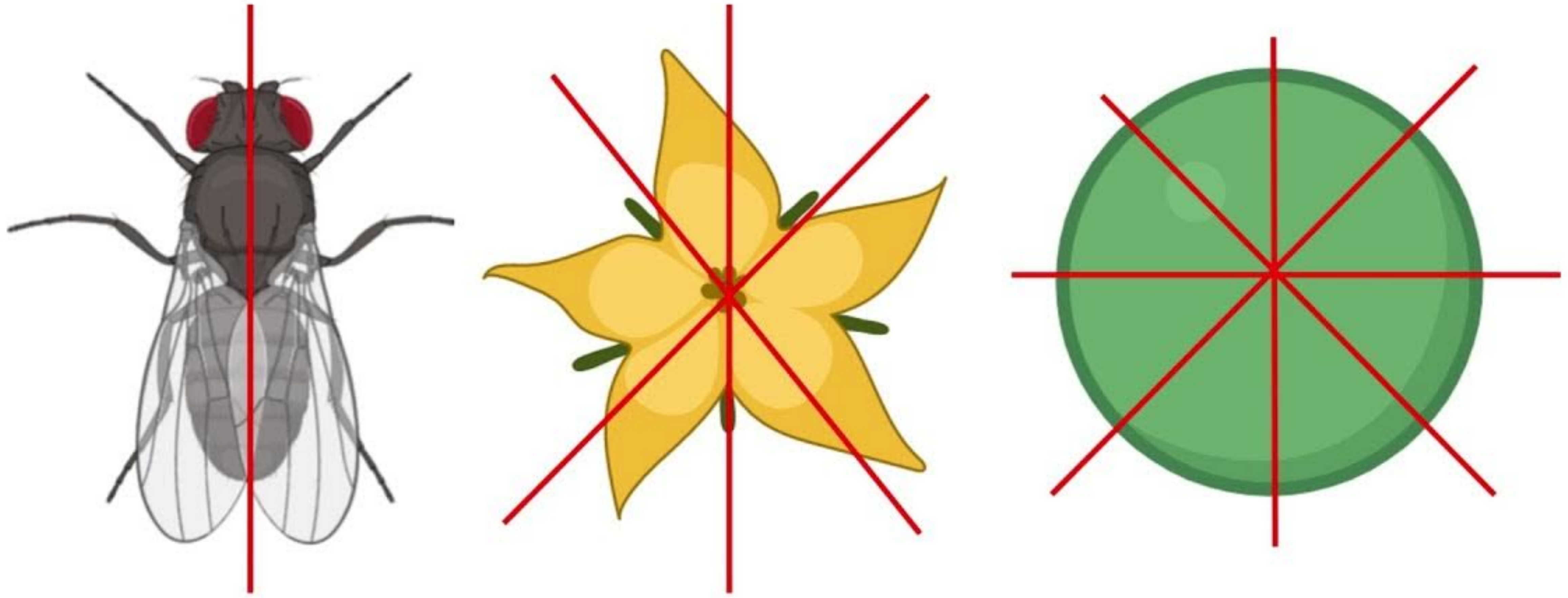
Credit: Helmet and the Norwegian University of Science and Technology's Kavli Institute for Systems Neuroscience

“Symmetry” in everyday use

We usually call things “symmetric” when they remain unchanged under rotations or mirroring:
rotation symmetry,
reflection symmetry

Lots of things in nature are symmetric!





“Symmetry” in mathematics and Physics

In mathematics, a symmetry of an object (such as a space, function, or equation) is a transformation that **preserves certain fundamental properties of the object**, such as distance, angle, or structure.

“Symmetry” in mathematics and Physics

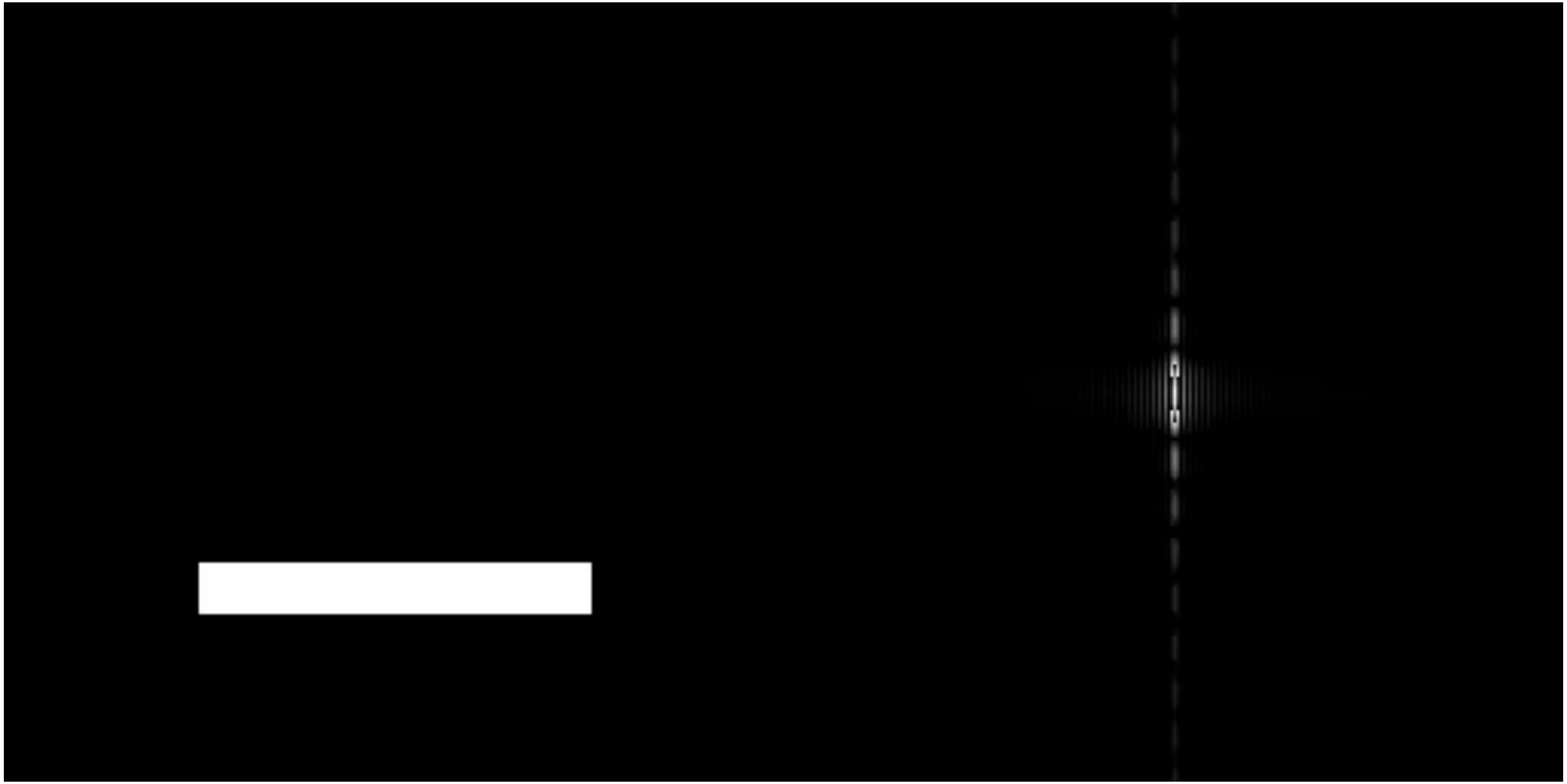
In mathematics, a symmetry of an object (such as a space, function, or equation) is a transformation that **preserves certain fundamental properties of the object**, such as distance, angle, or structure.

Have we encountered any symmetries so far?

Circular Shifts preserve the Fourier Magnitude

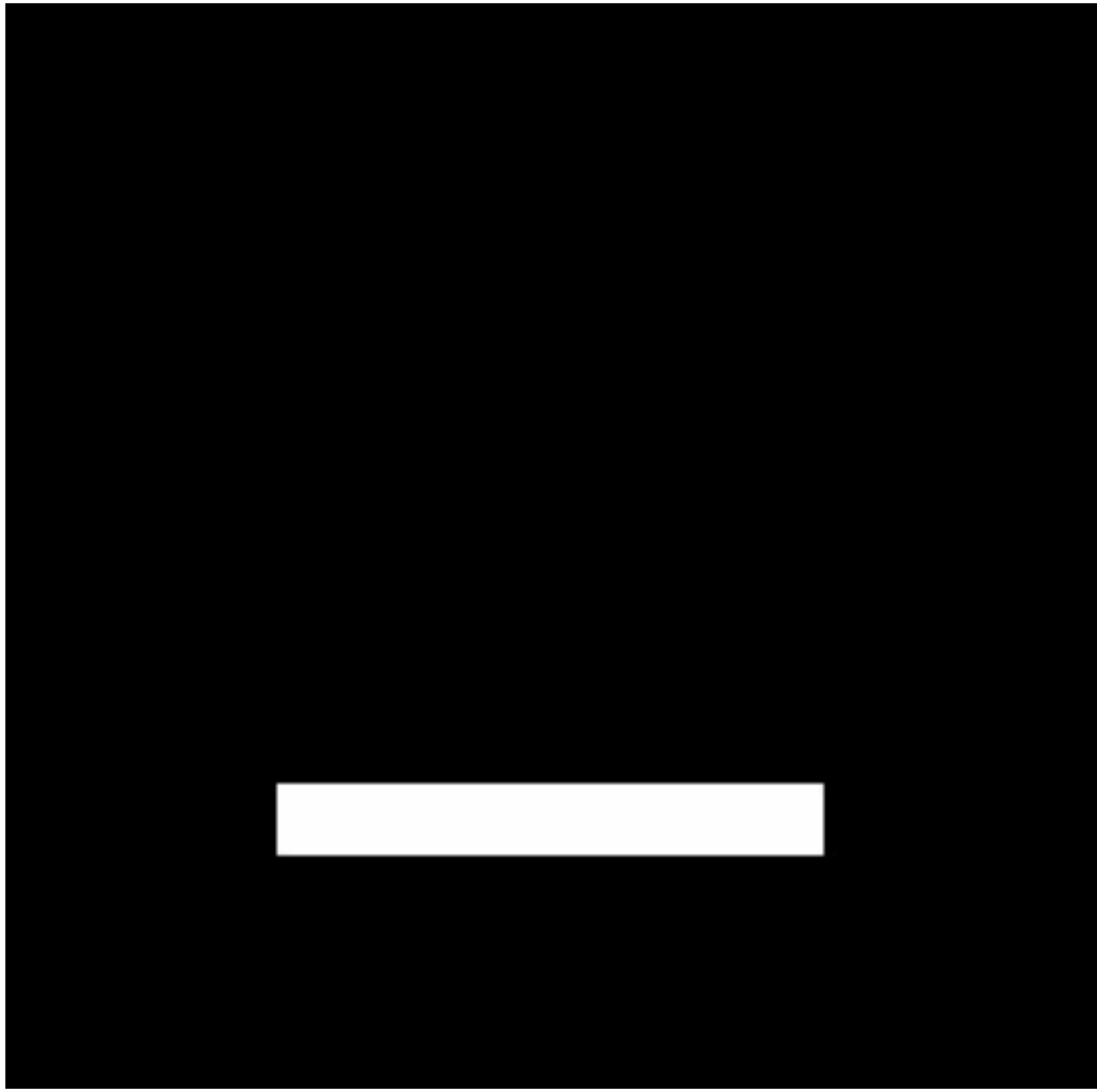
Image

Fourier Magnitudes

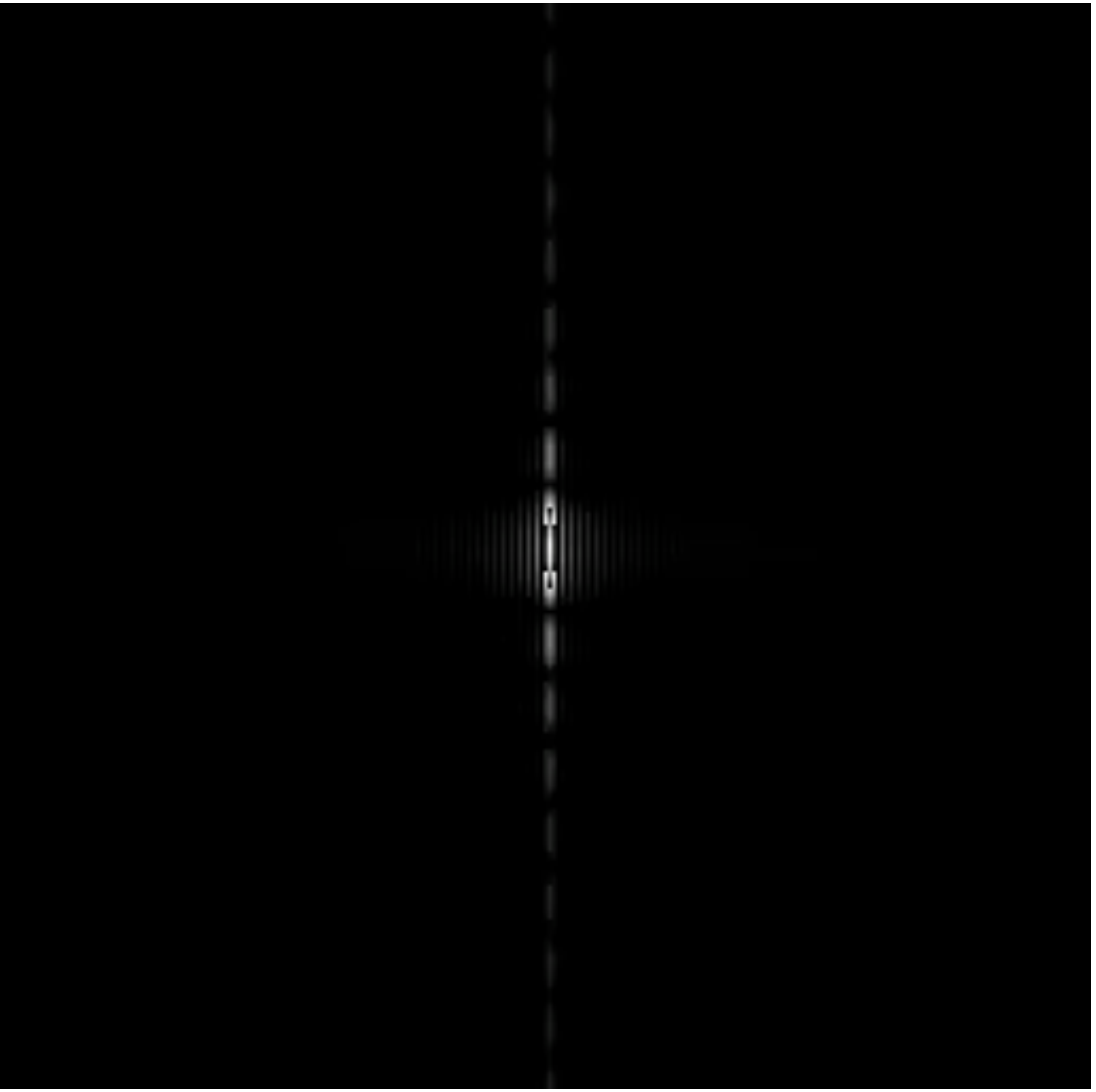


Circular Shifts preserve the Fourier Magnitude

Image



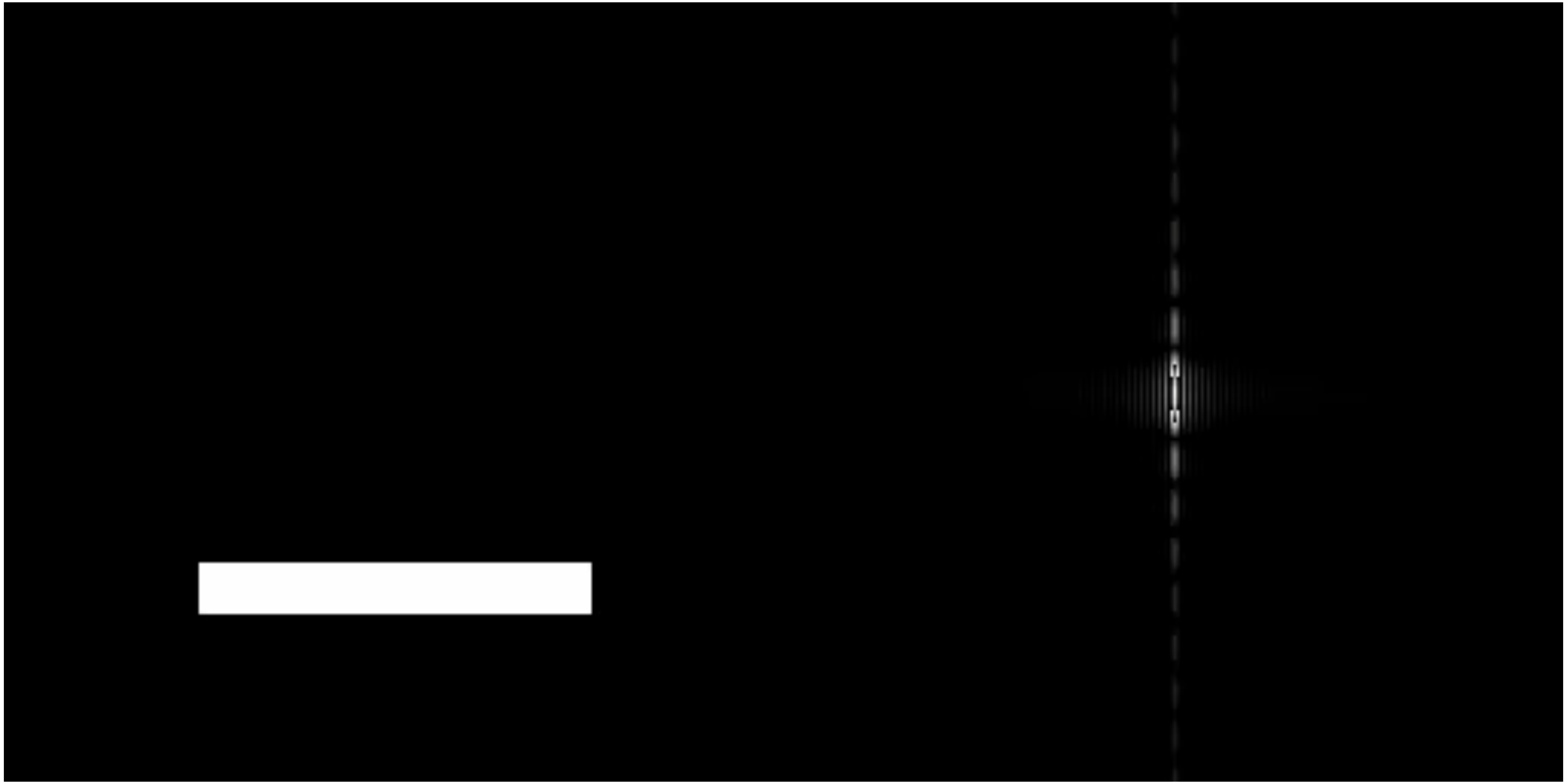
Fourier Magnitudes



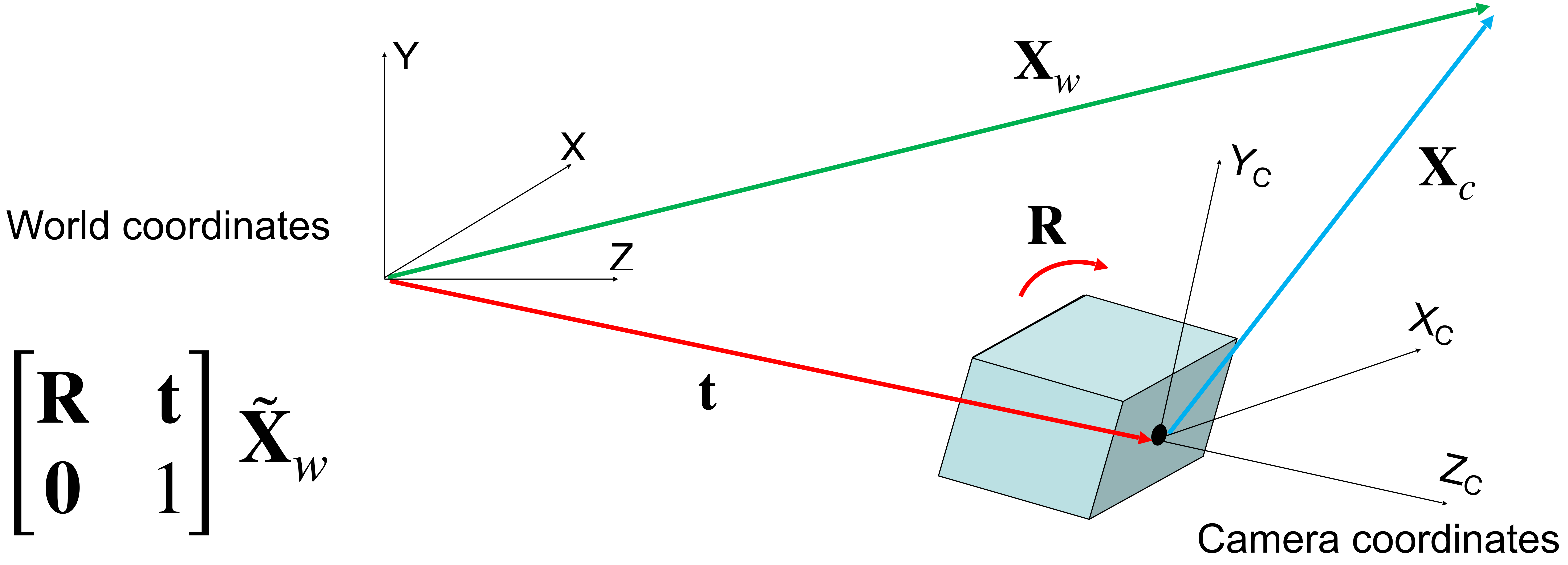
Circular Shifts preserve the Fourier Magnitude

Image

Fourier Magnitudes



SE(3) transformations preserve distances (vector norms)



$$\tilde{\mathbf{X}}_C = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tilde{\mathbf{X}}_w$$

Math Fundamentals: Basics of Group Theory

The abstract idea of a *group*.

A group $(G, \cdot\,)$ is a set of elements G equipped with a group operation $\cdot$, a function $G \times G \to G$, that satisfies the following four axioms:

- **Closure**: Given two elements g and h of G, $g \cdot h$ is also in G.
- **Associativity**: For $g, h, i \in G$ $\cdot$ is associative, i.e., $g \cdot (h \cdot i) = (g \cdot h) \cdot i$.
- **Identity element**: There exists an identity element $e \in G$ such that $e \cdot g = g \cdot e = g$ for any $g \in G$.
- **Inverse element**: For each $g \in G$ there exists an inverse element $g^{-1} \in G$ s.t. $g^{-1} \cdot g = g \cdot g^{-1} = e$.

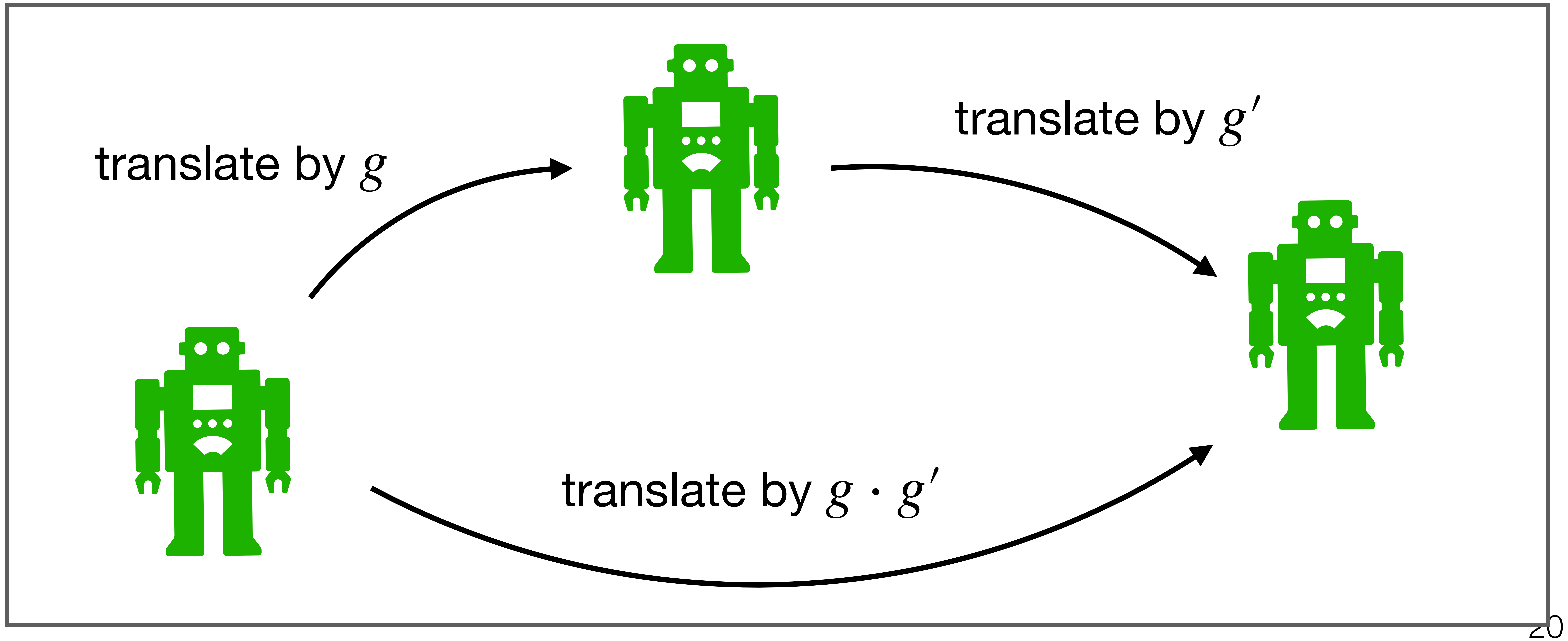
Slide credit: Erik Bekkers, University of Amsterdam

Translation group $(\mathbb{R}^2, +)$

The translation group consists of all possible translations in $\mathbb{R}^2$ and is equipped with the group operation of addition and group inverse:

$$g \cdot g' = (\mathbf{x} + \mathbf{x}')$$
$$g^{-1} = (-\mathbf{x})$$

with $g = (\mathbf{x}), g' = (\mathbf{x}')$ and $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^2$.



Slide credit: Erik Bekkers, University of Amsterdam

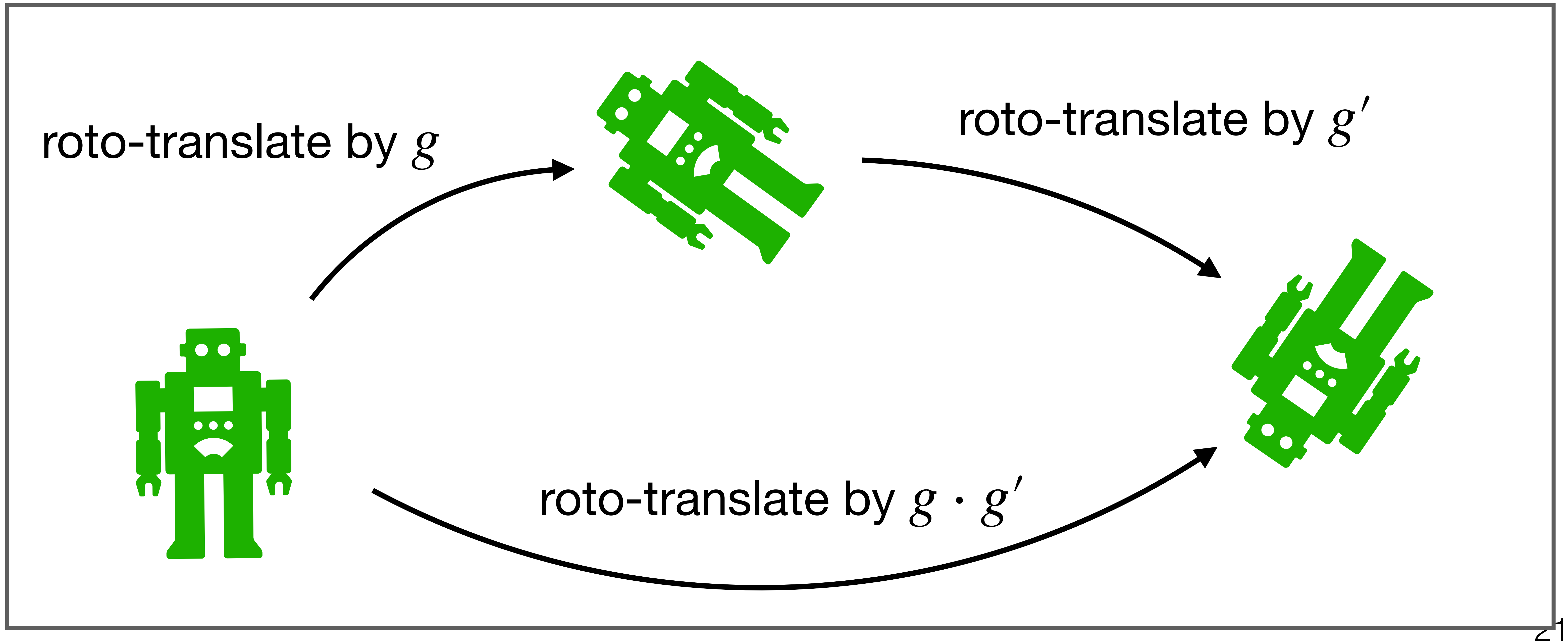
Roto-ranslation group $SE(2)$

***2**D **S**pecial **E**uclidean motion group*

The group $SE(2) = \mathbb{R}^2 \rtimes SO(2)$ consists of the coupled space $\mathbb{R}^2 \times S^1$ of translations vectors in $\mathbb{R}^2$, and rotations in $SO(2)$ (or equivalently orientations in S^1), and is equipped with the group product and group inverse:

$$g \cdot g' = (\mathbf{x}, \mathbf{R}_\theta) \cdot (\mathbf{x}', \mathbf{R}_{\theta'}) = (\mathbf{R}_\theta \mathbf{x}' + \mathbf{x}, \mathbf{R}_{\theta+\theta'})$$
$$g^{-1} = (-\mathbf{R}_\theta^{-1}\mathbf{x}, \mathbf{R}_\theta^{-1})$$

with $g = (\mathbf{x}, \mathbf{R}_\theta)$, $g' = (\mathbf{x}', \mathbf{R}_{\theta'})$.



Representations

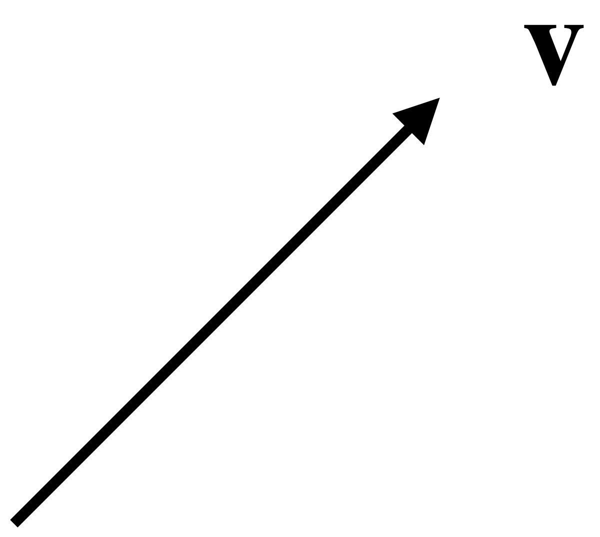
A representation $\rho : G \rightarrow GL(V)$ is a function from G to the general linear group (all invertible matrices) $GL(V)$.

That is $\rho(g)$ is a linear transformation that is parameterized by group elements $g \in G$ that transforms some vector $\mathbf{v} \in V$ **(e.g. an image)** such that

$$\rho(g') \circ \rho(g)[\mathbf{v}] = \rho(g' \cdot g)\,[\mathbf{v}]$$

Slide credit: Erik Bekkers, University of Amsterdam

Representations



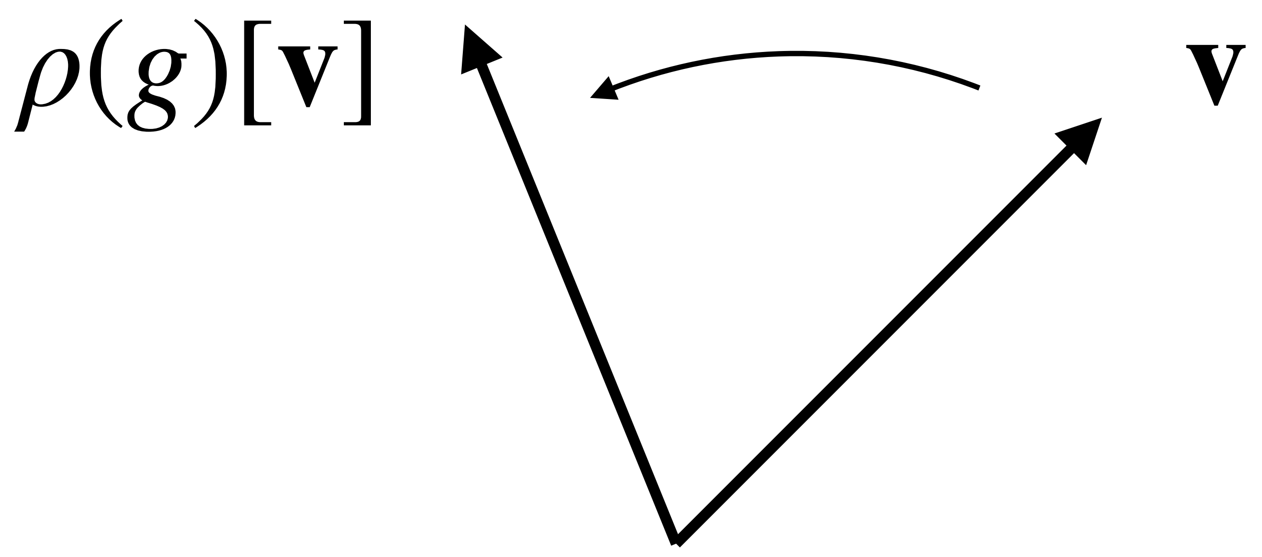
A representation $\rho : G \to GL(V)$ is a function from G to the general linear group (all invertible matrices) $GL(V)$.

That is $\rho(g)$ is a linear transformation that is parameterized by group elements $g \in G$ that transforms some vector $\mathbf{v} \in V$ **(e.g. an image)** such that

$$\rho(g') \circ \rho(g)[\mathbf{v}] = \rho(g' \cdot g)\,[\mathbf{v}]$$

Slide credit: Erik Bekkers, University of Amsterdam

Representations



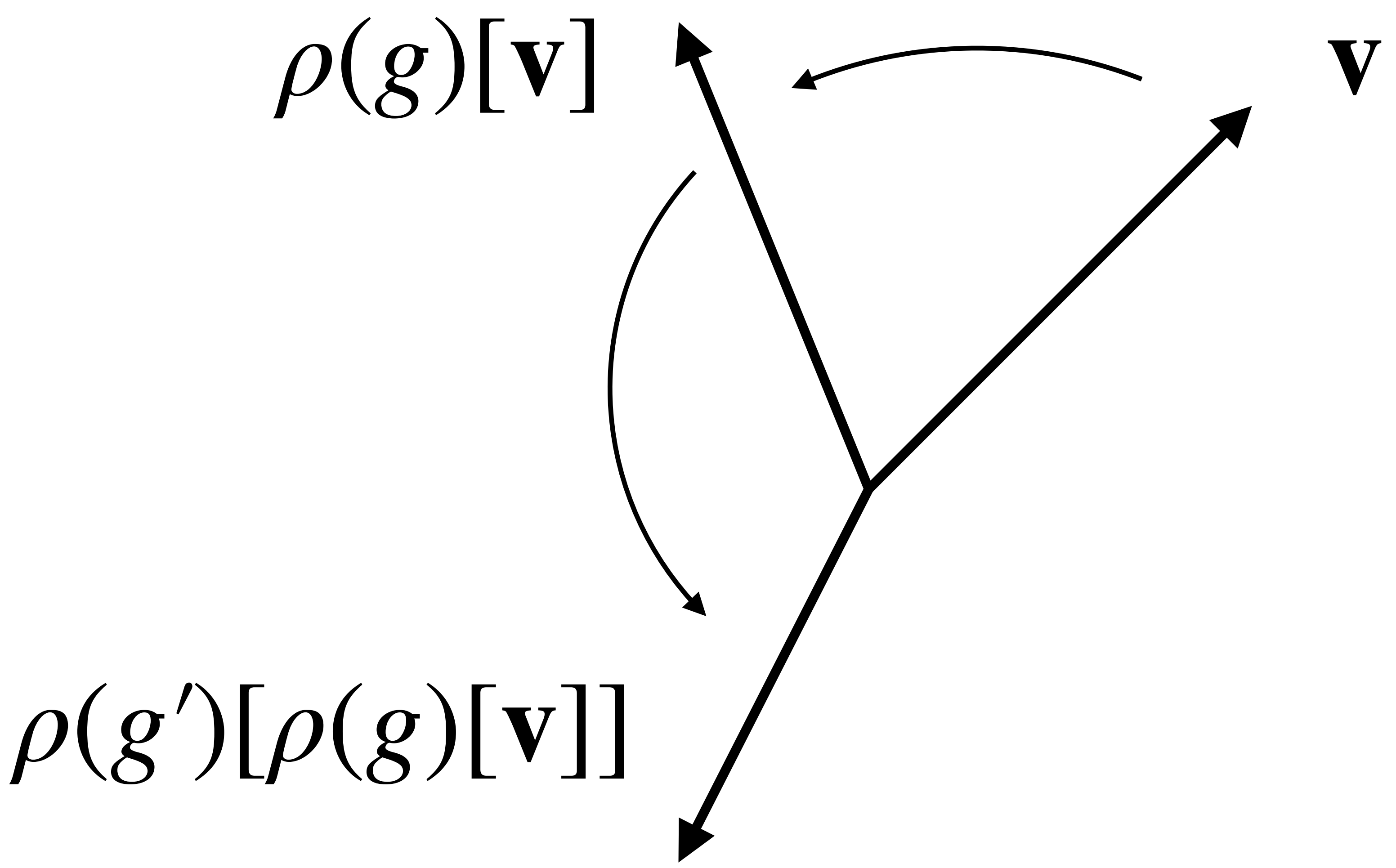
A representation $\rho : G \to GL(V)$ is a function from G to the general linear group (all invertible matrices) $GL(V)$.

That is $\rho(g)$ is a linear transformation that is parameterized by group elements $g \in G$ that transforms some vector $\mathbf{v} \in V$ **(e.g. an image)** such that

$$\rho(g') \circ \rho(g)[\mathbf{v}] = \rho(g' \cdot g)\,[\mathbf{v}]$$

Slide credit: Erik Bekkers, University of Amsterdam

Representations



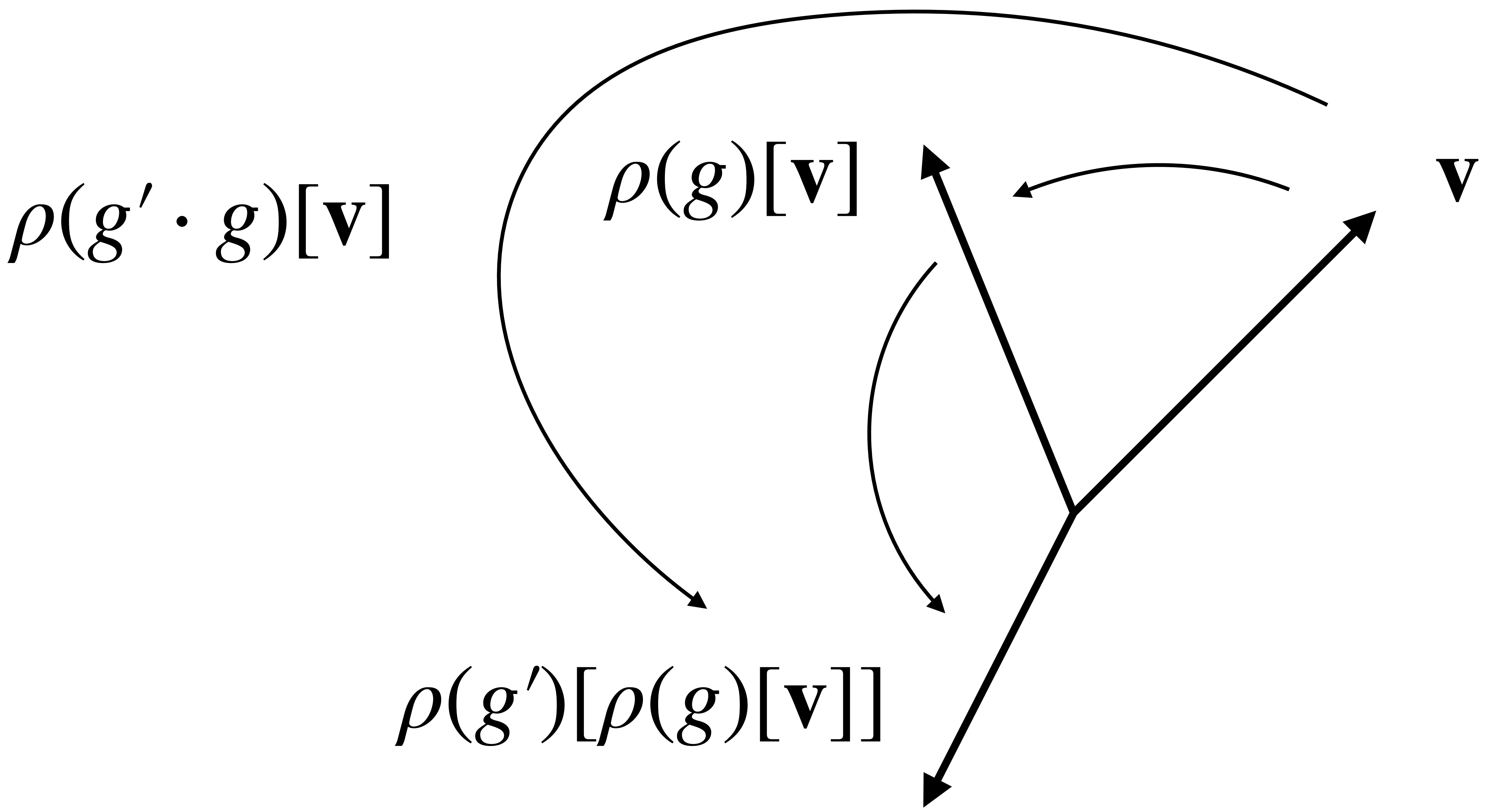
A representation $\rho : G \rightarrow GL(V)$ is a function from G to the general linear group (all invertible matrices) $GL(V)$.

That is $\rho(g)$ is a linear transformation that is parameterized by group elements $g \in G$ that transforms some vector $\mathbf{v} \in V$ **(e.g. an image)** such that

$$\rho(g') \circ \rho(g)[\mathbf{v}] = \rho(g' \cdot g)\,[\mathbf{v}]$$

Slide credit: Erik Bekkers, University of Amsterdam

Representations



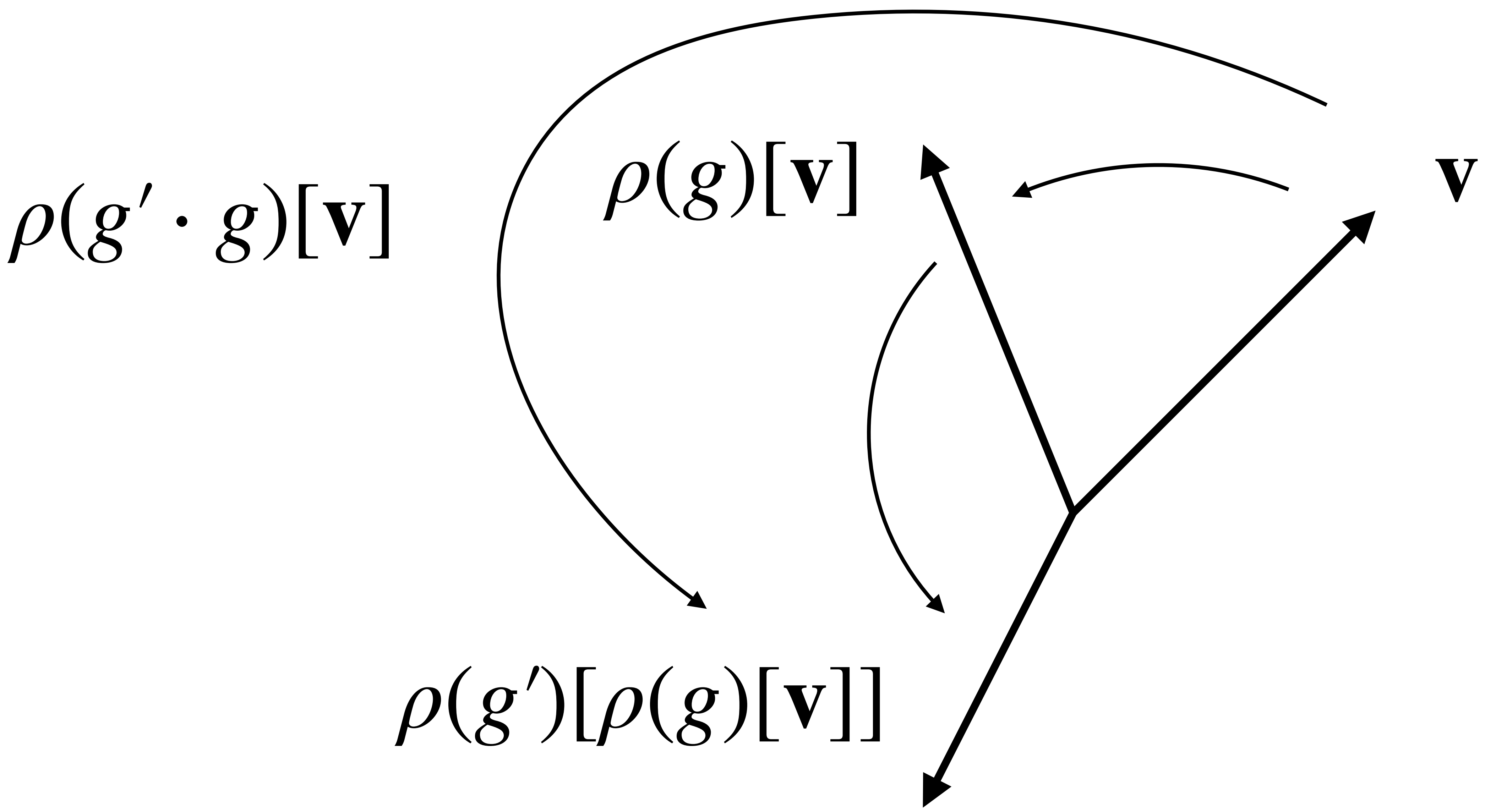
A representation $\rho : G \to GL(V)$ is a function from G to the general linear group (all invertible matrices) $GL(V)$.

That is $\rho(g)$ is a linear transformation that is parameterized by group elements $g \in G$ that transforms some vector $\mathbf{v} \in V$ **(e.g. an image)** such that

$$\rho(g') \circ \rho(g)[\mathbf{v}] = \rho(g' \cdot g)\,[\mathbf{v}]$$

Slide credit: Erik Bekkers, University of Amsterdam

Representations



$$\rho(g^{-1}) = [\rho(g)]^{-1}$$

A representation $\rho : G \to GL(V)$ is a function from G to the general linear group (all invertible matrices) $GL(V)$.

That is $\rho(g)$ is a linear transformation that is parameterized by group elements $g \in G$ that transforms some vector $\mathbf{v} \in V$ **(e.g. an image)** such that

$$\rho(g') \circ \rho(g)[\mathbf{v}] = \rho(g' \cdot g)\,[\mathbf{v}]$$

Slide credit: Erik Bekkers, University of Amsterdam

Left-regular Representations

A left-regular representation $\mathscr{L}_g$ is a representation that transforms functions f by transforming their domains via the inverse group action

$$\mathscr{L}_g[f](x) := f(g^{-1} \cdot x)$$

"group action" equals group product when domain is G

Slide credit: Erik Bekkers, University of Amsterdam

Left-regular Representations

Example:

$f \in \mathbb{L}_2(\mathbb{R}^2)$

- a 2D image

$G = SE(2)$

- the roto-translation group

$\mathscr{L}_g(f)(\mathbf{y}) = f(\mathbf{R}_\theta^{-1}(\mathbf{y} - \mathbf{x}))$

- a roto-translation of the image

A left-regular representation $\mathscr{L}_g$ is a representation that transforms functions f by transforming their domains via the inverse group action

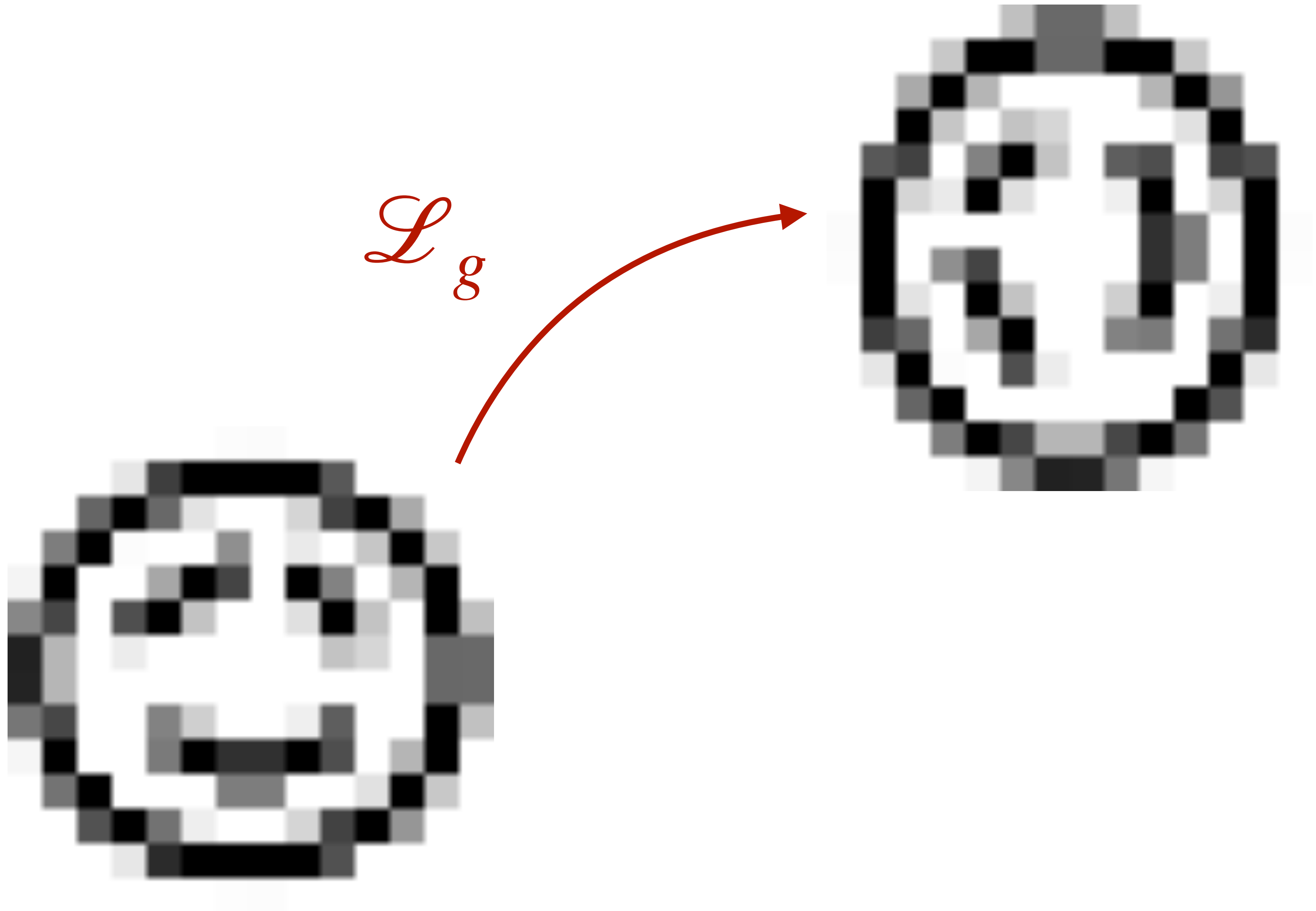
$$\mathscr{L}_g[f](x) := f(g^{-1} \cdot x)$$

"group action" equals group product when domain is G

Slide credit: Erik Bekkers, University of Amsterdam

Left-regular Representations

Example:



$f \in \mathbb{L}_2(\mathbb{R}^2)$

- a 2D image

$G = SE(2)$

- the roto-translation group

$\mathscr{L}_g(f)(\mathbf{y}) = f(\mathbf{R}_\theta^{-1}(\mathbf{y} - \mathbf{x}))$

- a roto-translation of the image

A left-regular representation $\mathscr{L}_g$ is a representation that transforms functions f by transforming their domains via the inverse group action

$$\mathscr{L}_g[f](x) := f(g^{-1} \cdot x)$$

"group action" equals group product when domain is G

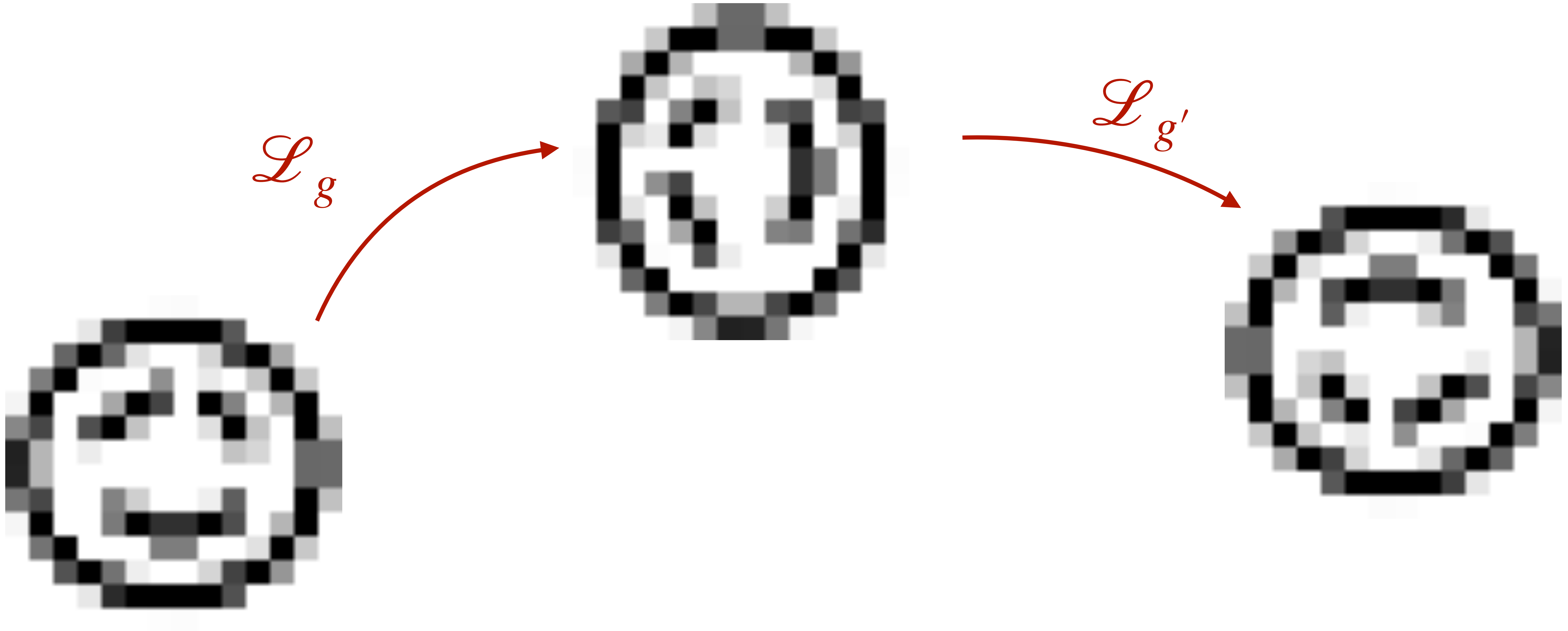
Left-regular Representations

Example:

$f \in \mathbb{L}_2(\mathbb{R}^2)$
- a 2D image

$G = SE(2)$
- the roto-translation group

$\mathscr{L}_g(f)(\mathbf{y}) = f(\mathbf{R}_\theta^{-1}(\mathbf{y} - \mathbf{x}))$
- a roto-translation of the image



A left-regular representation $\mathscr{L}_g$ is a representation that transforms functions f by transforming their domains via the inverse group action

$$\mathscr{L}_g[f](x) := f(g^{-1} \cdot x)$$

"group action" equals group product when domain is G

Slide credit: Erik Bekkers, University of Amsterdam

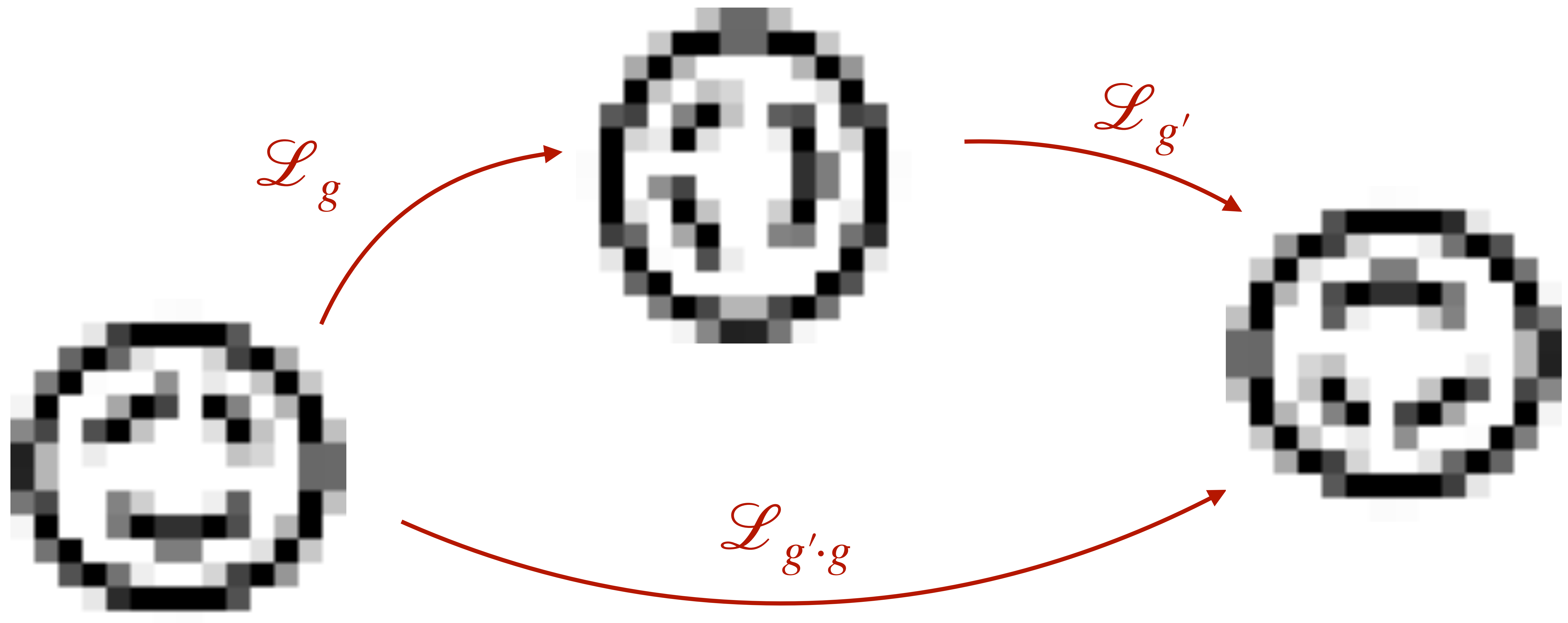
Left-regular Representations

Example:

$f \in \mathbb{L}_2(\mathbb{R}^2)$
- a 2D image

$G = SE(2)$
- the roto-translation group

$\mathscr{L}_g(f)(\mathbf{y}) = f(\mathbf{R}_\theta^{-1}(\mathbf{y} - \mathbf{x}))$
- a roto-translation of the image



A left-regular representation $\mathscr{L}_g$ is a representation that transforms functions f by transforming their domains via the inverse group action

$$\mathscr{L}_g[f](x) := f(g^{-1} \cdot x)$$

"group action" equals group product when domain is G

Slide credit: Erik Bekkers, University of Amsterdam

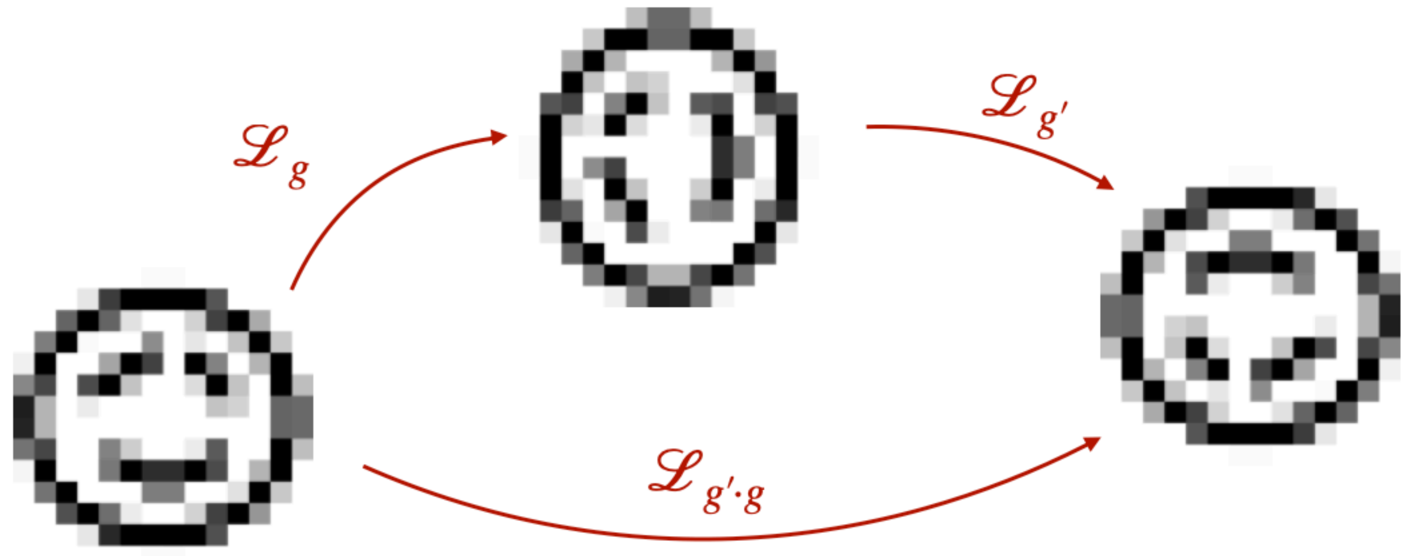
Group actions

Slide credit: Erik Bekkers, University of Amsterdam

Group actions

Left regular representation (the action on $\mathbb{L}_2(X)$)

$$\mathscr{L}_g f$$

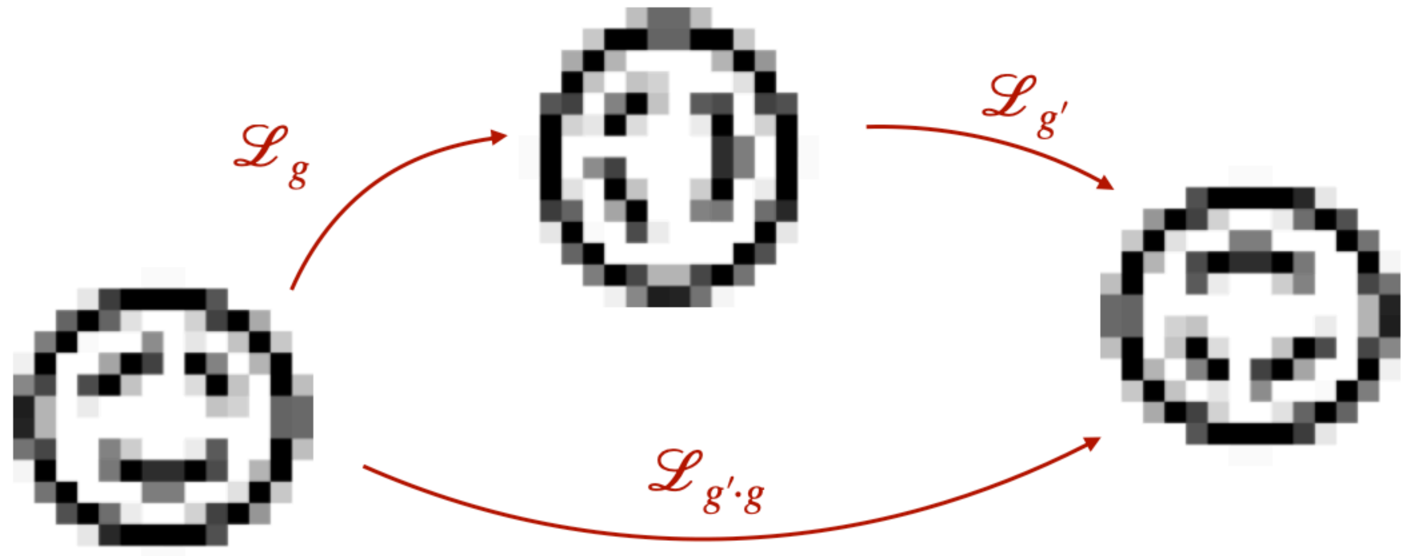


Slide credit: Erik Bekkers, University of Amsterdam

Group actions

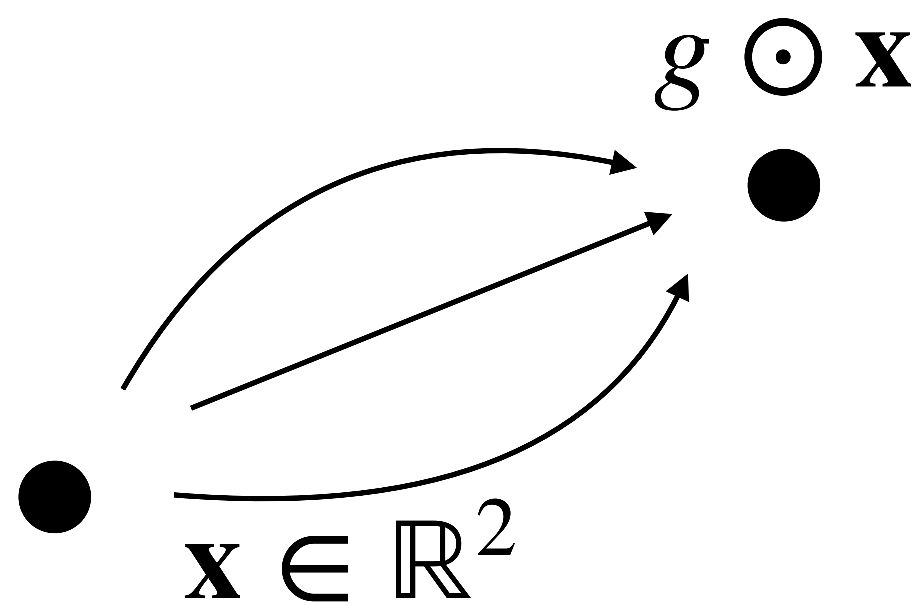
Left regular representation (the action on $\mathbb{L}_2(X)$)

$\mathscr{L}_g f$ $\qquad\qquad$ gf



Group action (the action on $\mathbb{R}^d$)

$g \odot \mathbf{x}$ $\qquad\qquad$ $g\mathbf{x}$

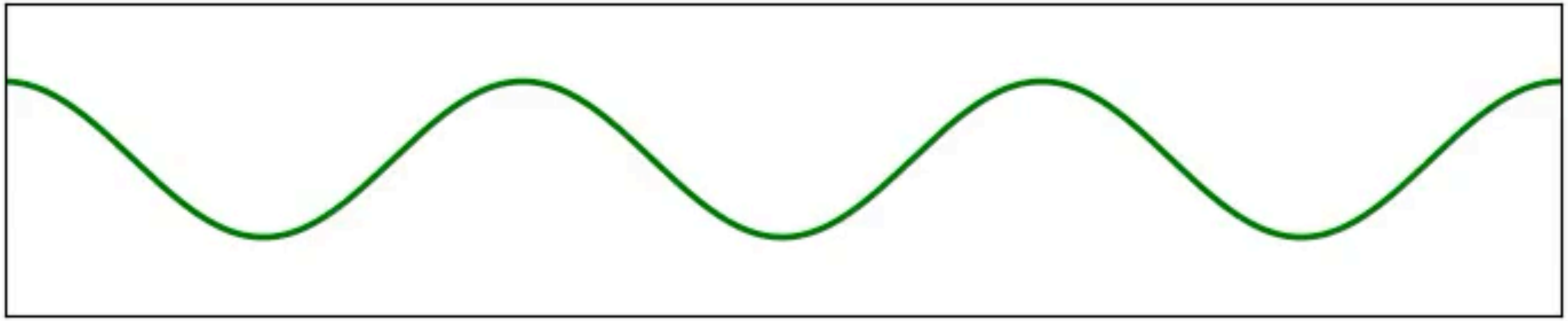


Slide credit: Erik Bekkers, University of Amsterdam

Shifting a sine wave

This is how we might think
about translation in
"coordinate world"

$$\sin(x - \theta) = sin(g^{-1}x)$$

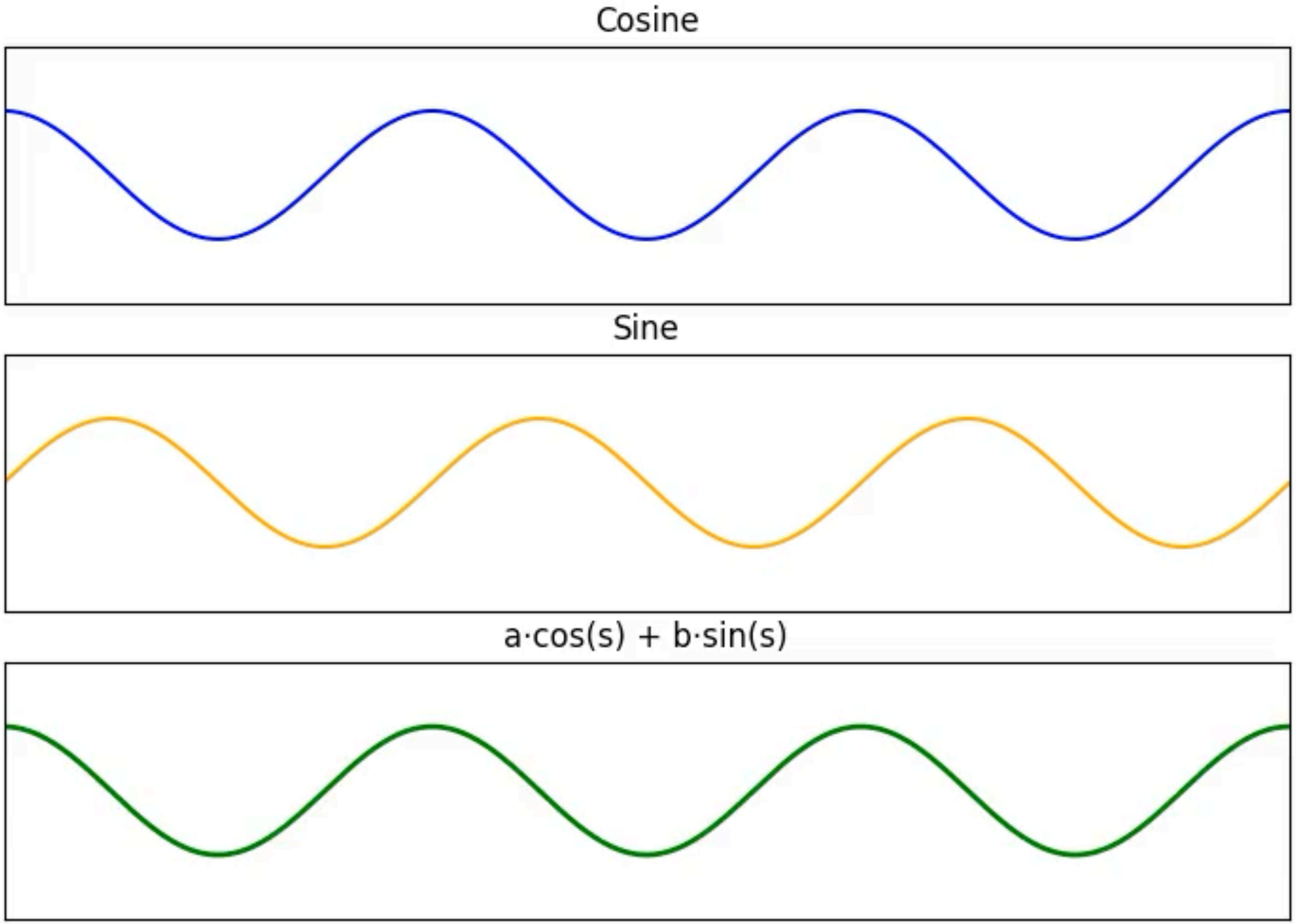


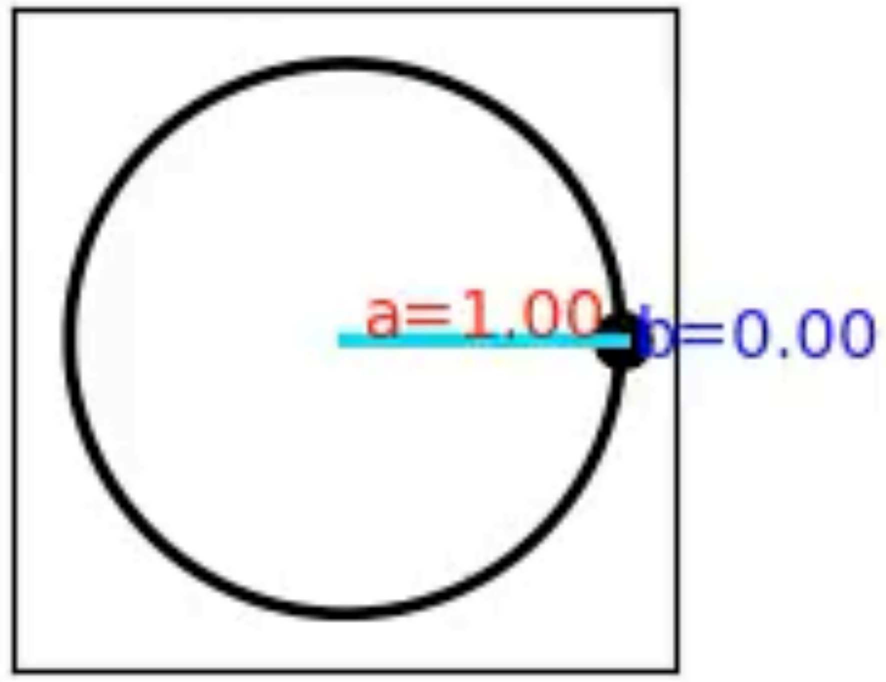
Translating a sine wave in the Fourier Basis

Express in (cos, sin) basis

Translation group acts via *rotation* of coefficients!

$$< \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \begin{bmatrix} a \\ b \end{bmatrix}, >$$



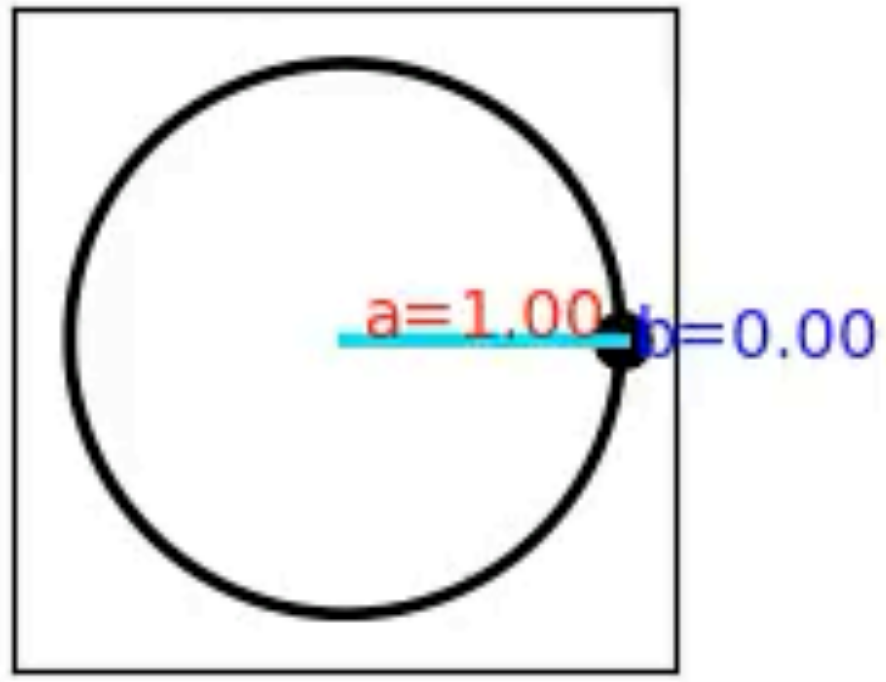


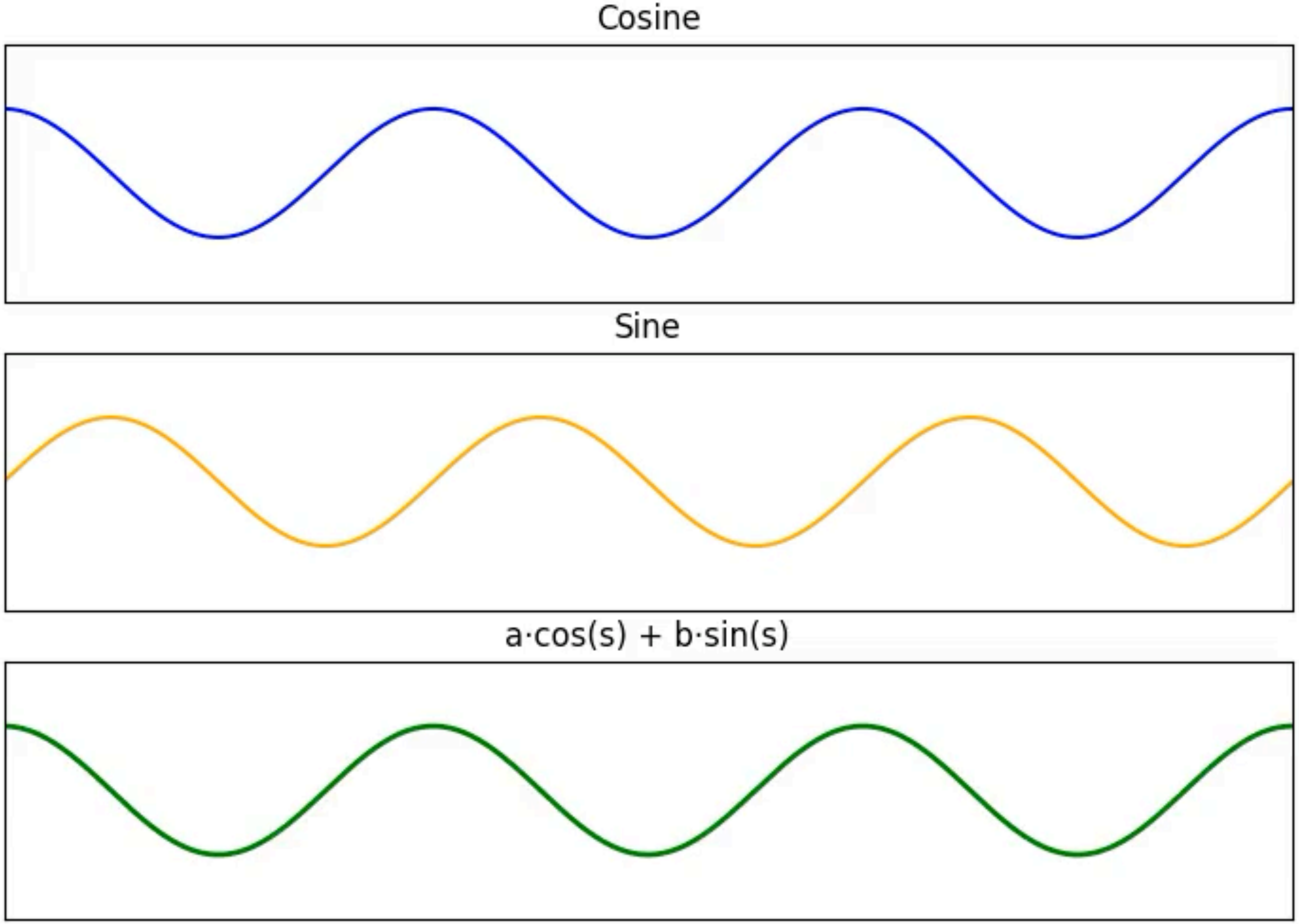
Translating a sine wave in the Fourier Basis

Express in (cos, sin) basis

Translation group acts via *rotation* of coefficients!

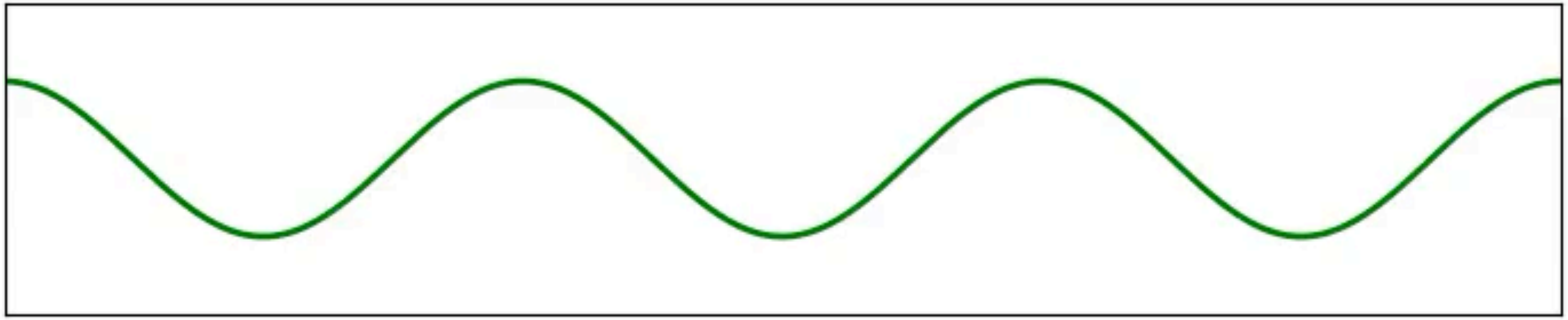
$$< \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \begin{bmatrix} a \\ b \end{bmatrix}, >$$





Circular shifts “act on” functions by rotating their fourier coefficients

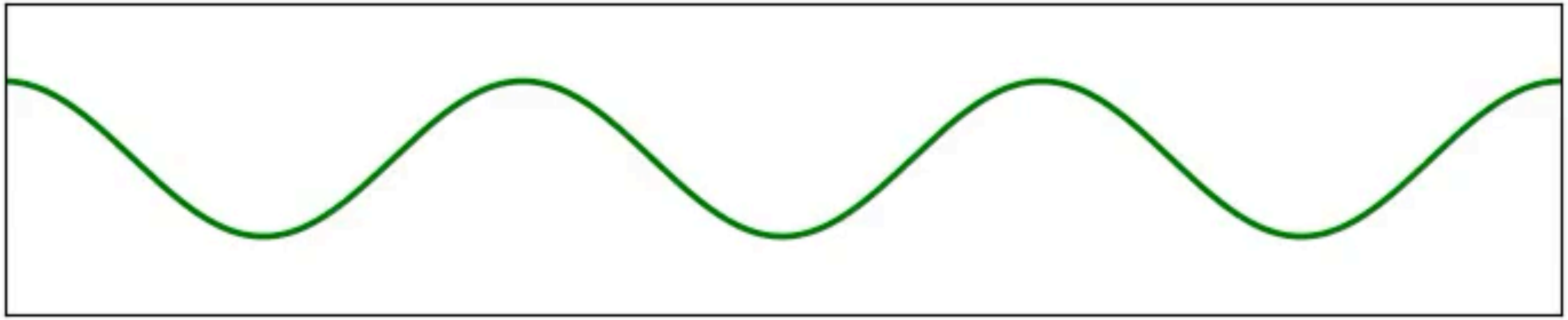
$$< \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}, \begin{bmatrix} cos(x) \\ sin(x) \end{bmatrix} > = sin(x - \theta)$$



The 2D rotation matrices are the left-regular group representations of the cyclical shift group

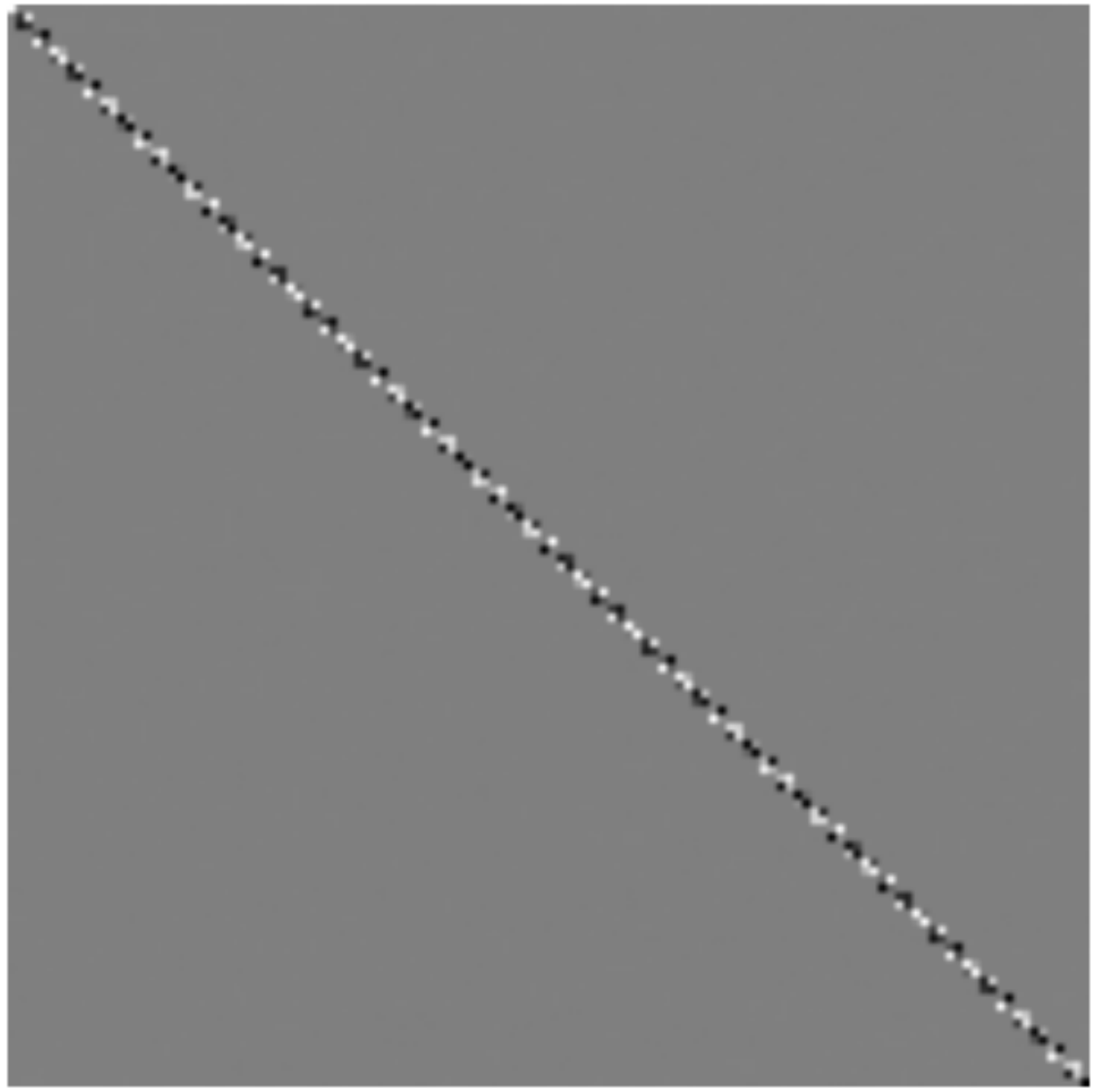
$$< \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}, \begin{bmatrix} cos(x) \\ sin(x) \end{bmatrix} > = sin(x - \theta)$$

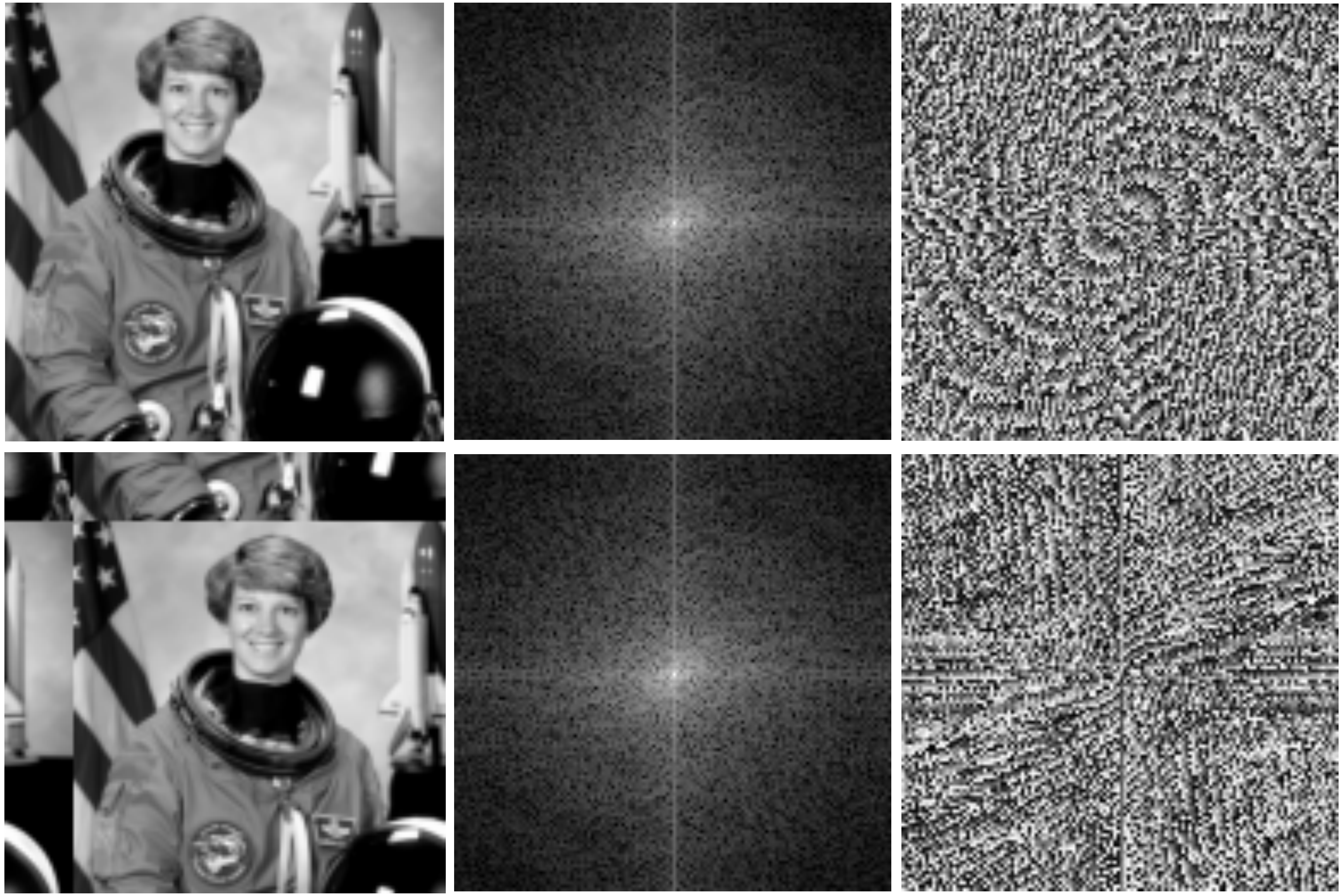
$$\mathscr{L}_g(f)(x) = f(g^{-1}x)$$



Shifting images

$$\mathscr{L}_g[f(x)] = f(g^{-1}x)$$





Steerable basis

A vector $Y(x) = \begin{pmatrix} \vdots \\ Y_l(x) \\ \vdots \end{pmatrix} \in \mathbb{K}^L$ with (basis) functions $Y_l \in \mathbb{L}_2(X)$ is steerable if

$$\forall_{g \in G}: \qquad Y(g\,x) = \rho(g) Y(x),$$

where $g\,x$ denotes the action of G on X and $\rho(g) \in \mathbb{K}^{L \times L}$ is a representation of G.

I.e., we can transform all basis functions simply by taking a linear combination of the original basis functions.

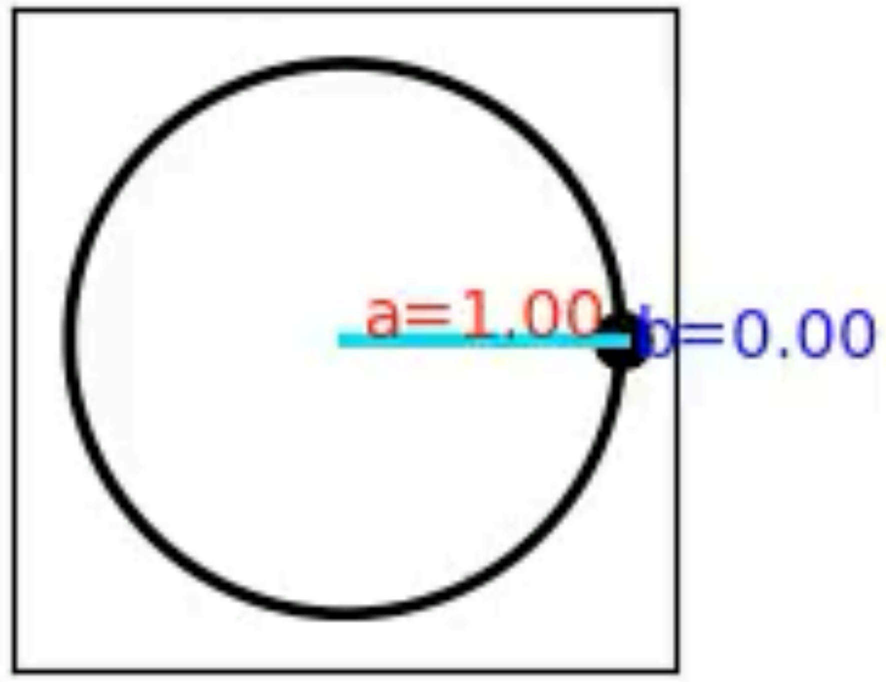
Slide credit: Erik Bekkers, University of Amsterdam

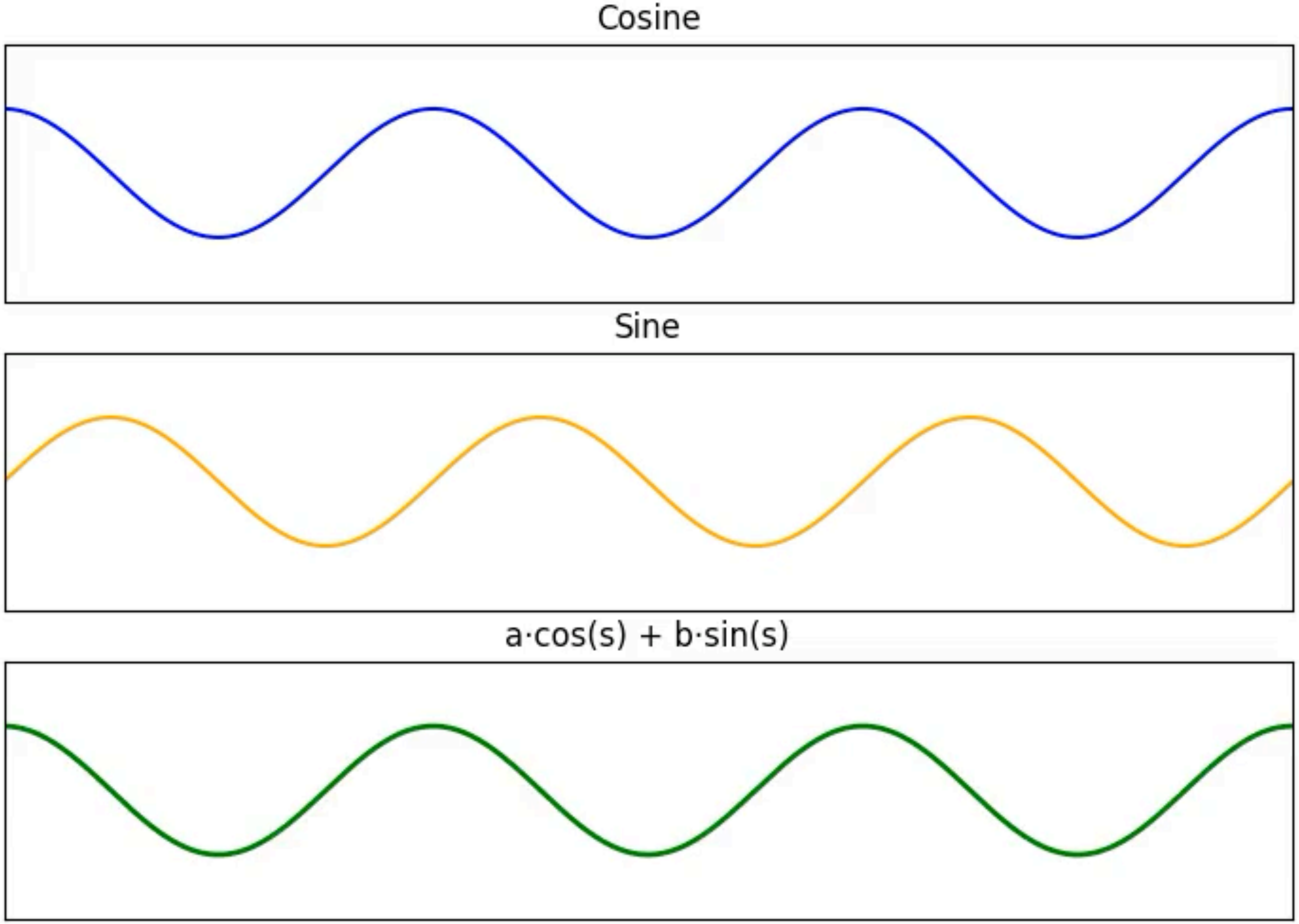
Translating a sine wave in the Fourier Basis

Express in
(cos, sin) basis

Translation group acts
via *rotation* of
coefficients!

$$< \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \begin{bmatrix} a \\ b \end{bmatrix} >$$



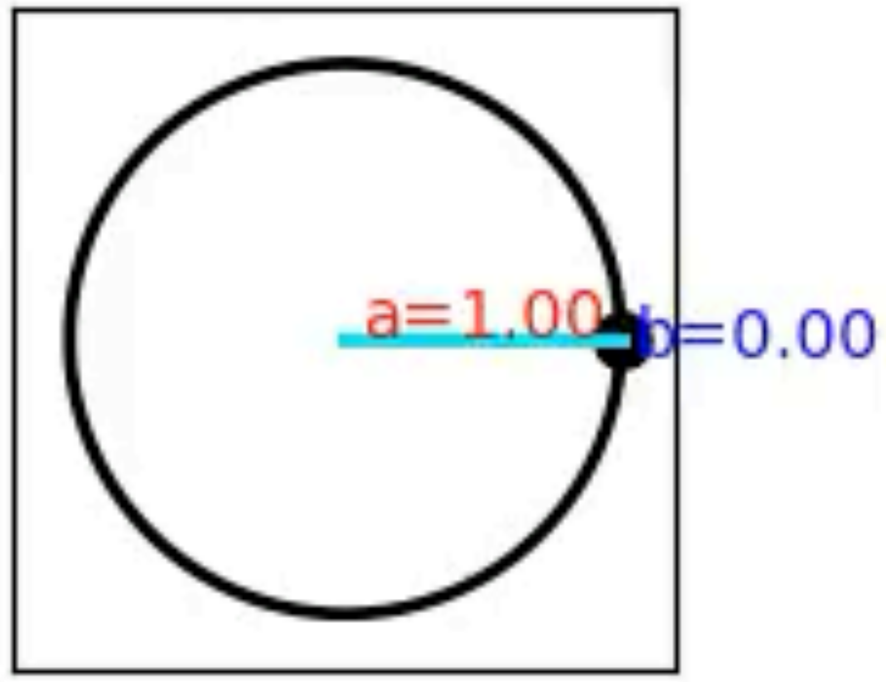


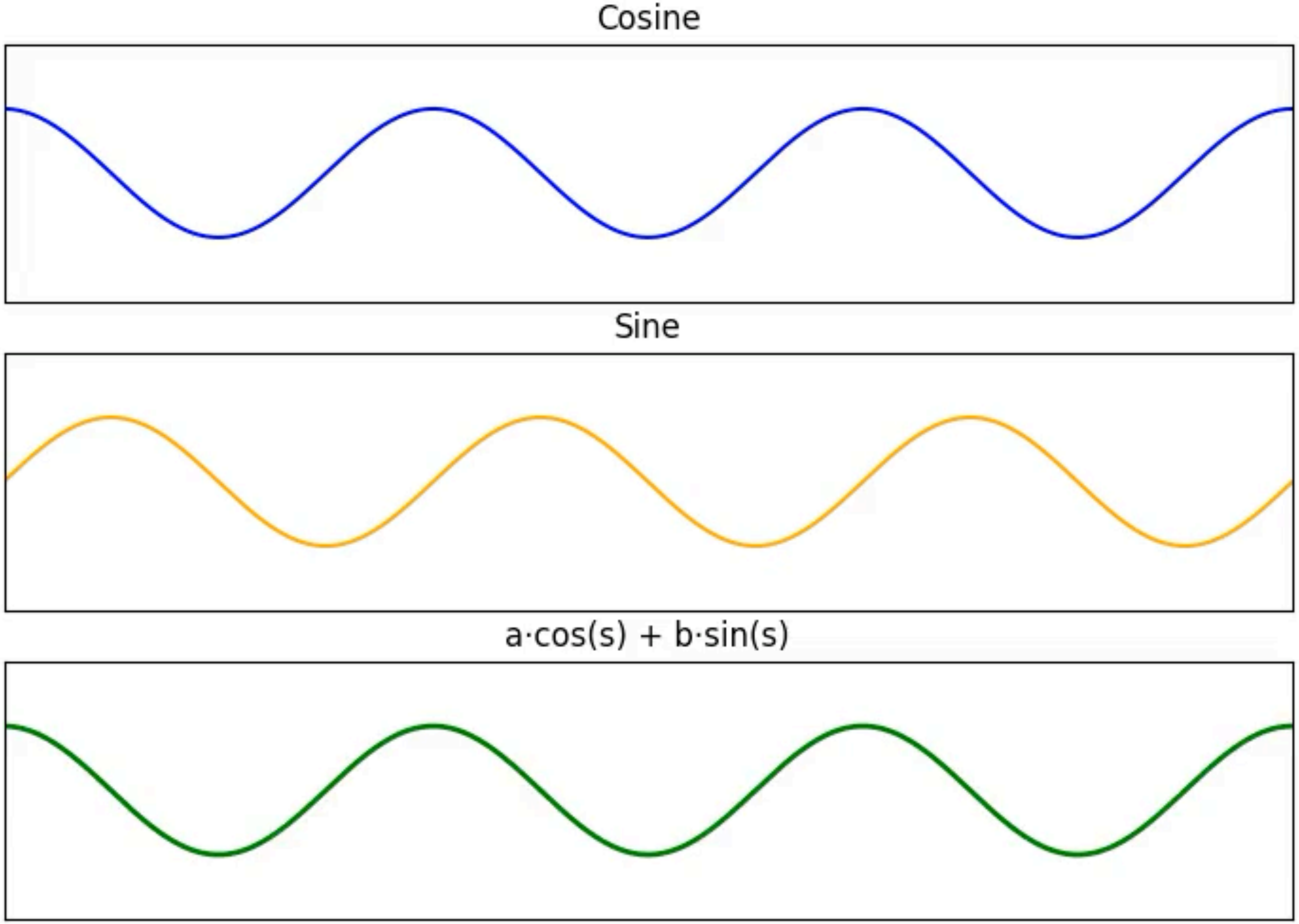
Translating a sine wave in the Fourier Basis

Express in
(cos, sin) basis

Translation group acts
via *rotation* of
coefficients!

$$< \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \begin{bmatrix} a \\ b \end{bmatrix} >$$

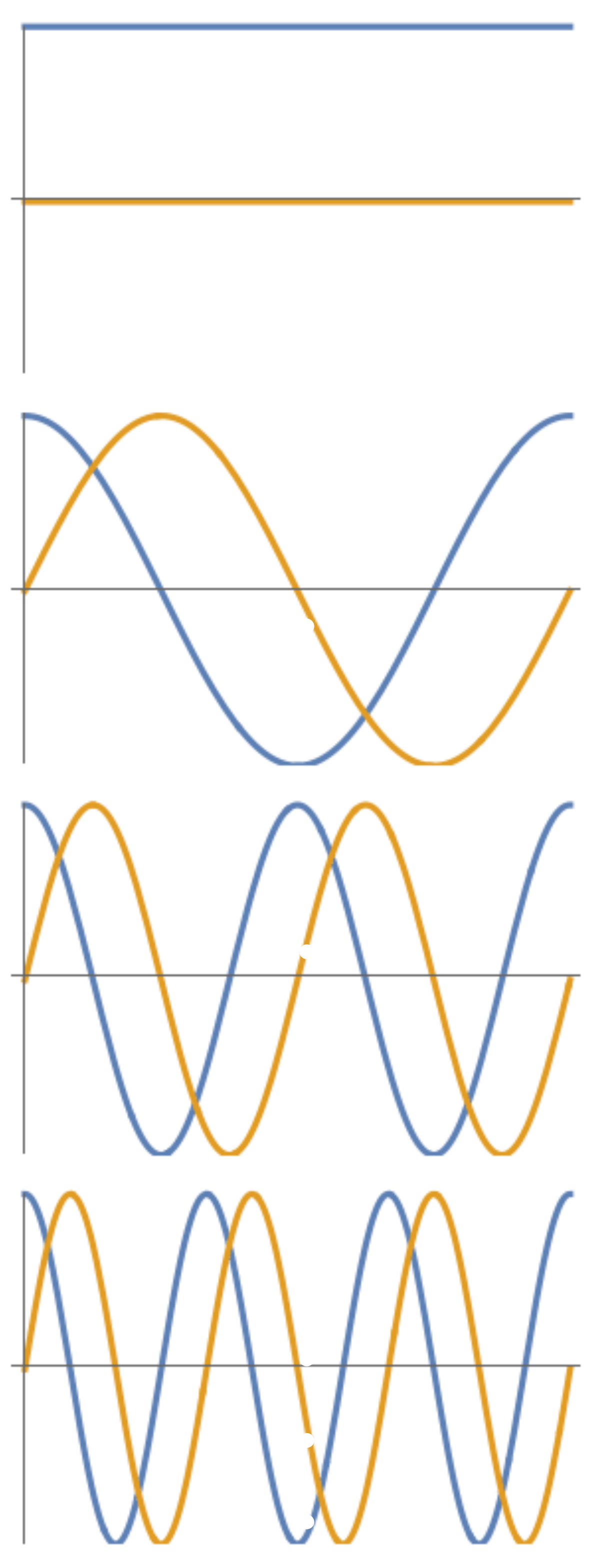




Steering the Fourier Basis

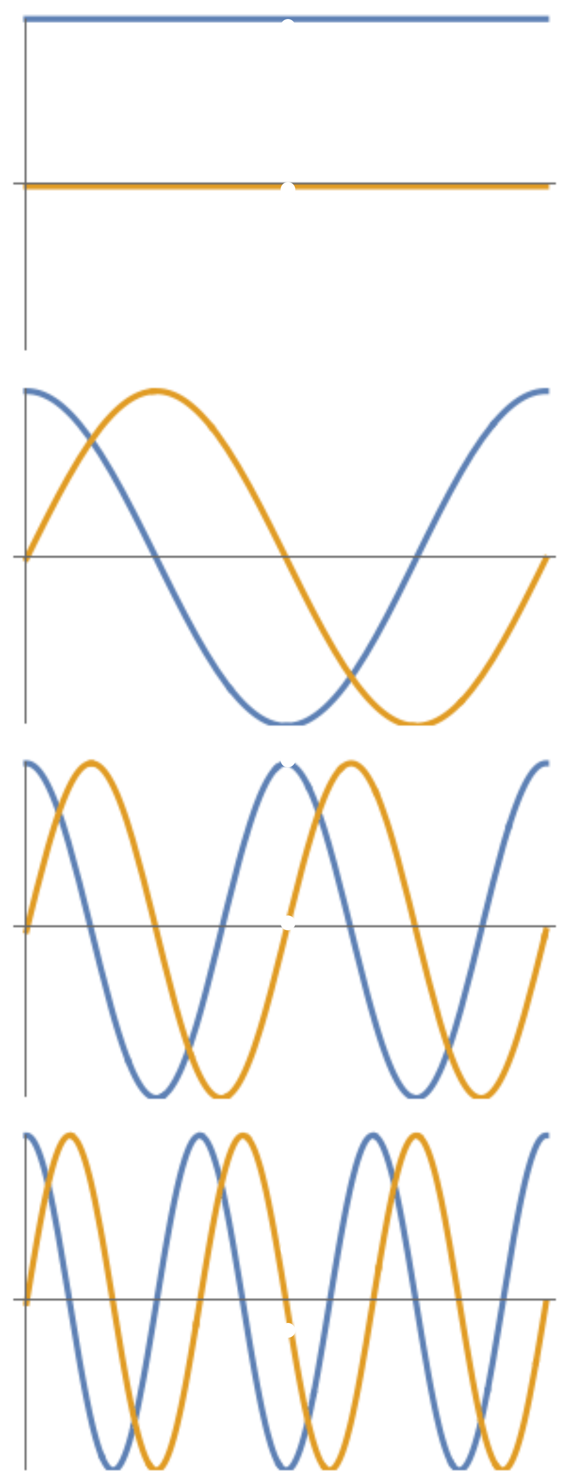
$$\rho(-\theta) = \bigoplus_{l=-L}^{L} \rho_l(-\theta)$$

$Y(\alpha - \theta)$



$$= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cos\theta_1 & -\sin\theta_1 & 0 & 0 & 0 & 0 \\ 0 & \sin\theta_1 & \cos\theta_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\theta_2 & -\sin\theta_2 & 0 & 0 \\ 0 & 0 & 0 & \sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cos\theta_3 & -\sin\theta_3 \\ 0 & 0 & 0 & 0 & 0 & \sin\theta_3 & \cos\theta_3 \end{pmatrix}$$

$Y(\alpha)$



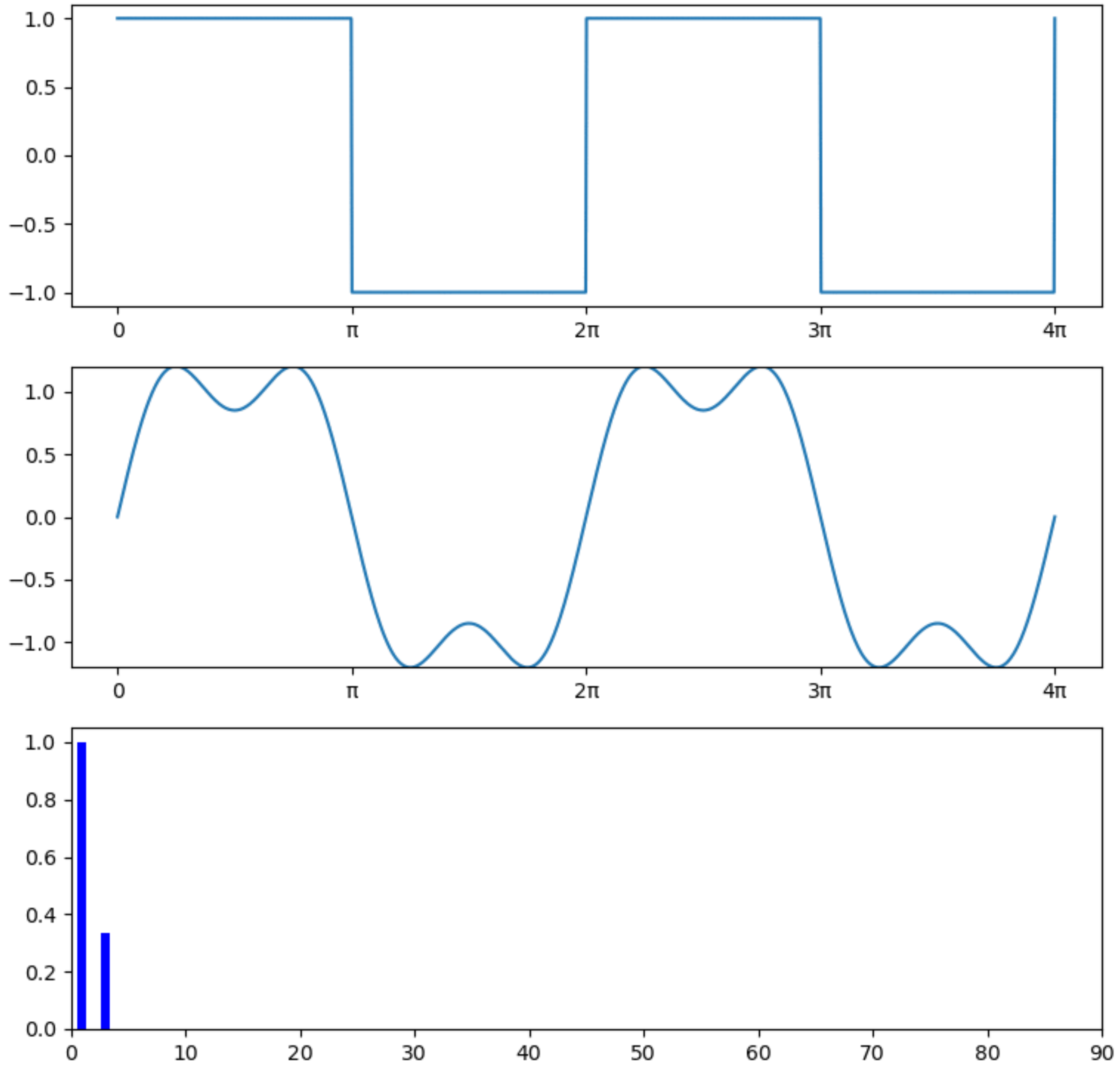
1D Fourier Basis

Any(**) univariate function can be expressed as a weighted sum of sinusoids of different frequencies (1807)

Jean-Baptiste Joseph Fourier (1768-1830)

Example: series for a square wave

$$\sum_{k=1,3,5,\ldots}^{\infty} \frac{1}{k}\sin(kt)$$



Slide credit: S. Lazebnik, C. Rupprecht

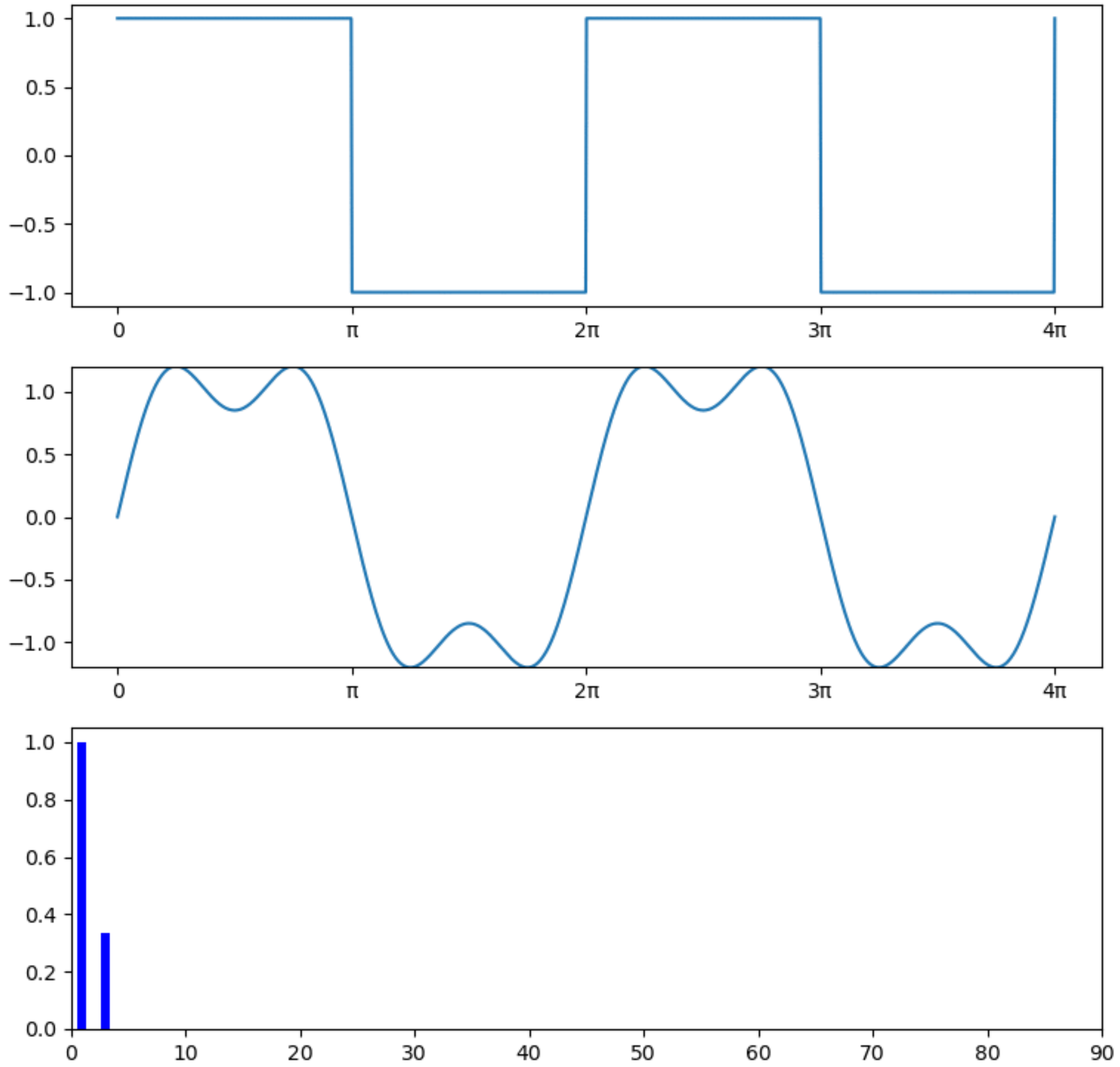
1D Fourier Basis

Example: series for a square wave

$$\sum_{k=1,3,5,\ldots}^{\infty} \frac{1}{k}\sin(kt)$$

Any(**) univariate function can be expressed as a weighted sum of sinusoids of different frequencies (1807)

Jean-Baptiste Joseph Fourier (1768-1830)



Slide credit: S. Lazebnik, C. Rupprecht

1D Fourier Basis

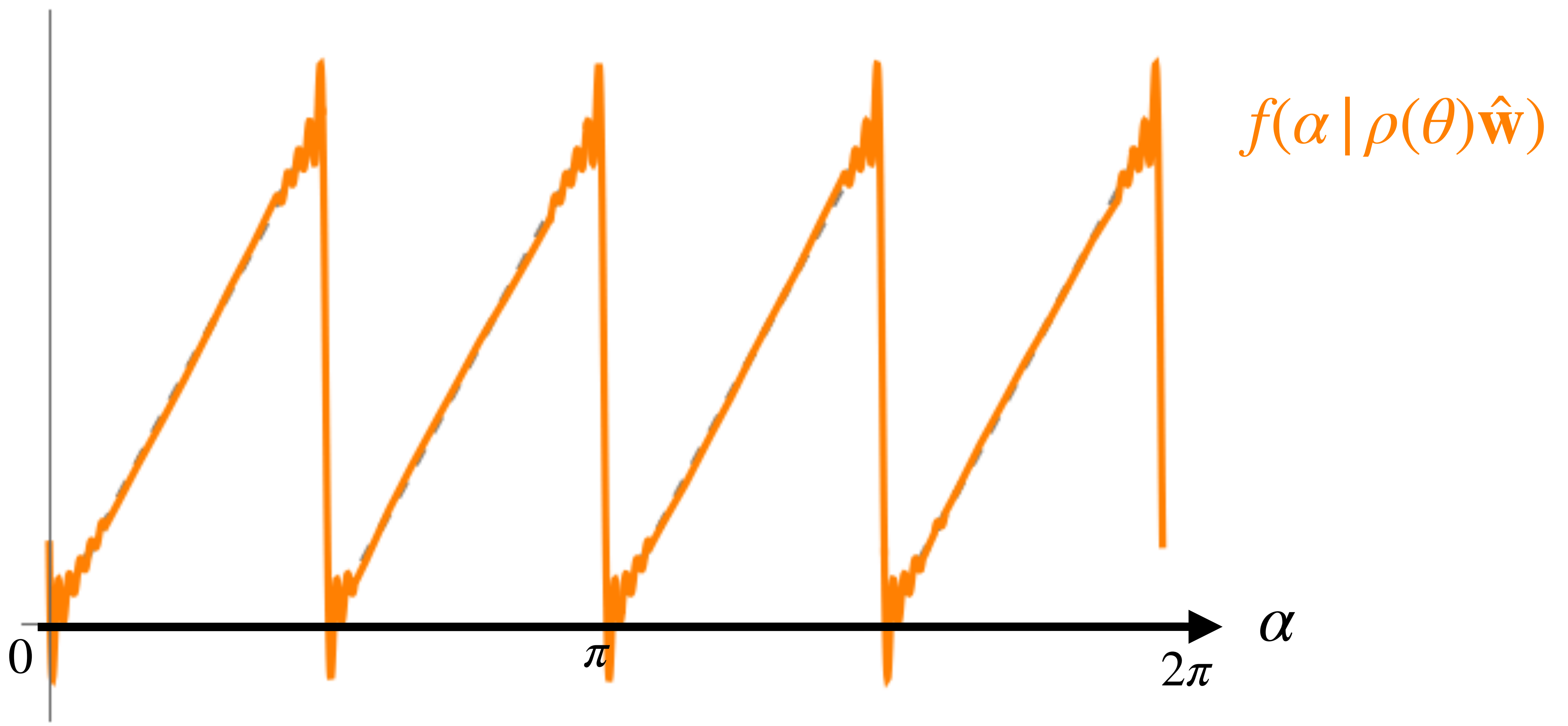
Let $$f(\alpha\,|\,\hat{\mathbf{w}}) = \hat{\mathbf{w}}^{\dagger} Y(\alpha)$$

Then we can steer/shift this function by transforming the weights $\hat{\mathbf{w}}$

$$f(\alpha - \theta\,|\,\hat{\mathbf{w}}) = f(\alpha\,|\,\rho(\theta)\hat{\mathbf{w}})$$

Proof:
$$\begin{aligned} f(\alpha - \theta\,|\,\hat{\mathbf{w}}) &= \hat{\mathbf{w}}^{\dagger} Y(\alpha - \theta) \\ &= \hat{\mathbf{w}}^{\dagger} \rho(-\theta) Y(\alpha) \\ &= \hat{\mathbf{w}}^{\dagger} \rho(\theta)^{\dagger} Y(\alpha) \\ &= (\rho(\theta)\hat{\mathbf{w}})^{\dagger} Y(\alpha) \\ &= f(\alpha\,|\,\rho(\theta)\hat{\mathbf{w}}) \end{aligned}$$

L=101



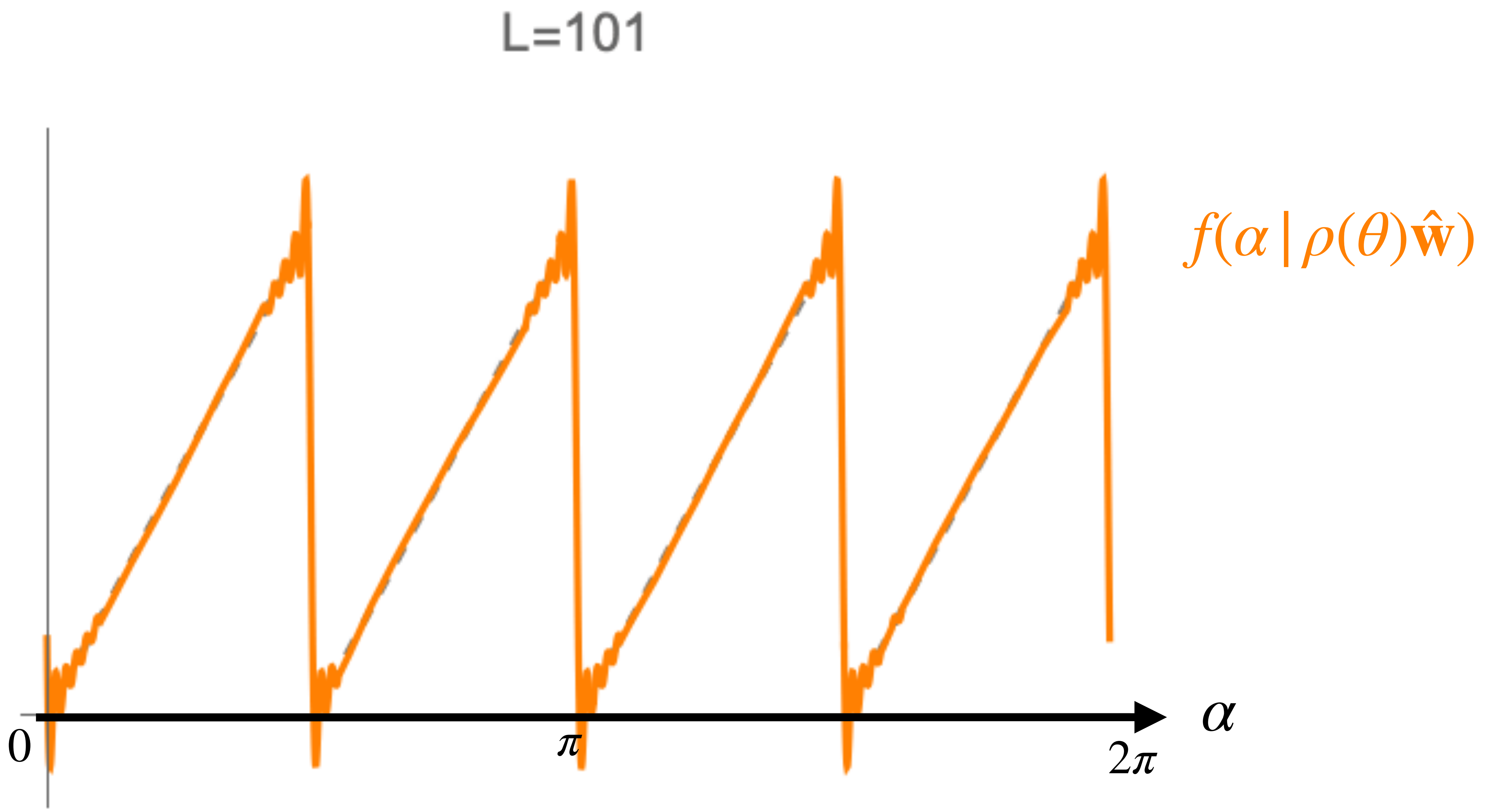
1D Fourier Basis

Let $$f(\alpha\,|\,\hat{\mathbf{w}}) = \hat{\mathbf{w}}^\dagger Y(\alpha)$$

Then we can steer/shift this function by transforming the weights $\hat{\mathbf{w}}$

$$f(\alpha - \theta\,|\,\hat{\mathbf{w}}) = f(\alpha\,|\,\rho(\theta)\hat{\mathbf{w}})$$

Proof: $$\begin{aligned} f(\alpha - \theta\,|\,\hat{\mathbf{w}}) &= \hat{\mathbf{w}}^\dagger Y(\alpha - \theta) \\ &= \hat{\mathbf{w}}^\dagger \rho(-\theta) Y(\alpha) \\ &= \hat{\mathbf{w}}^\dagger \rho(\theta)^\dagger Y(\alpha) \\ &= (\rho(\theta)\hat{\mathbf{w}})^\dagger Y(\alpha) \\ &= f(\alpha\,|\,\rho(\theta)\hat{\mathbf{w}}) \end{aligned}$$



The Fourier Basis is the steerable basis for circular shifts

The Fourier Basis is the steerable basis for circular shifts

The Fourier Basis is a *steerable basis* for circular shifts where they become block-diagonal!

The Fourier Basis is the steerable basis for circular shifts

Why should you care?

The Fourier Basis is the steerable basis for circular shifts

Translations aren't the *only* group of interest. You have already seen a number of others: SO(2), SO(3), SE(3)…

The Design and Use of Steerable Filters

William T. Freeman and Edward H. Adelson

1991

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 13, NO. 9, SEPTEMBER 1991 891

The Design and Use of Steerable Filters

William T. Freeman and Edward H. Adelson

1991

More recently :)

The Design and Use of Steerable Filters

William T. Freeman and Edward H. Adelson

I. INTRODUCTION

ORIENTED filters are used in many vision and image processing tasks, such as texture analysis, edge detection, image data compression, motion analysis, and image enhancement. In many of these tasks, it is useful to apply filters of arbitrary orientation under adaptive control and to examine the filter output as a function of both orientation and phase.

The Design and Use of Steerable Filters

William T. Freeman and Edward H. Adelson

I. INTRODUCTION

ORIENTED filters are used in many vision and image processing tasks, such as texture analysis, edge detection, image data compression, motion analysis, and image enhance-
ment. In mar
arbitrary orie
the filter out

With the correct filter set and the correct interpolation rule, it is possible to determine the response of a filter of arbitrary orientation without explicitly applying that filter.

The Design and Use of Steerable Filters

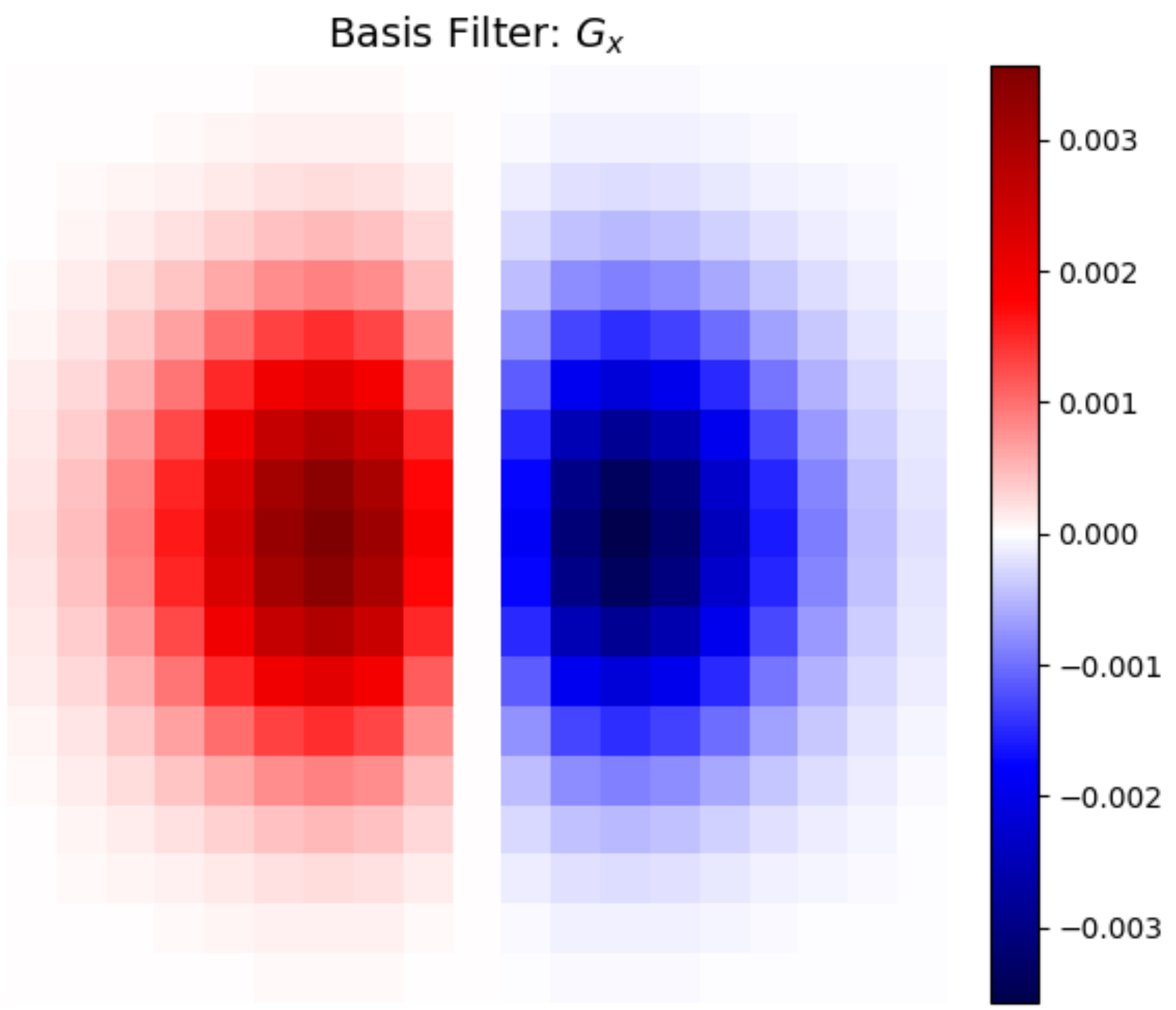
William T. Freeman and Edward H. Adelson

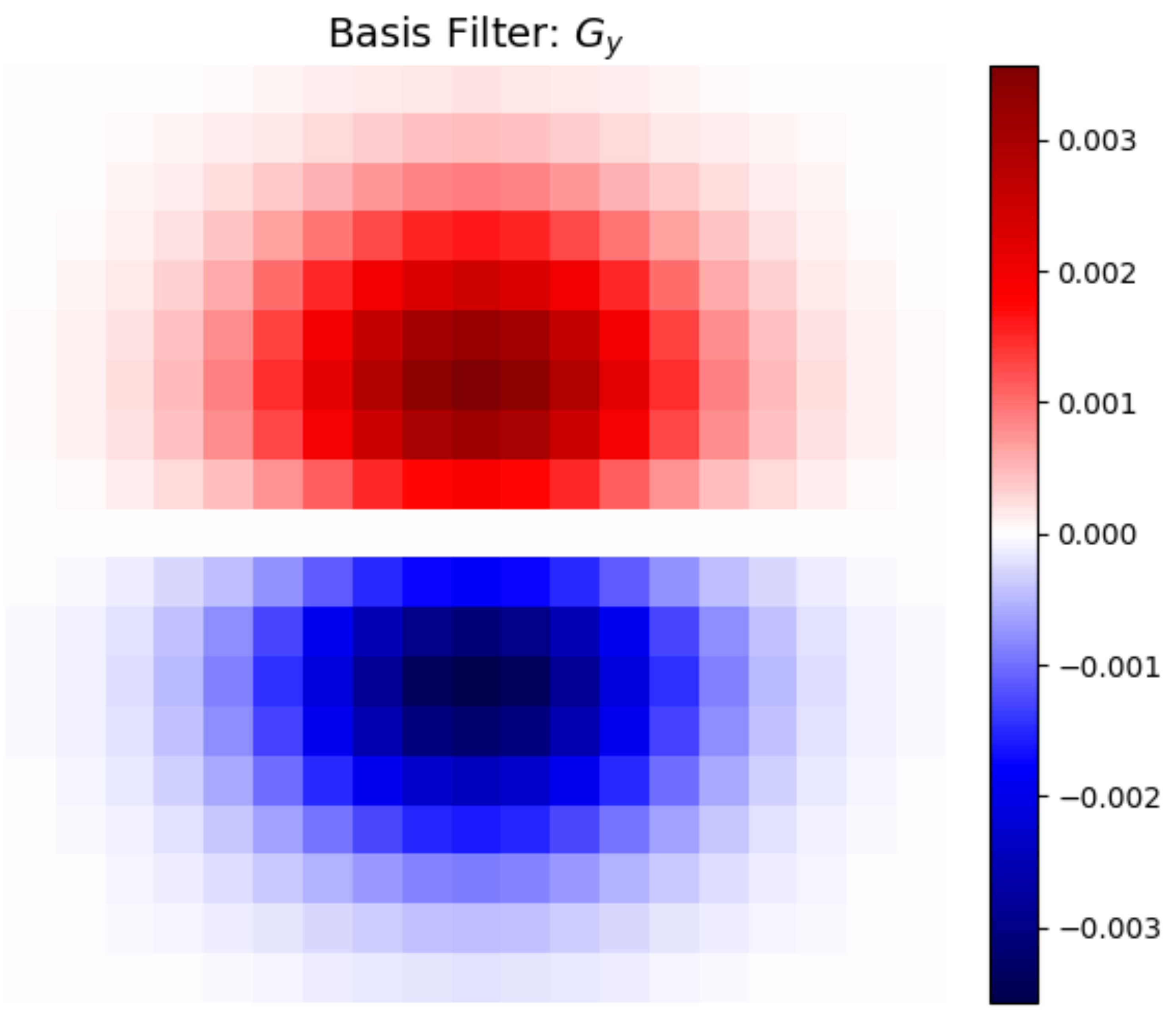
I. INTRODUCTION

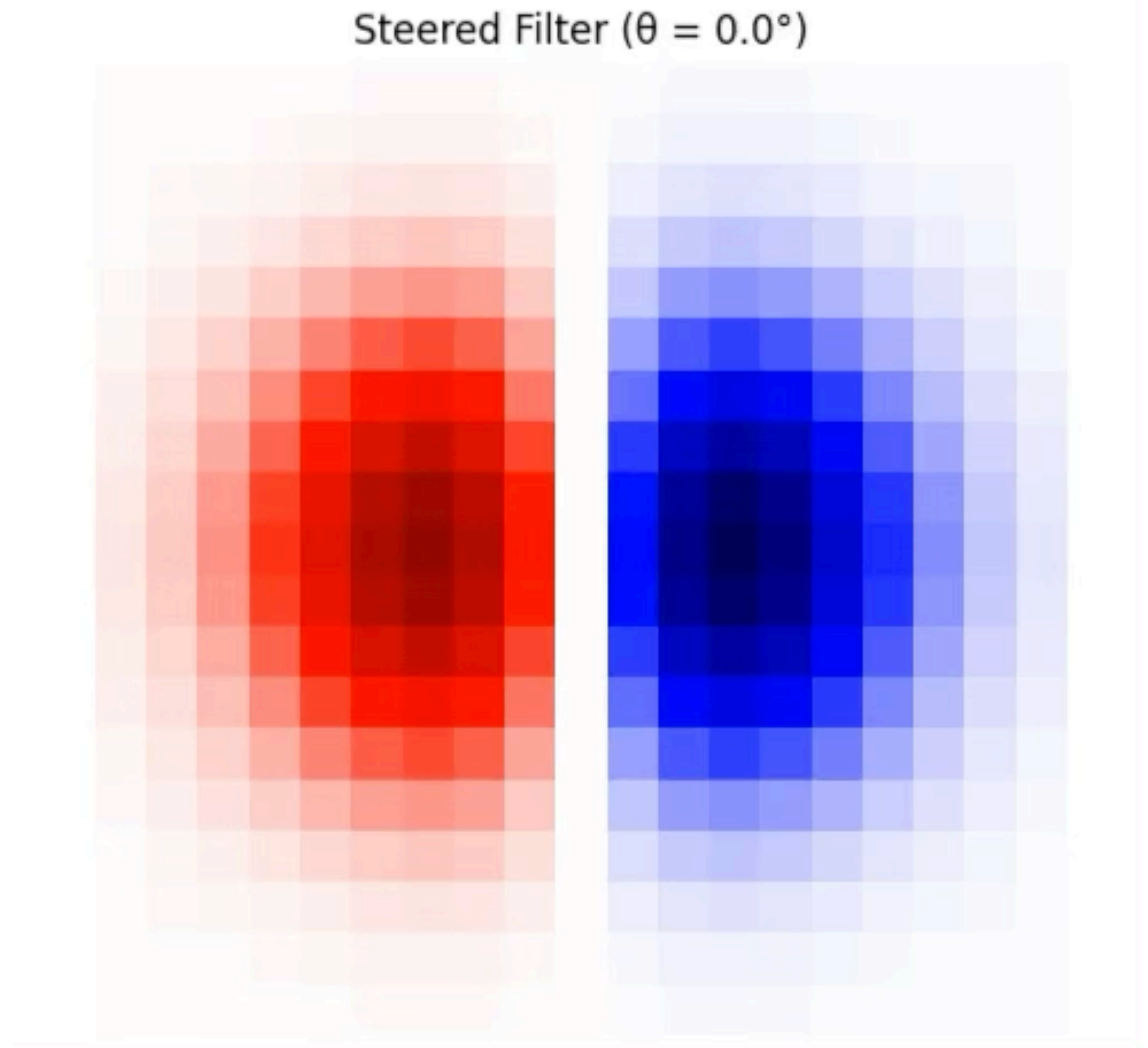
Derives a basis for *rotation-steerable* 2D image filters!

With the correct filter set and the correct interpolation rule, it is possible to determine the response of a filter of arbitrary orientation without explicitly applying that filter.

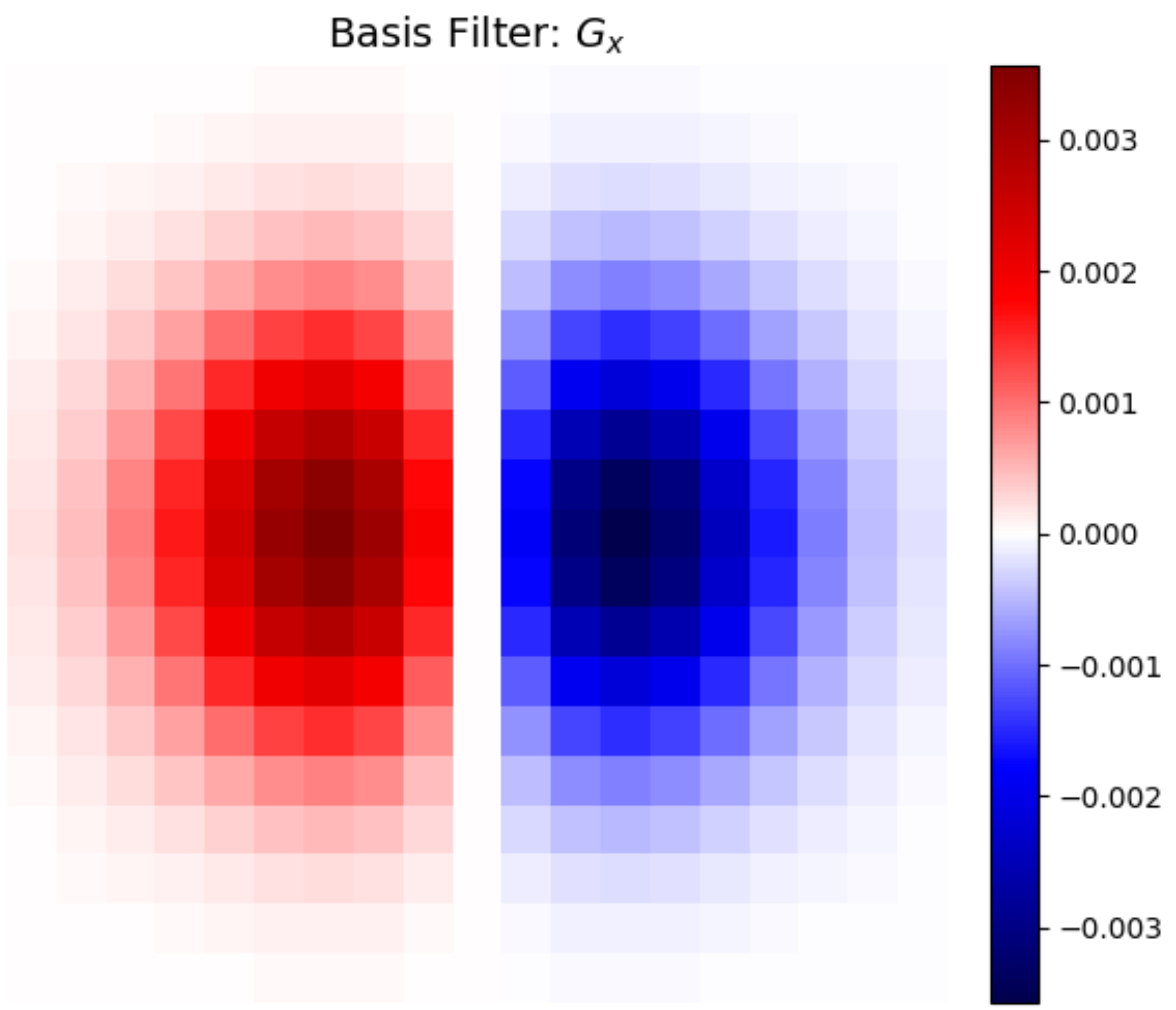
Steerable Filters

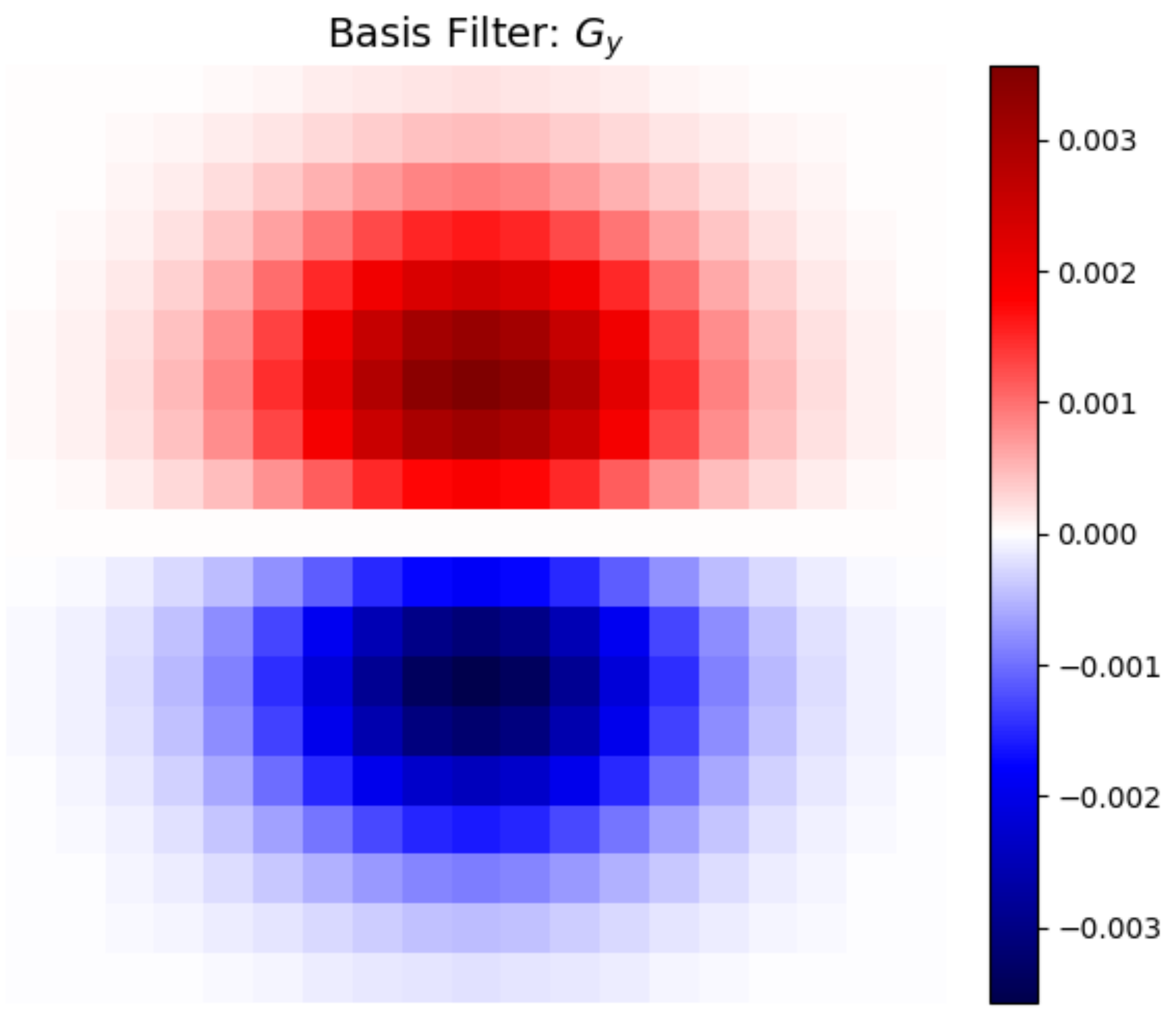


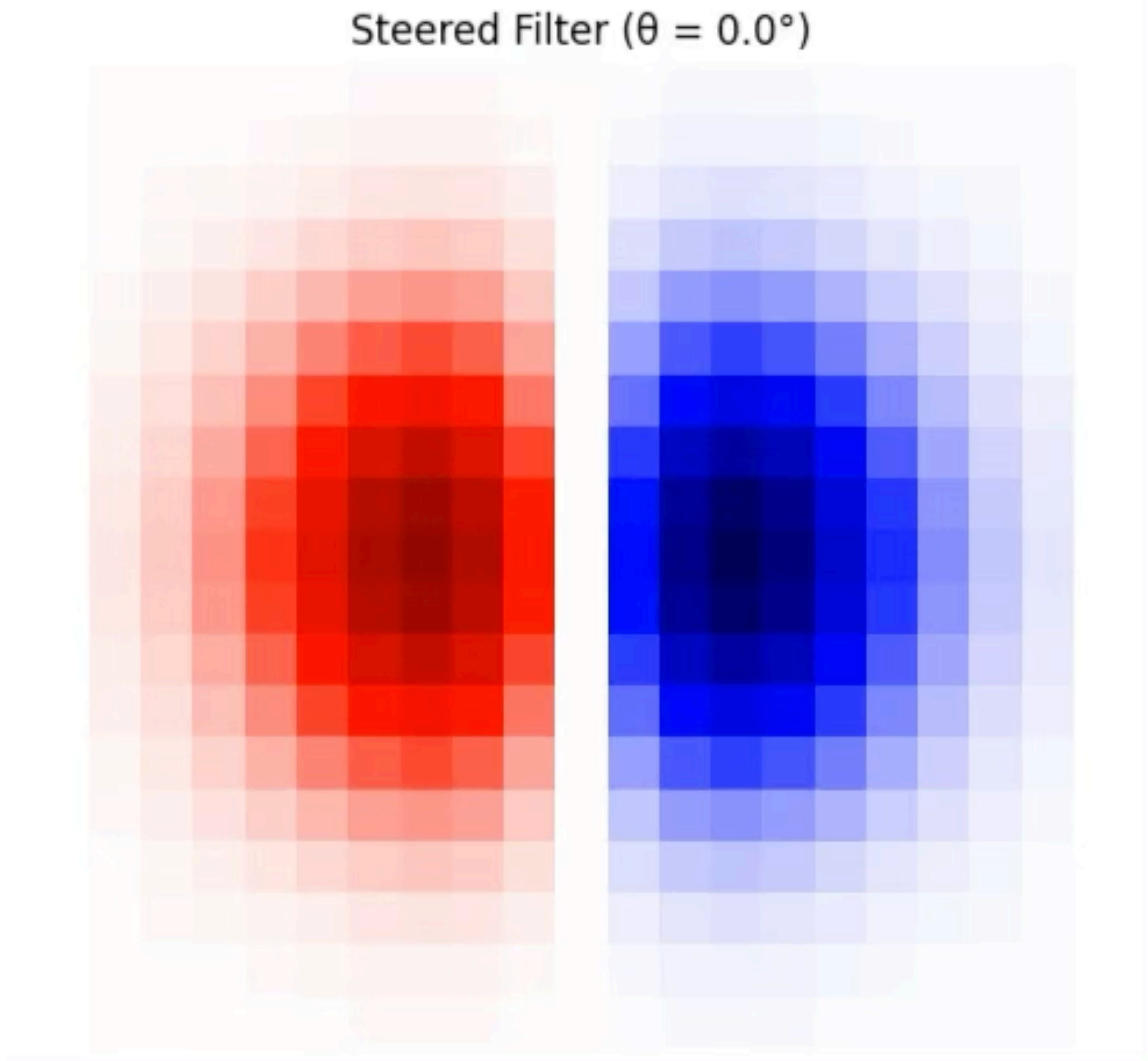




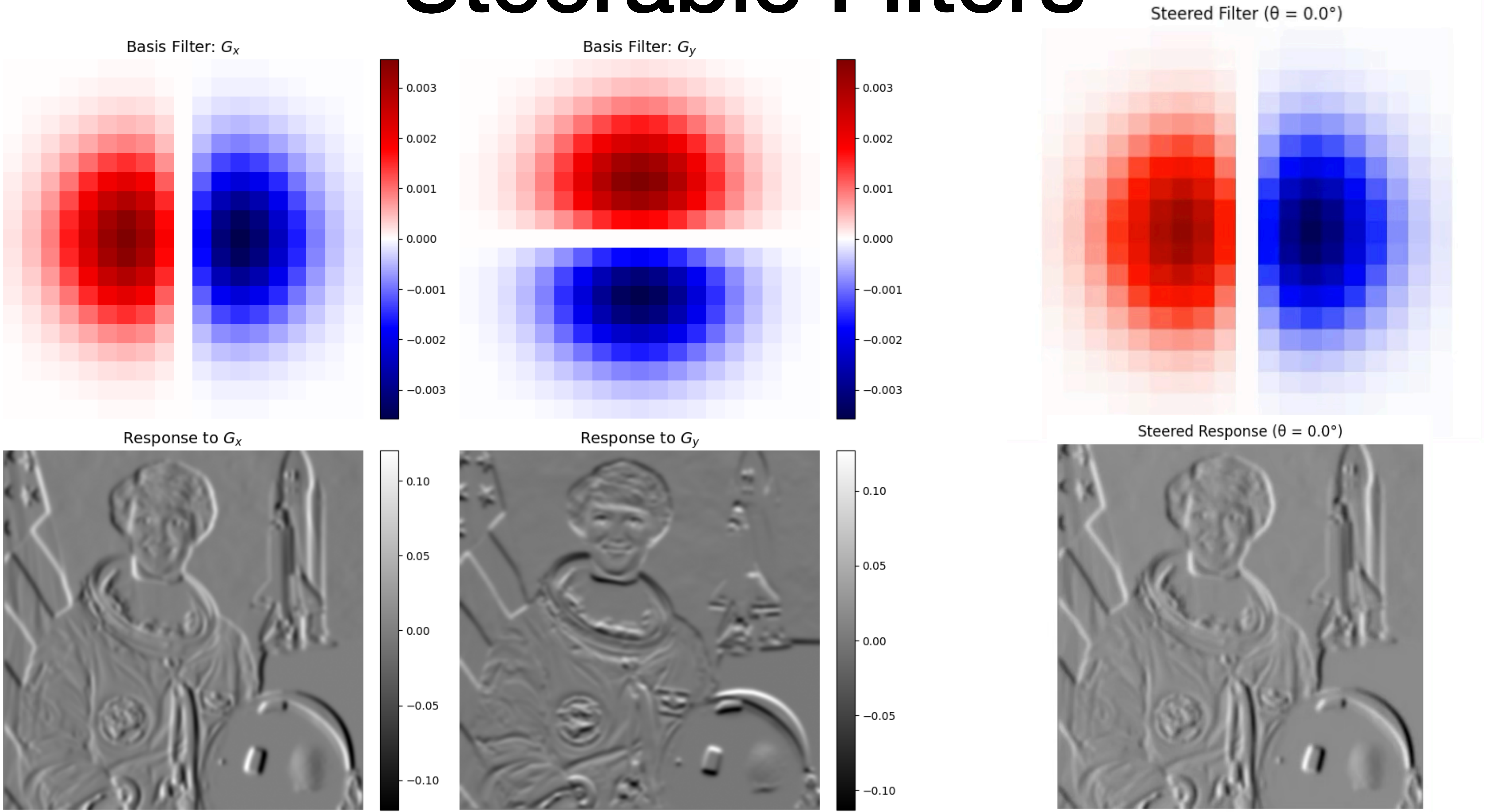
Steerable Filters



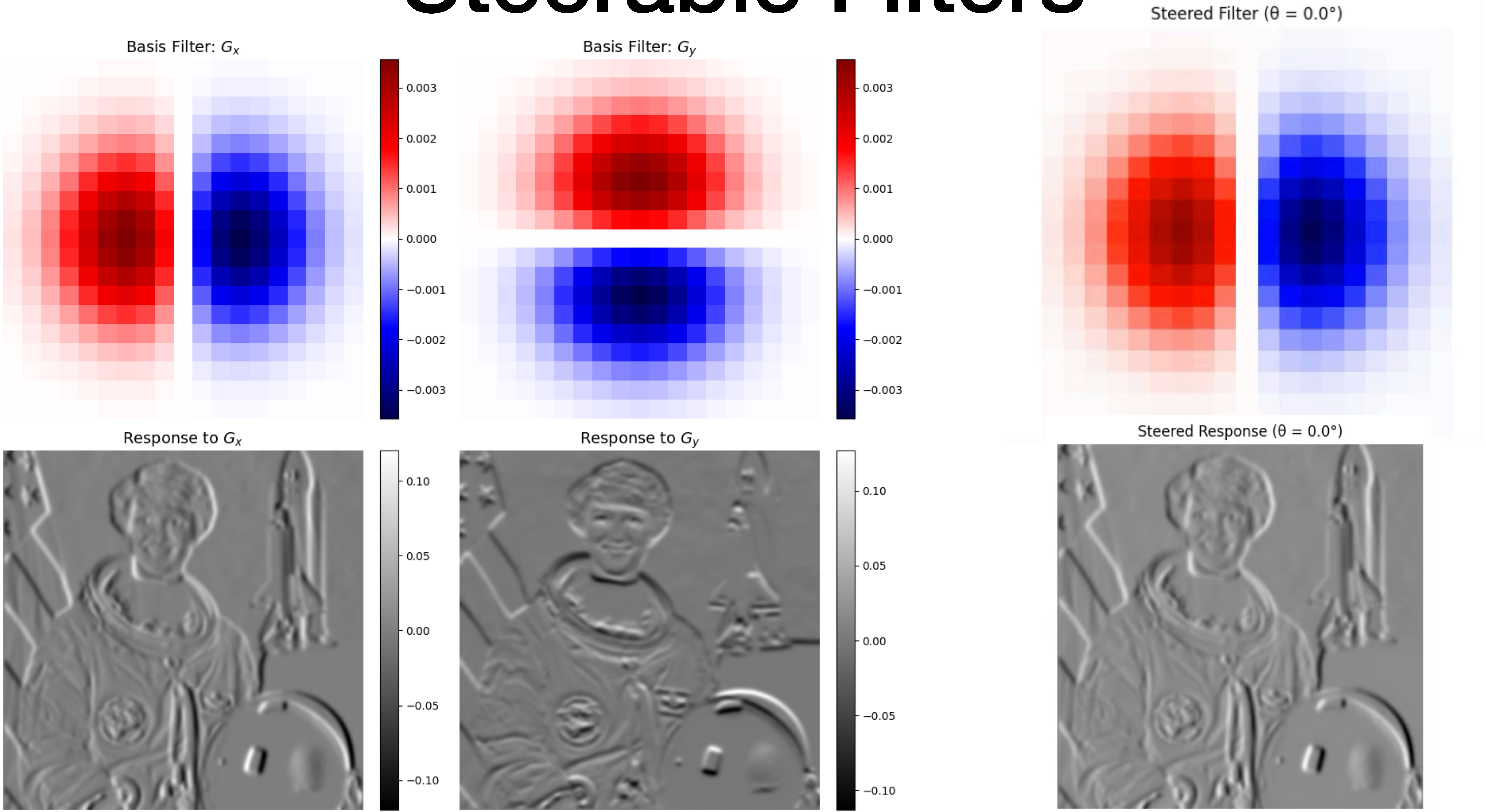




Steerable Filters

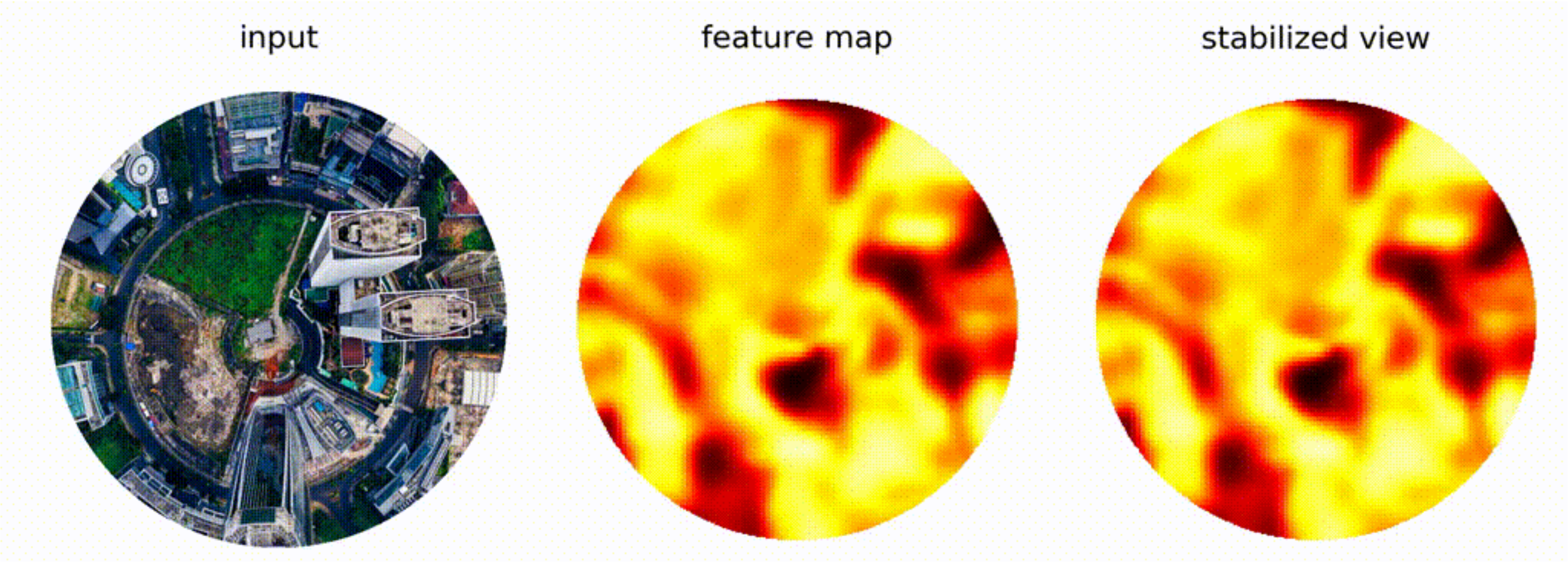


Steerable Filters



Geometric guarantees (equivariance)

CNN

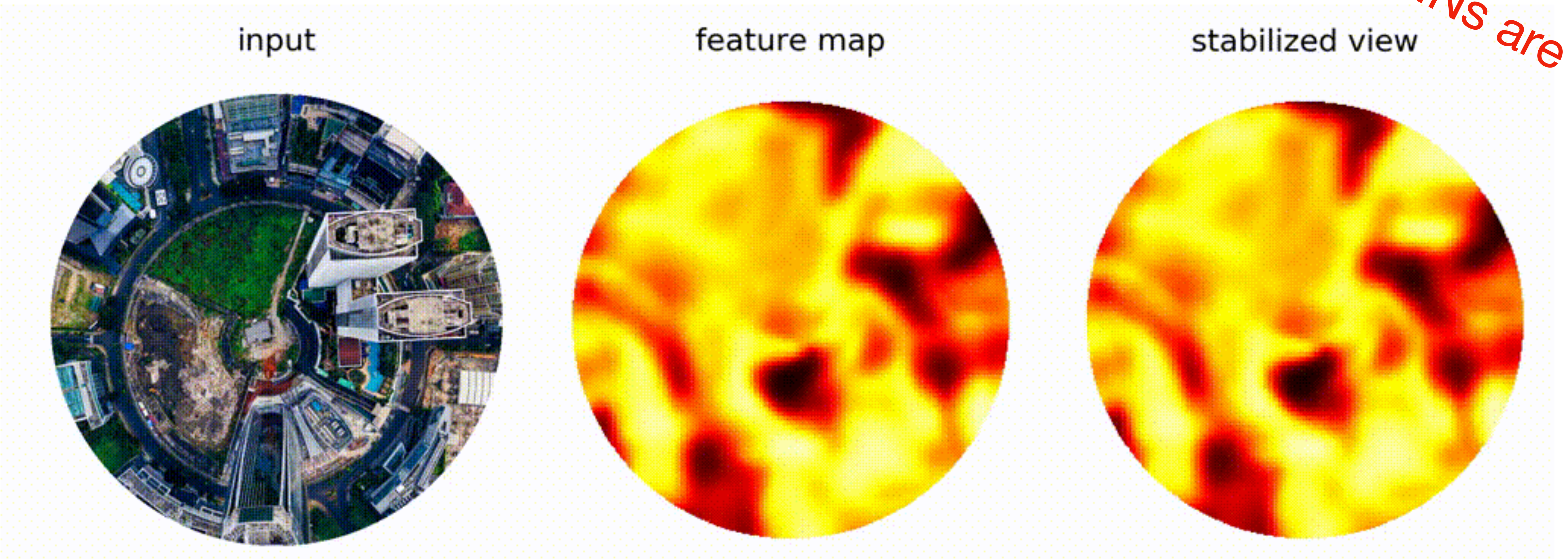


Figures source:
https://github.com/QUVA-Lab/e2cnn

Slide curtesy of Erik Bekkers from UVA Deep Learning II Course

Geometric guarantees (equivariance)

CNN



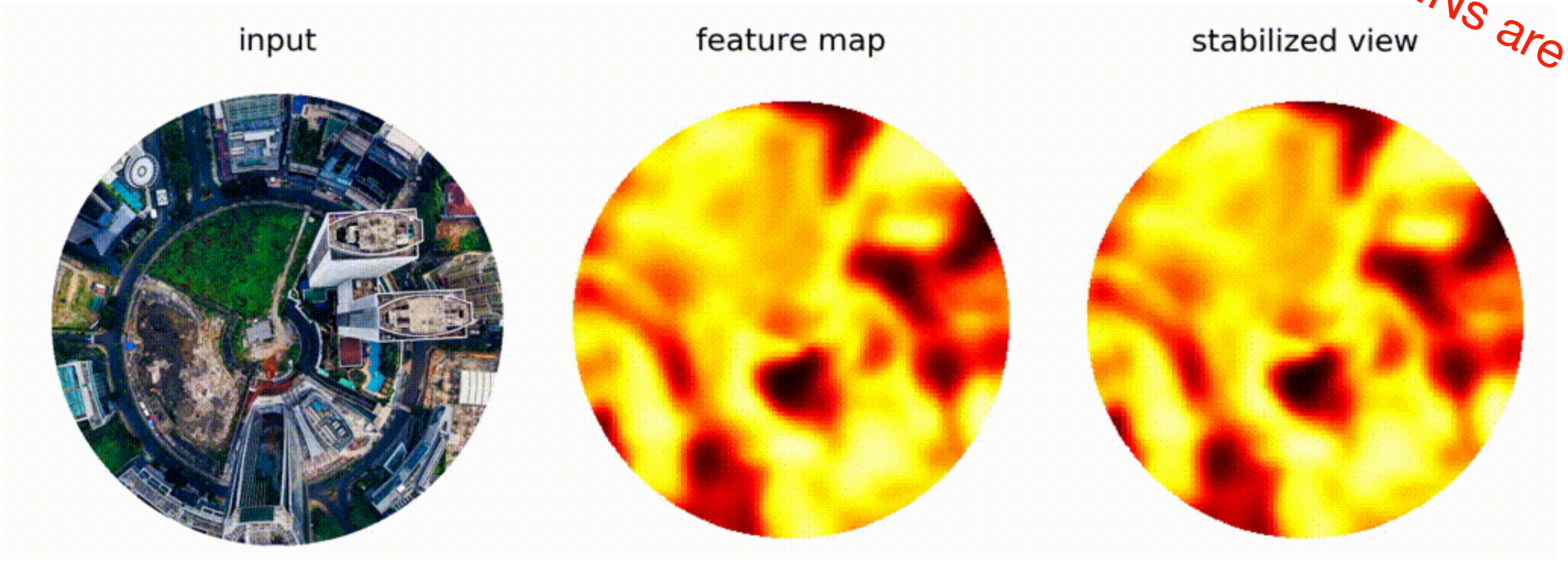
CNNs are not rotation equivariant

Figures source:
https://github.com/QUVA-Lab/e2cnn

Slide curtesy of Erik Bekkers from UVA Deep Learning II Course

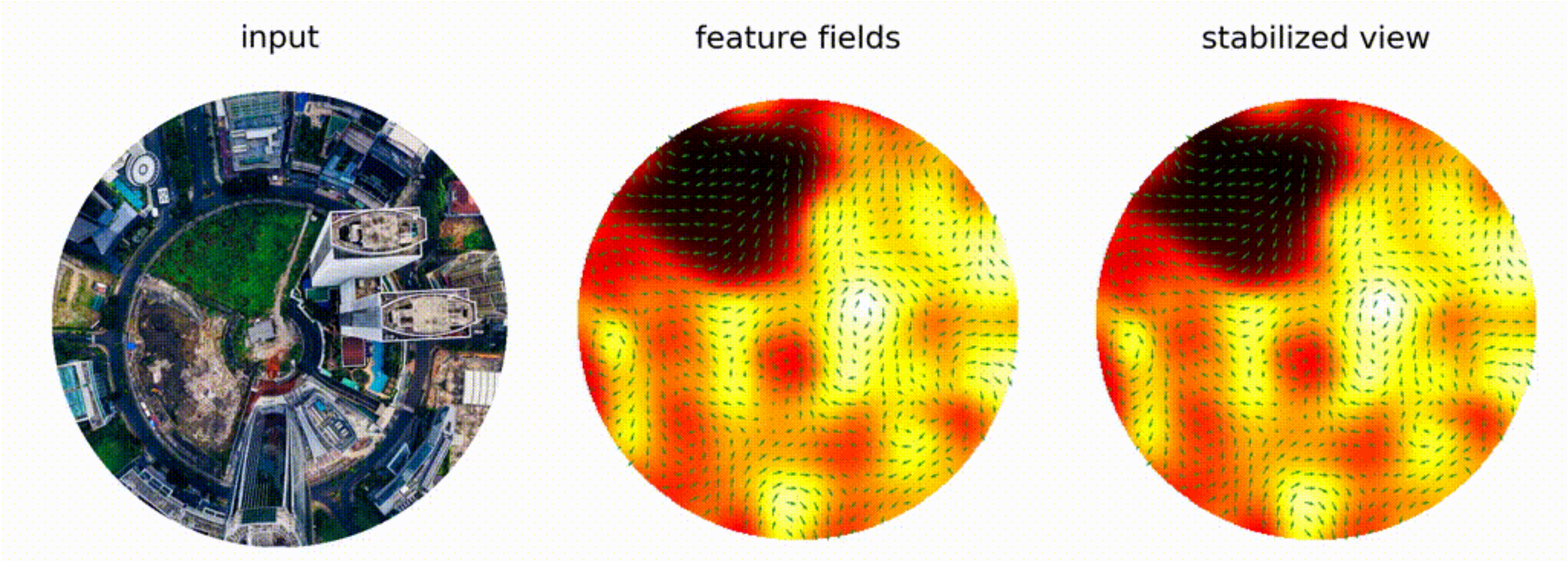
Geometric guarantees (equivariance)

CNN



CNNs are not rotation equivariant

Equivariant NN

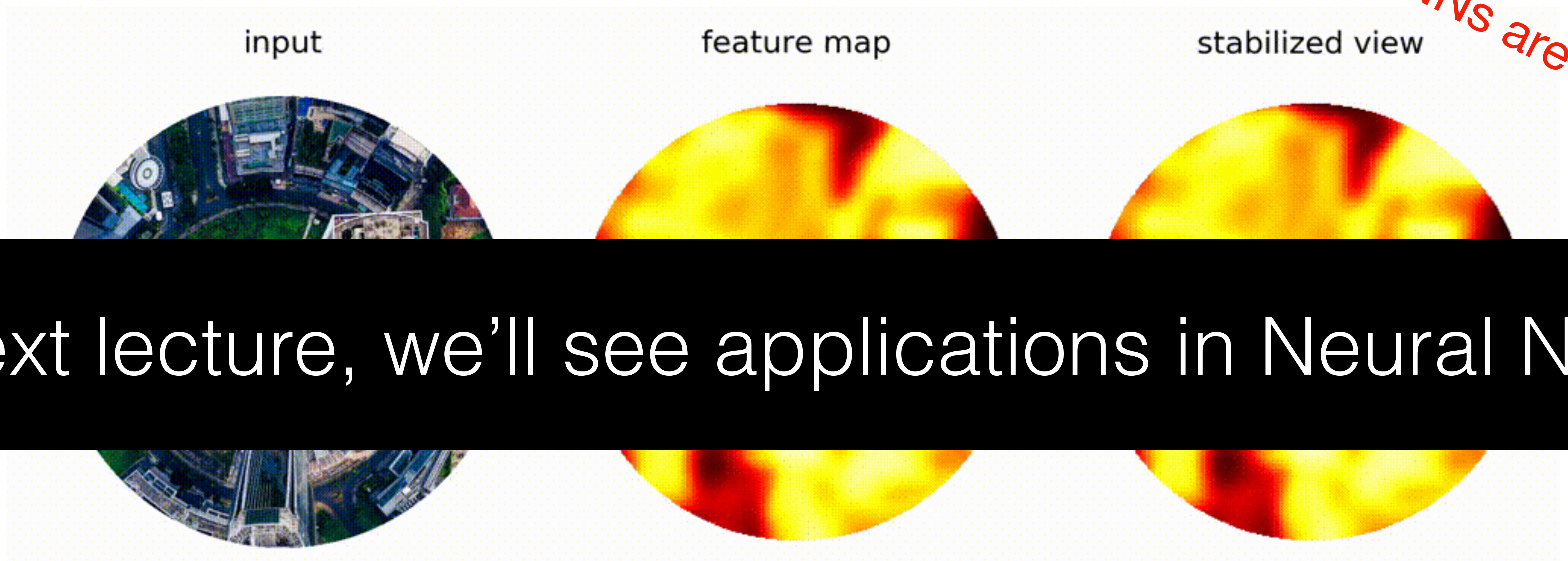


Figures source:
https://github.com/QUVA-Lab/e2cnn

Geometric guarantees (equivariance)

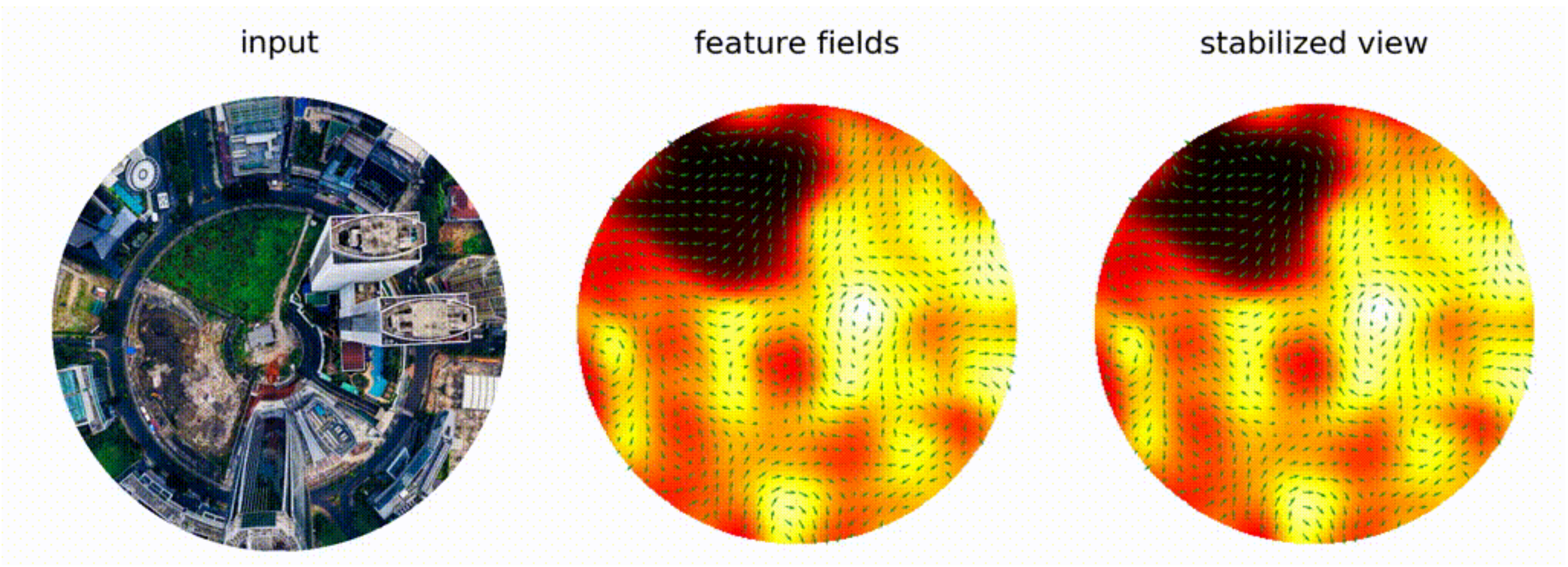
CNNs are not rotation equivariant

CNN



In the next lecture, we'll see applications in Neural Nets!

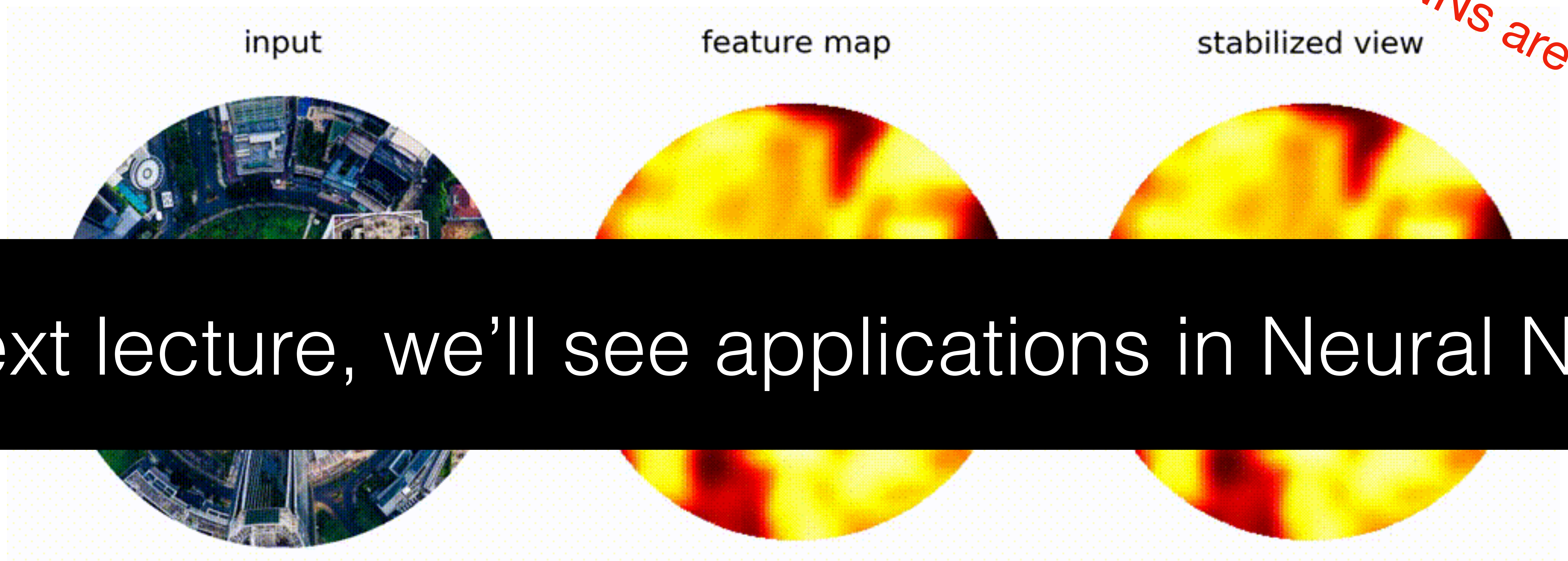
Equivariant NN



Figures source:
https://github.com/QUVA-Lab/e2cnn

Geometric guarantees (equivariance)

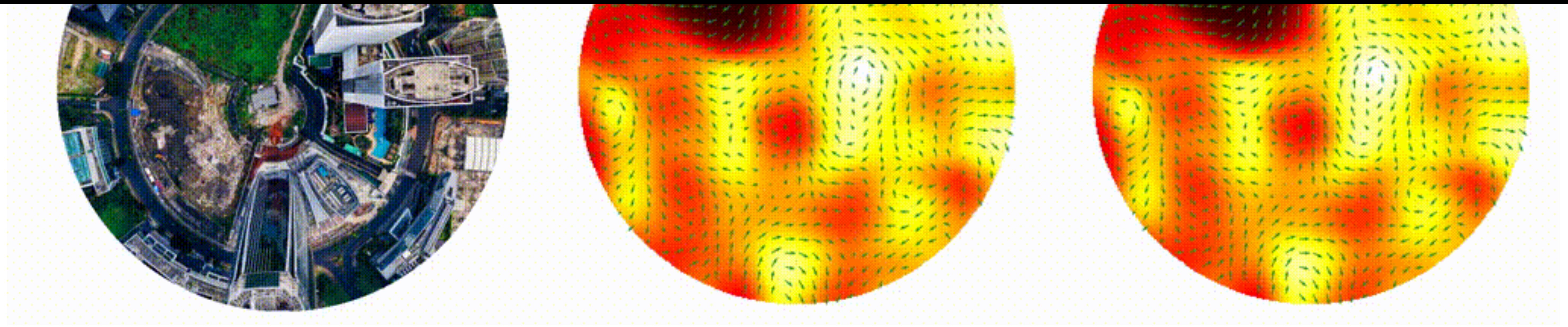
CNN



CNNs are not rotation equivariant

In the next lecture, we'll see applications in Neural Nets!

How can we *find* a steerable basis for a given transformation?



Figures source:
https://github.com/QUVA-Lab/e2cnn

Recall: The Eigenvalue Problem

Recall: The Eigenvalue Problem

Recall the eigenvalue problem for matrices:

Recall: The Eigenvalue Problem

Recall the eigenvalue problem for matrices:

$$A\mathbf{v} = \lambda\mathbf{v}\,.$$

Recall: The Eigenvalue Problem

Recall the eigenvalue problem for matrices:

$$A\mathbf{v} = \lambda\mathbf{v}\,.$$

Eigenvectors $\mathbf{v}$ are special vectors whose direction is preserved under the action of A, though their magnitude may be scaled by the eigenvalue λ.

Recall: The Eigenvalue Problem

Recall the eigenvalue problem for matrices:

$$A\mathbf{v} = \lambda\mathbf{v}\,.$$

Eigenvectors $\mathbf{v}$ are special vectors whose direction is preserved under the action of A, though their magnitude may be scaled by the eigenvalue λ.

Now, what happens if A is replaced by a linear operator L that acts on functions? In this case, we look for *eigenfunctions*, which are functions that are preserved in shape, up to a scaling factor, under the action of L. The eigenvalue λ represents how the eigenfunction is scaled.

The Spectrum of Self-Adjoint Operators

The Spectrum of Self-Adjoint Operators

Let T be a linear operator on an inner product space V, and let $\mathbf{w}, \mathbf{v} \in V$.
Then we call T self-adjoint iff:

The Spectrum of Self-Adjoint Operators

Let T be a linear operator on an inner product space V, and let $\mathbf{w}, \mathbf{v} \in V$.
Then we call T self-adjoint iff:

$$< T\mathbf{w}, \mathbf{v} > = < \mathbf{w}, T\mathbf{v} > .$$

The Spectrum of Self-Adjoint Operators

Let T be a linear operator on an inner product space V, and let $\mathbf{w}, \mathbf{v} \in V$. Then we call T self-adjoint iff:

$$< T\mathbf{w}, \mathbf{v} > = < \mathbf{w}, T\mathbf{v} > .$$

For a finite-dimensional V, this is equivalent to T being *hermitian*, or in the real case, *symmetric.*

The Spectrum of Self-Adjoint Operators

Let T be a linear operator on an inner product space V, and let $\mathbf{w}, \mathbf{v} \in V$.
Then we call T self-adjoint iff:

$$< T\mathbf{w}, \mathbf{v} > = < \mathbf{w}, T\mathbf{v} > .$$

For a finite-dimensional V, this is equivalent to T being *hermitian*, or in the real case, *symmetric.*

Fact (special case of the spectral theorem):
The eigenvectors of a self-adjoint operator form an orthogonal basis for the vector space V.

The Spectrum of Self-Adjoint Operators

Let T be a linear operator on an inner product space V, and let $\mathbf{w}, \mathbf{v} \in V$.
Then we call T self-adjoint iff:

$$< T\mathbf{w}, \mathbf{v} > = < \mathbf{w}, T\mathbf{v} >$$

Let's look at some examples!

the real case, *symmetric.*

Fact (special case of the spectral theorem):
The eigenvectors of a self-adjoint operator form an orthogonal basis for the vector space V.

Examples of Real Symmetric Operators and their Eigenspectrum

Examples of Real Symmetric Operators and their Eigenspectrum

Consider the matrix

Examples of Real Symmetric Operators and their Eigenspectrum

Consider the matrix

$$B = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix}$$

Examples of Real Symmetric Operators and their Eigenspectrum

Consider the matrix

$$B = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix}$$

Examples of Real Symmetric Operators and their Eigenspectrum

Consider the matrix

$$B = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix}$$

Hermitian?

Examples of Real Symmetric Operators and their Eigenspectrum

Consider the matrix

$$B = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix}$$

Hermitian? No!

Examples of Real Symmetric Operators and their Eigenspectrum

$$B = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix}$$

has a single eigenvalue $\lambda = 2$ with multiplicity 3,
and a single eigenvector:

$$\mathbf{v} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

Basis for $\mathbb{R}^3$?

Examples of Real Symmetric Operators and their Eigenspectrum

$$B = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix}$$

has a single eigenvalue $\lambda = 2$ with multiplicity 3,
and a single eigenvector:

$$\mathbf{v} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

Basis for $\mathbb{R}^3$? No.

Examples of Real Symmetric Operators and their Eigenspectrum

Examples of Real Symmetric Operators and their Eigenspectrum

Consider the matrix

Examples of Real Symmetric Operators and their Eigenspectrum

Consider the matrix

$$A = \begin{pmatrix} 3 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

Examples of Real Symmetric Operators and their Eigenspectrum

Consider the matrix

$$A = \begin{pmatrix} 3 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

Examples of Real Symmetric Operators and their Eigenspectrum

Consider the matrix

$$A = \begin{pmatrix} 3 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

Hermitian?

Examples of Real Symmetric Operators and their Eigenspectrum

Consider the matrix

$$A = \begin{pmatrix} 3 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

Hermitian? Yes.

Examples of Real Symmetric Operators and their Eigenspectrum

$$A = \begin{pmatrix} 3 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

A has eigenvalues $\lambda_1 = 4$, and $\lambda_2 = 2$,
where the latter is *degenerate* with multiplicity 2. Eigenvectors are:

$$\mathbf{v}_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \mathbf{v}_2 = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} \text{ and } \mathbf{v}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Basis for $\mathbb{R}^3$?

Examples of Real Symmetric Operators and their Eigenspectrum

$$A = \begin{pmatrix} 3 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

A has eigenvalues $\lambda_1 = 4$, and $\lambda_2 = 2$,
where the latter is *degenerate* with multiplicity 2. Eigenvectors are:

$$\mathbf{v}_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \mathbf{v}_2 = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} \text{ and } \mathbf{v}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Basis for $\mathbb{R}^3$? Yes.

The Spectrum of Self-Adjoint Operators

Let T be a linear operator on an inner product space V, and let $\mathbf{w}, \mathbf{v} \in V$.
Then we call T self-adjoint iff:

$$< T\mathbf{w}, \mathbf{v} > = < \mathbf{w}, T\mathbf{v} > .$$

For a finite-dimensional V, this is equivalent to T being *hermitian*, or in the real case, *symmetric.*

The spectral theorem:
The eigenvectors of a self-adjoint operator form a basis for the vector space V.

The Spectrum of Self-Adjoint Operators

Let T be a linear operator on an inner product space V, and let $\mathbf{w}, \mathbf{v} \in V$.
Then we call T self-adjoint iff:

$$< T\mathbf{w}, \mathbf{v} > = < \mathbf{w}, T\mathbf{v} > .$$

Questions?

[redacted]

The spectral theorem:
The eigenvectors of a self-adjoint operator form
a basis for the vector space V.

Linear Combinations of Eigenvectors with the same Eigenvalue

Linear Combinations of Eigenvectors with the same Eigenvalue

Let $v_1, \ldots, v_n$ be a set of Eigenvectors of a matrix A with *the same eigenvalue* λ.

Linear Combinations of Eigenvectors with the same Eigenvalue

Let $v_1, \ldots, v_n$ be a set of Eigenvectors of a matrix A with *the same eigenvalue* λ.

Consider a linear combination of $v_1, \ldots, v_n$:

Linear Combinations of Eigenvectors with the same Eigenvalue

Let $v_1, \ldots, v_n$ be a set of Eigenvectors of a matrix A with *the same eigenvalue* λ.

Consider a linear combination of $v_1, \ldots, v_n$:

$$u = c_1\mathbf{v_1} + c_2\mathbf{v_2} + \ldots + c_n\mathbf{v_n}$$

Linear Combinations of Eigenvectors with the same Eigenvalue

Let $v_1, ..., v_n$ be a set of Eigenvectors of a matrix A with *the same eigenvalue* λ.

Consider a linear combination of $v_1, ..., v_n$:

$$u = c_1\mathbf{v_1} + c_2\mathbf{v_2} + ... + c_n\mathbf{v_n}$$

Then we have:

Linear Combinations of Eigenvectors with the same Eigenvalue

Let $v_1, \ldots, v_n$ be a set of Eigenvectors of a matrix A with *the same eigenvalue* λ.

Consider a linear combination of $v_1, \ldots, v_n$:

$$u = c_1\mathbf{v_1} + c_2\mathbf{v_2} + \ldots + c_n\mathbf{v_n}$$

Then we have:

$$A\mathbf{u} = A(c_1\mathbf{v_1} + \ldots + c_n\mathbf{v_n}) = c_1A\mathbf{v_1} + \ldots + c_nA\mathbf{v_n} = c_1\lambda\mathbf{v_1} + \ldots + c_n\lambda\mathbf{v_n} = \lambda\mathbf{u}$$

Linear Combinations of Eigenvectors with the same Eigenvalue

Let $v_1, ..., v_n$ be a set of Eigenvectors of a matrix A with *the same eigenvalue* λ.

Consider a linear combination of $v_1, ..., v_n$:

$$u = c_1\mathbf{v_1} + c_2\mathbf{v_2} + \ldots + c_n\mathbf{v_n}$$

Then we have:

$$A\mathbf{u} = A(c_1\mathbf{v_1} + \ldots + c_n\mathbf{v_n}) = c_1A\mathbf{v_1} + \ldots + c_nA\mathbf{v_n} = c_1\lambda\mathbf{v_1} + \ldots + c_n\lambda\mathbf{v_n} = \lambda\mathbf{u}$$

Thus, $\mathbf{u}$ is *also* an eigenvector of A. Hence, $\mathbf{v}_1, ..., \mathbf{v}_n$ spans a vector space, which we call an *eigenspace.*

Commuting Matrices

Assume two matrices commute:

$$AB = BA$$

What can we say about their eigenvectors and eigenvalues?

Commuting Matrices

Assume two matrices commute:

$$AB = BA$$

What can we say about their eigenvectors and eigenvalues?

Assume that ϕ is an eigenvector of A with eigenvalue λ. Then:

$$AB\phi = BA\phi = B\lambda\phi = \lambda B\phi\,.$$

Commuting Matrices

Assume two matrices commute:

$$AB = BA$$

What can we say about their eigenvectors and eigenvalues?

Assume that ϕ is an eigenvector of A with eigenvalue λ. Then:

$$AB\phi = \lambda B\phi\,.$$

In other words, $B\phi$ is *also* an eigenvector of A with the same eigenvalue!

An operator that commutes with a group action

Assume a **self-adjoint** operator L and a group transformation T such that:

$$LT = TL\,.$$

Further assume that ϕ is an eigenfunction of L. Then:

$$LT\phi = \lambda T\phi$$

In words, the *transformed eigenfunction* is still part of the same *eigenspace:* Its eigenvalue did not change!

An operator that commutes with a group action

An operator that commutes with a group action

Assume a **self-adjoint** operator L that **commutes** with a group transformation T. Then:

$$LT\phi = \lambda T\phi$$

In words, the *transformed eigenfunction* is still part of the same *eigenspace.*

An operator that commutes with a group action

Assume a **self-adjoint** operator L that **commutes** with a group transformation T. Then:

$$LT\phi = \lambda T\phi$$

In words, the *transformed eigenfunction* is still part of the same *eigenspace.*

For a *non-degenerate* λ_1, since $T\phi$ and ϕ are both eigenvectors of L with the same eigenvalue, that means that they must be scalar multiples of each other: $T\phi = a\phi$ for some a .

An operator that commutes with a group action

Assume a **self-adjoint** operator L that **commutes** with a group transformation T. Then:

$$LT\phi = \lambda T\phi$$

In words, the *transformed eigenfunction* is still part of the same *eigenspace.*

For a *non-degenerate* λ_1, since $T\phi$ and ϕ are both eigenvectors of L with the same eigenvalue, that means that they must be scalar multiples of each other: $T\phi = a\phi$ for some a .

For a degenerate eigenvalue λ_2, we know that the *eigenvalue* hasn't changed. That means that $T\phi$ must be a *linear combination* of all the eigenfunctions that belong to that eigenvalue:

$$T\phi = \sum_i c_{ij}\phi_i$$

An operator that commutes with a group action

Assume a **self-adjoint** operator L that **commutes** with a group transformation T. Then:

$$LT\phi = \lambda T\phi$$

In words, the *transformed eigenfunction* is still part of the same *eigenspace.*

For a *non-degenerate* λ_1, since $T\phi$ and ϕ are both eigenvectors of L with the same eigenvalue, that means that they must be scalar multiples of each other: $T\phi = a\phi$ for some a .

For a degenerate eigenvalue λ_2, we know that the *eigenvalue* hasn't changed. That means that $T\phi$ must be a *linear combination* of all the eigenfunctions that belong to that eigenvalue:

$$T\phi = \sum_i c_{ij}\phi_i$$

In other words, ϕ must be a linear combination of *other eigenfunctions in the same eigenspace!*

An operator that commutes with a group action

Assume a **self-adjoint** operator L that **commutes** with a group transformation T. Then:

[redacted]

This is **exactly** the steerability property!

Fo[redacted]ue, that means that they must be scalar multiples of each other: $T\phi = a\phi$ for some a.

For a degenerate eigenvalue λ_2, we know that the *eigenvalue* hasn't changed. That means that $T\phi$ must be a *linear combination* of all the eigenfunctions that belong to that eigenvalue:

$$T\phi = \sum_i c_{ij}\phi_i$$

In other words, ϕ must be a linear combination of *other eigenfunctions in the same eigenspace!*

An operator that commutes with a group action

Assume a **self-adjoint** operator L that **commutes** with a group transformation T. Then:

This is **exactly** the steerability property!

Fo ... ue,

that means that they must be scalar multiples of each other: $T\phi = a\phi$ for some a .

Fo ... at

Finally, b/c L is self-adjoint, we can express any function as a linear combination of its eigenvectors!

In other words, ϕ must be a linear combination of *other eigenfunctions in the same eigenspace!*

Recipe for finding a steerable basis in which group action is “simple”

1. Find a *linear operator* such that…
   1. …it *commutes* with the group action. This means that its eigenspace is unchanged under the group action.
   2. …its eigenvectors are guaranteed to be a complete basis for the function space (it is self-adjoint).
2. Compute its Eigendecomposition.
3. The eigenfunctions then **must form a steerable basis** for the group transformation!

Recipe for finding a steerable basis in which group action is “simple”

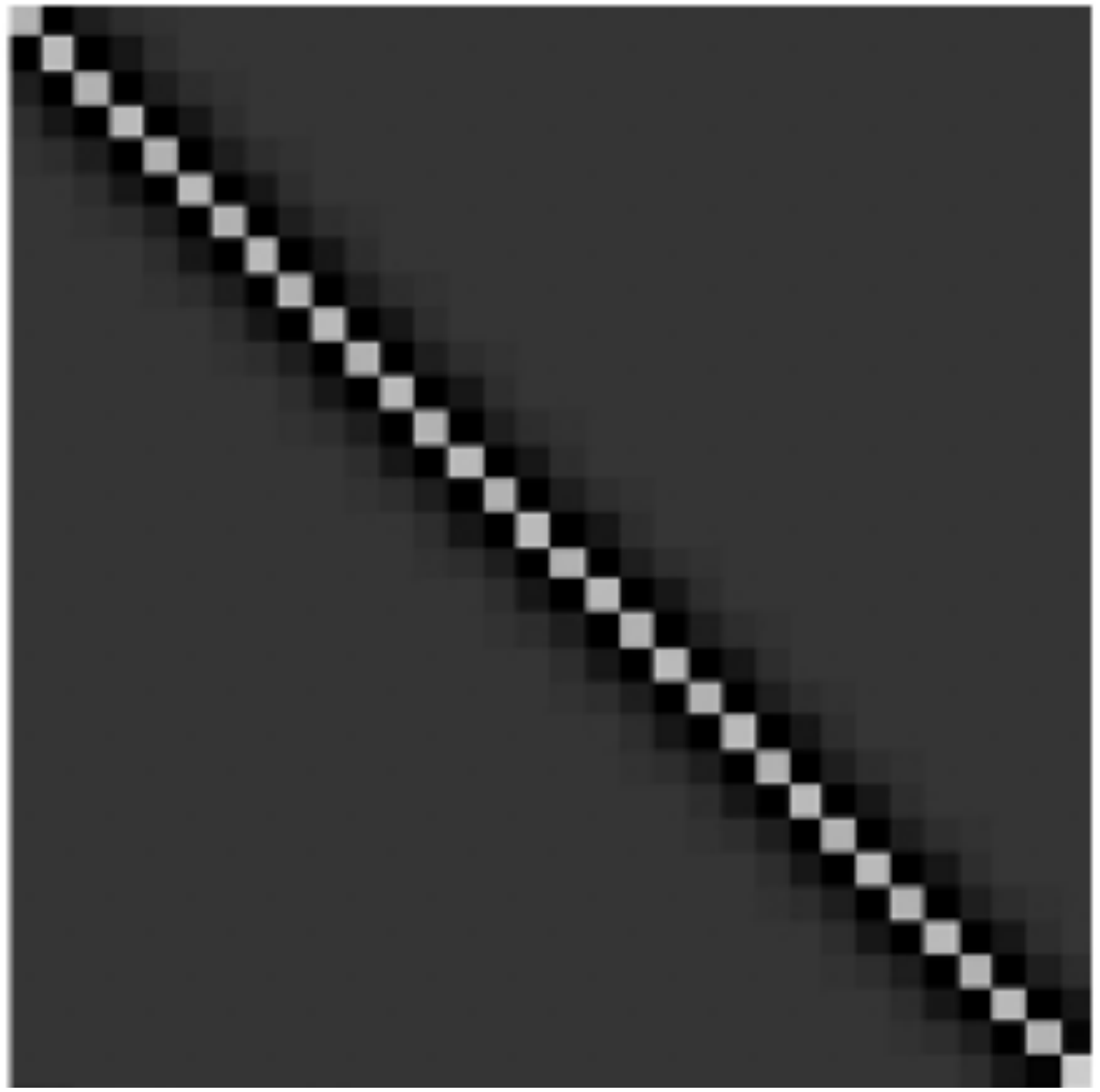
1. Find a *linear operator* such that…
    1. …it *commutes* with the group action. This means that its

Let’s see if we can **derive the Fourier Basis!!**

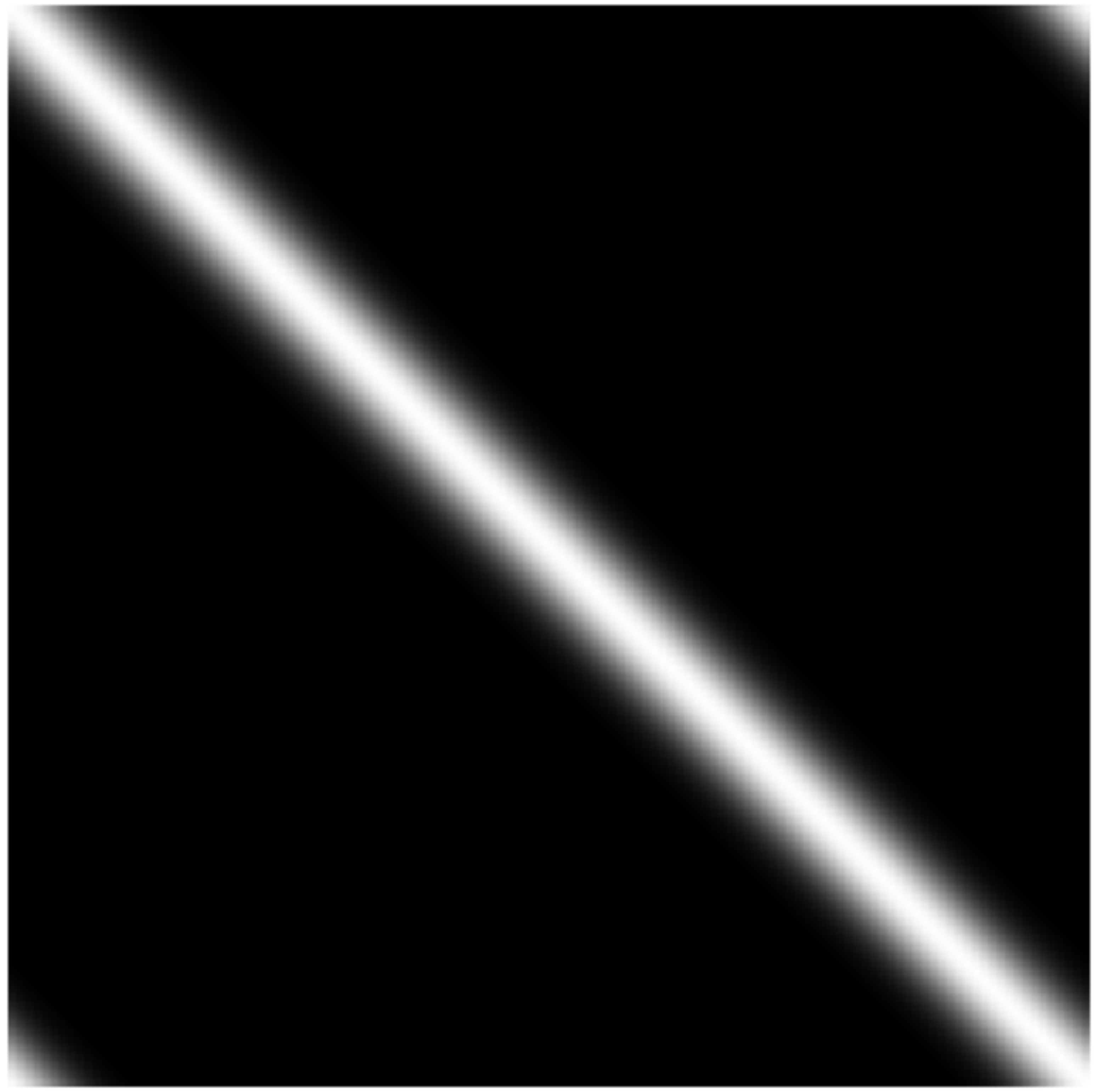
2. Compute its Eigendecomposition.
3. The eigenfunctions then **must form a steerable basis** for the group transformation!

Step 1: Find a self-adjoint operator that commutes with translations.

Laplace operator



Gaussian Blur



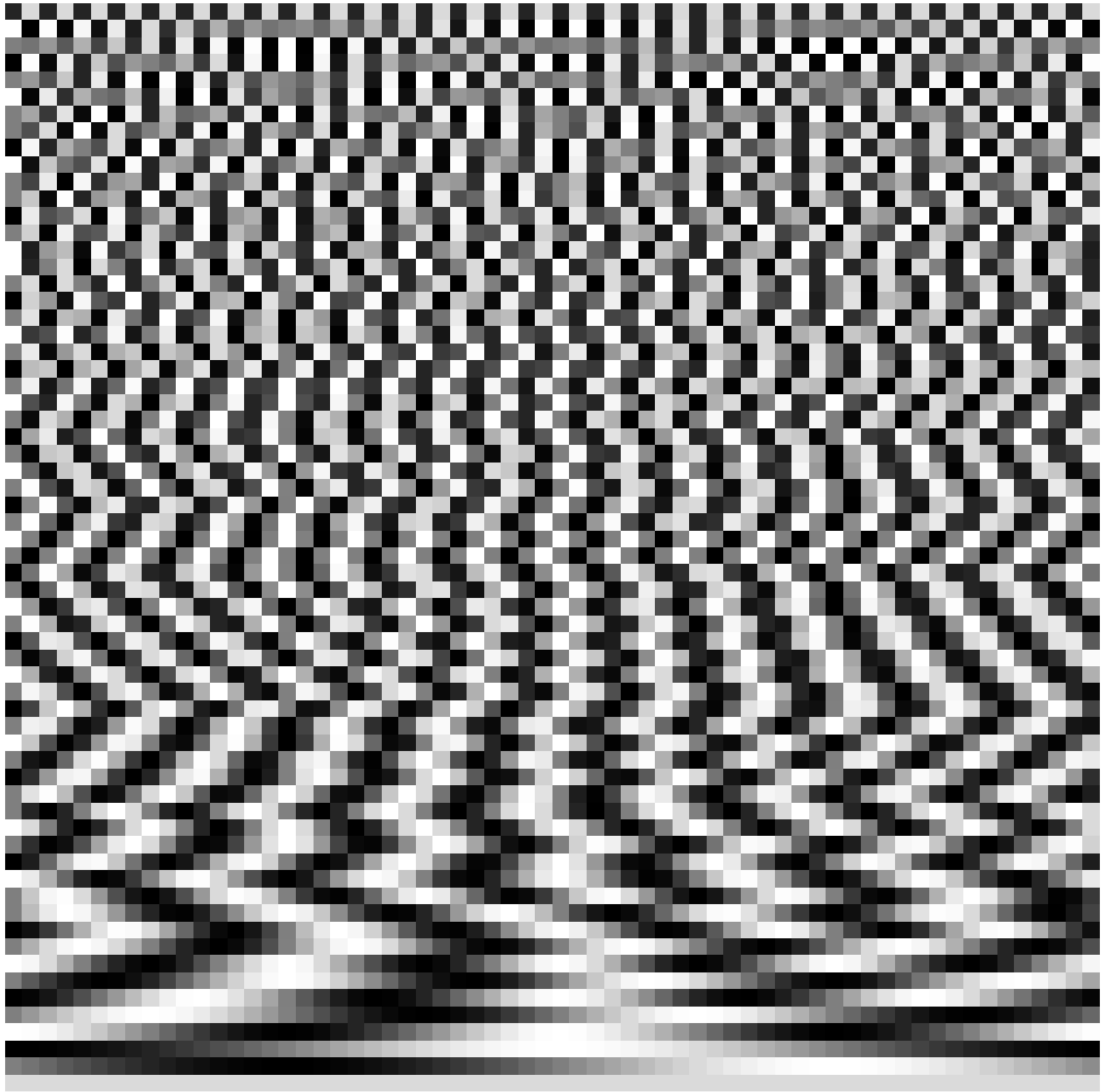
• • •

Any convolution with a centered, symmetric kernel will do!

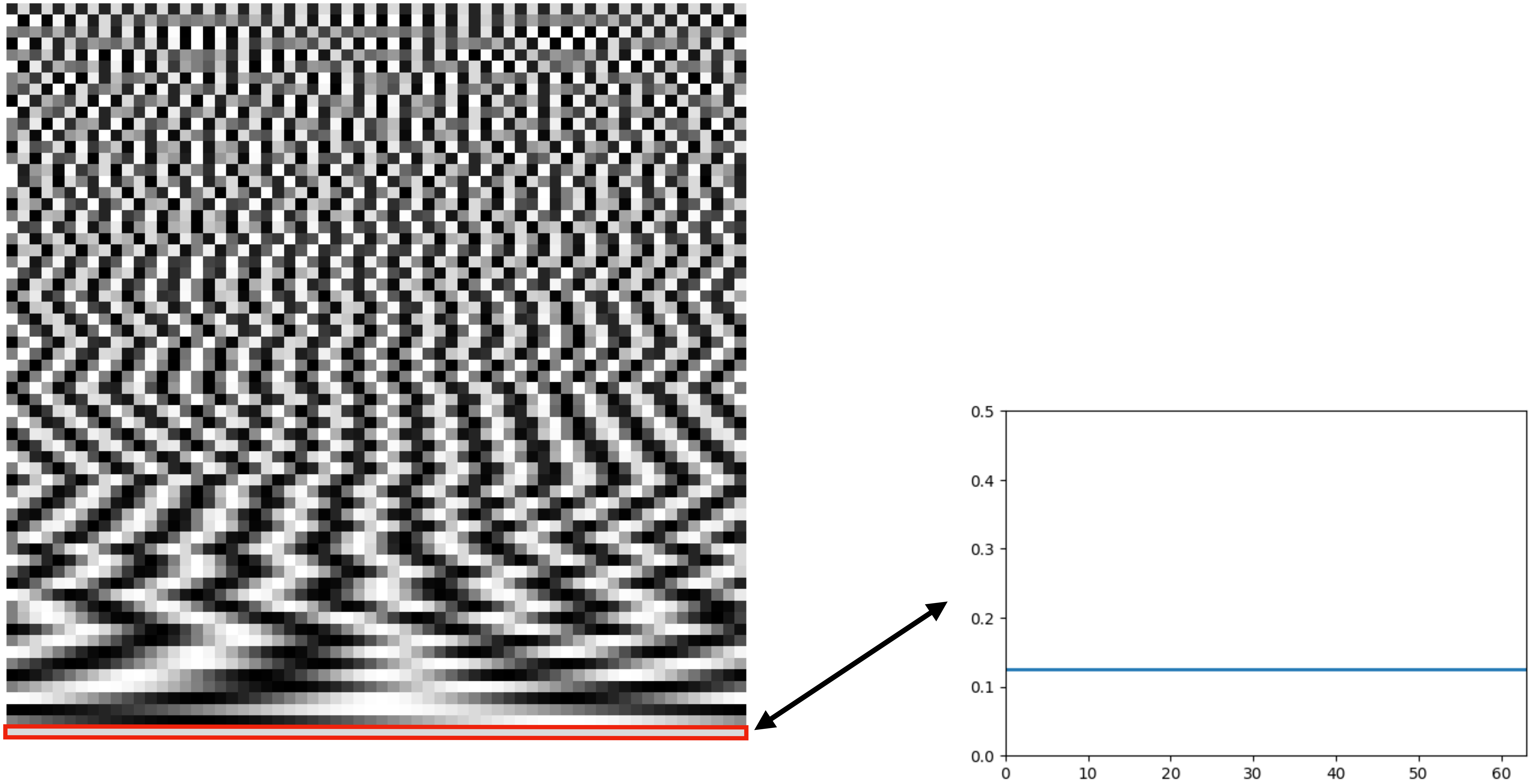
Step 2: Compute its Eigenspectrum

```
import numpy as np

n = 128

# Create Toeplitz matrix with periodic BCs
main_diag = -2 * np.ones(n)
off_diag = np.ones(n-1)

L = np.diag(main_diag) + np.diag(off_diag, 1) + np.diag(off_diag, -1)
L[0, -1] = 1
L[-1, 0] = 1

# Compute eigenvalues and eigenvectors
eigenvalues, eigenvectors = np.linalg.eigh(L)
```

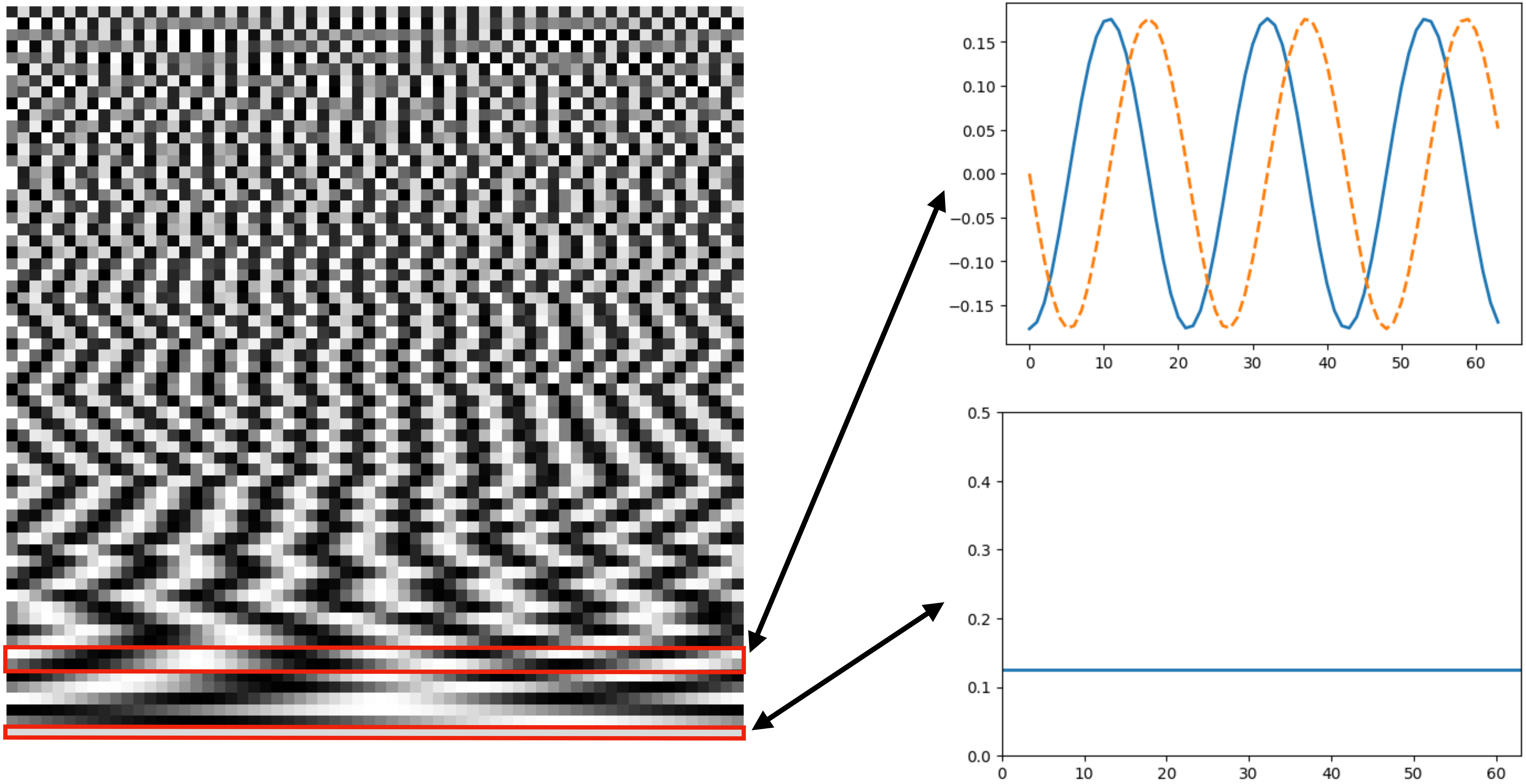
Eigenfunctions



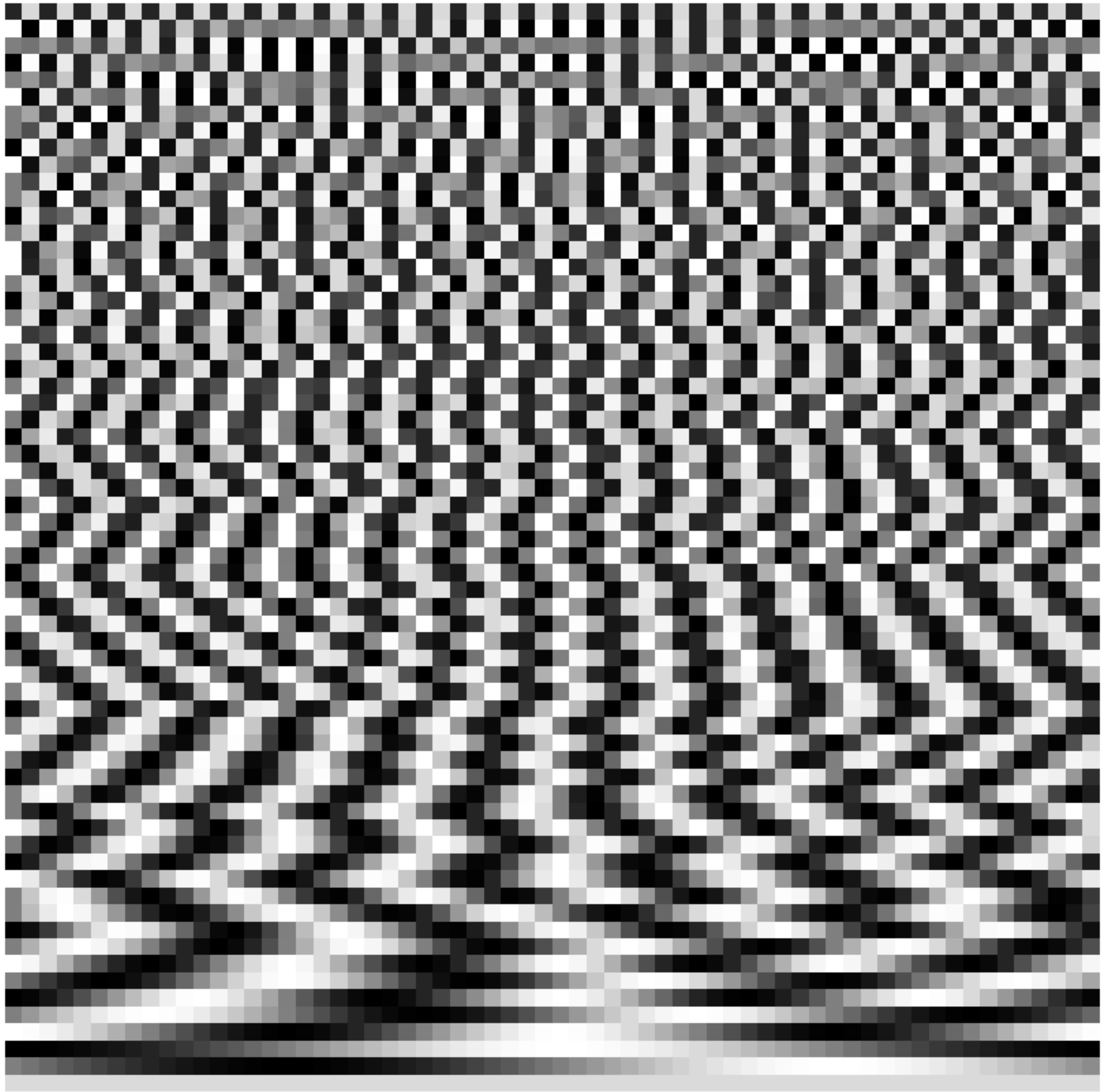
Eigenfunctions



Eigenfunctions

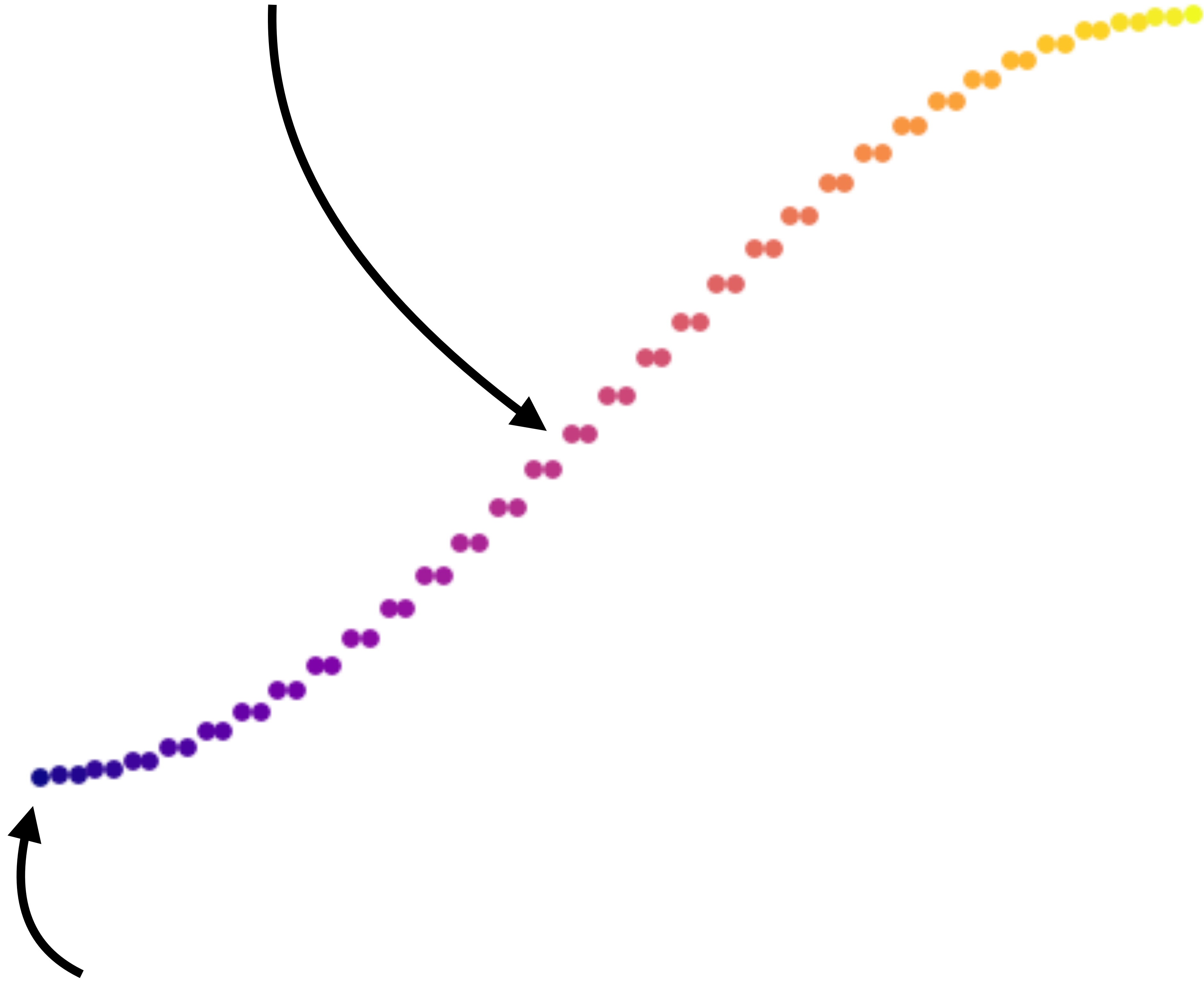


Eigenfunctions



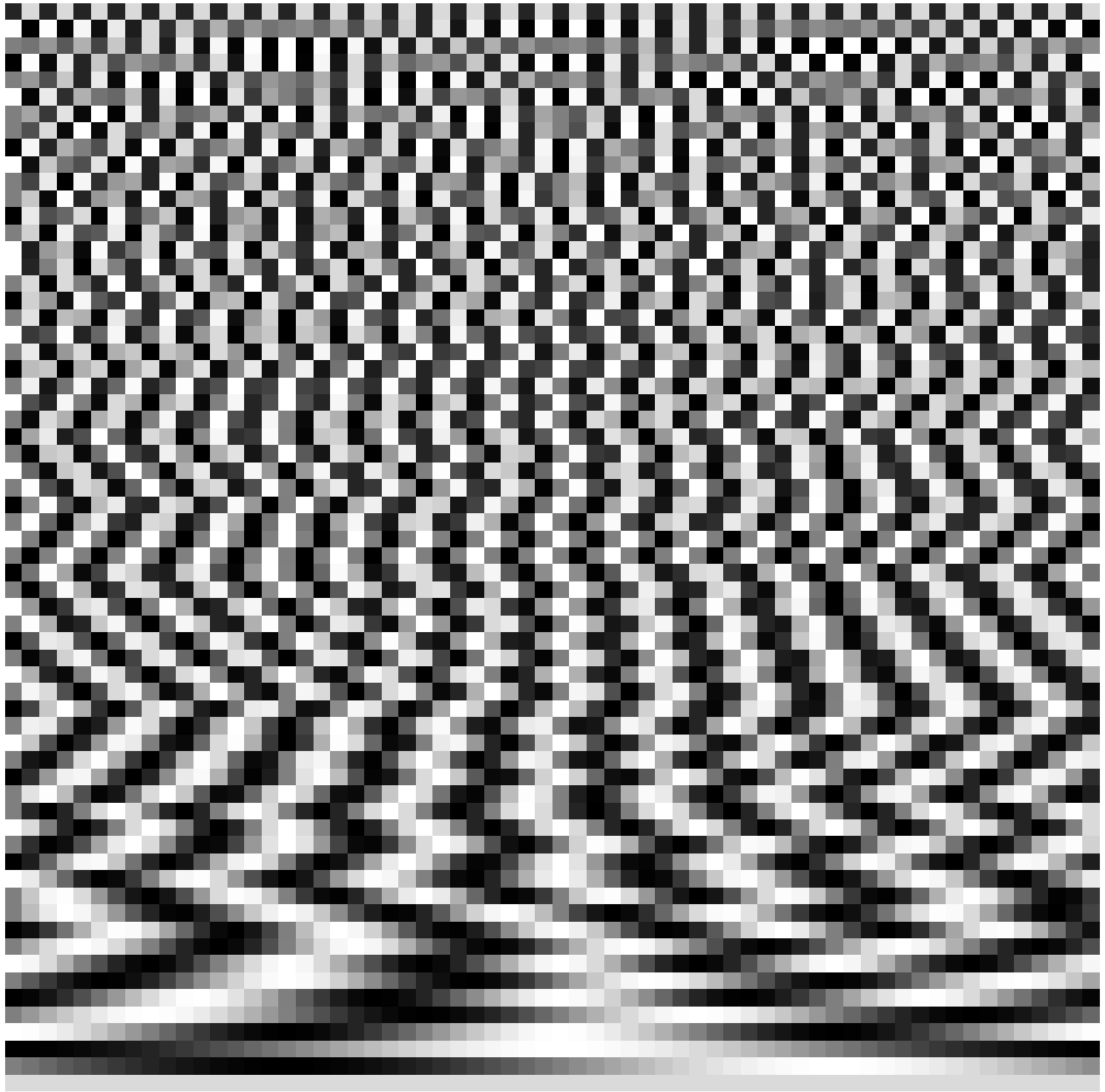
Eigenvalues

Then, pairs of equal eigenvalues

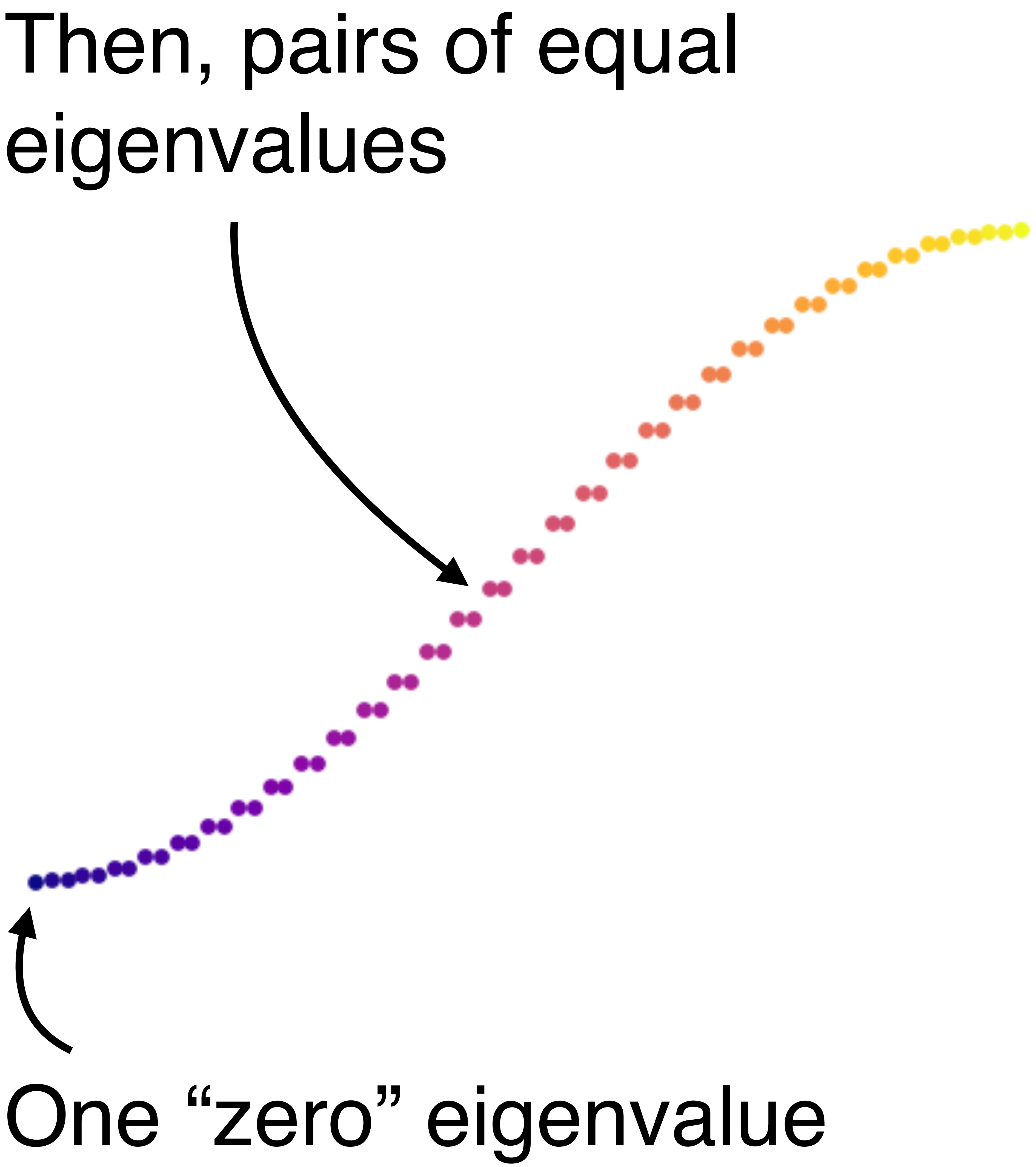


One “zero” eigenvalue

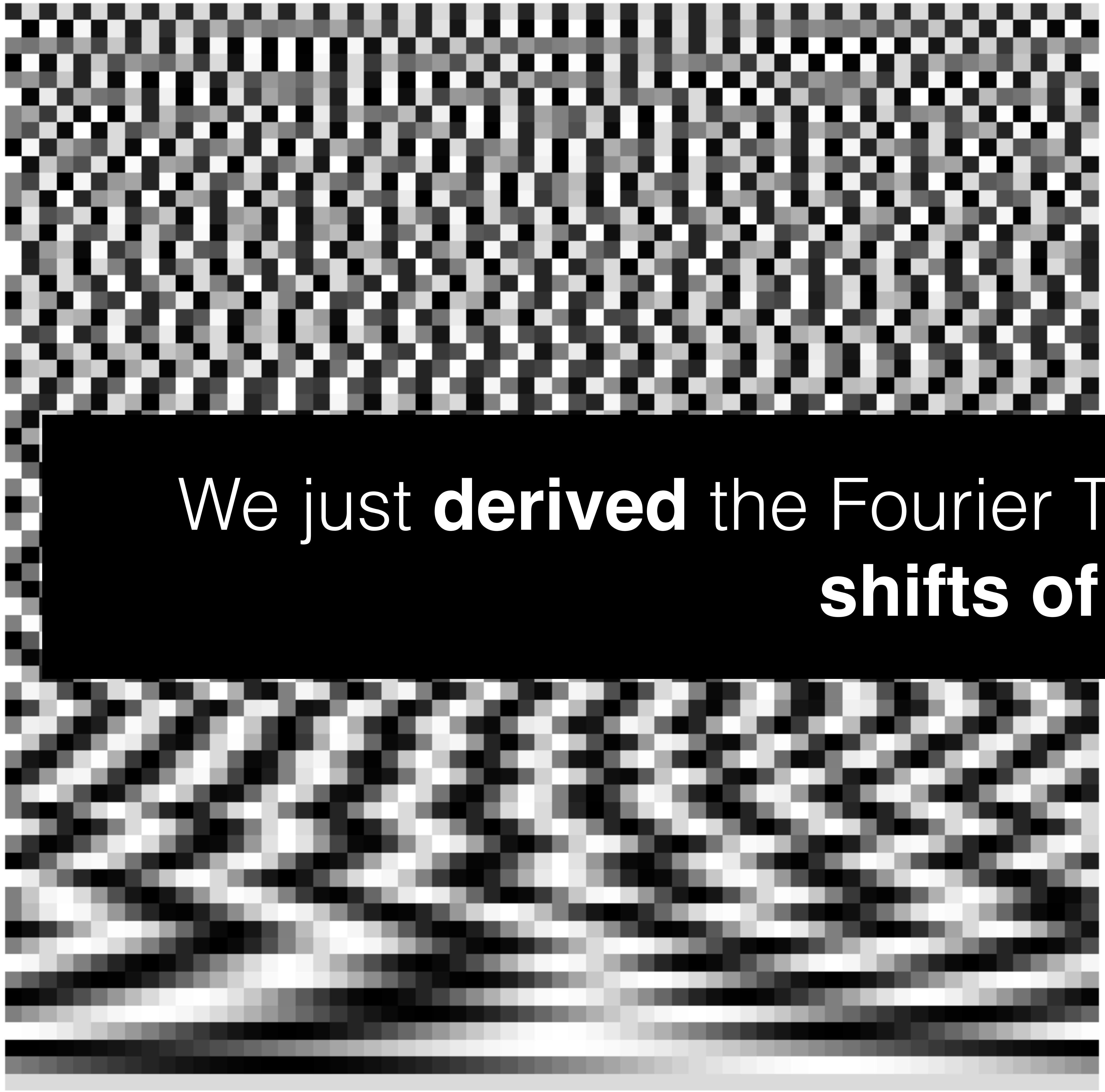
Eigenfunctions



Eigenvalues

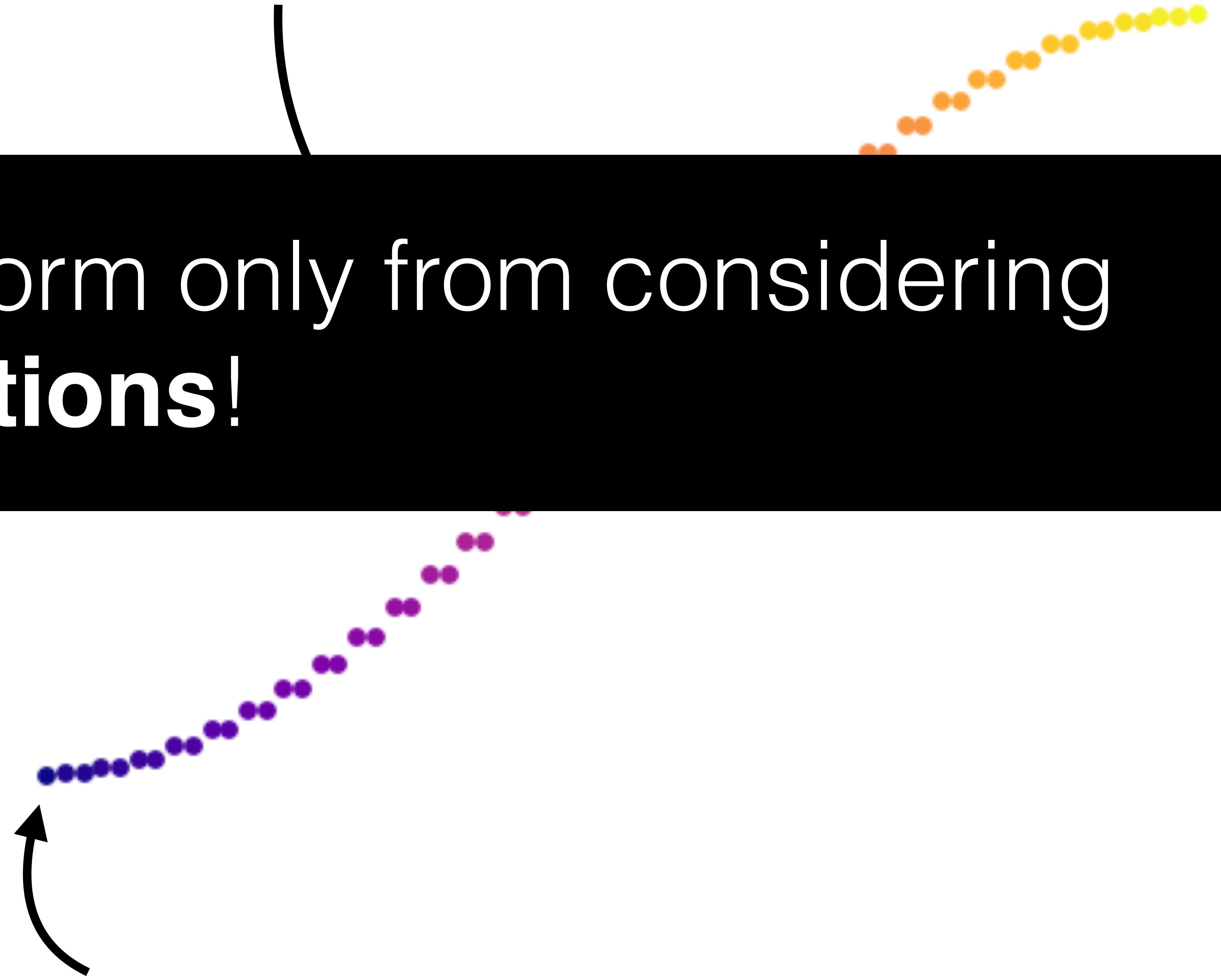


Eigenfunctions



Eigenvalues

Then, pairs of equal eigenvalues

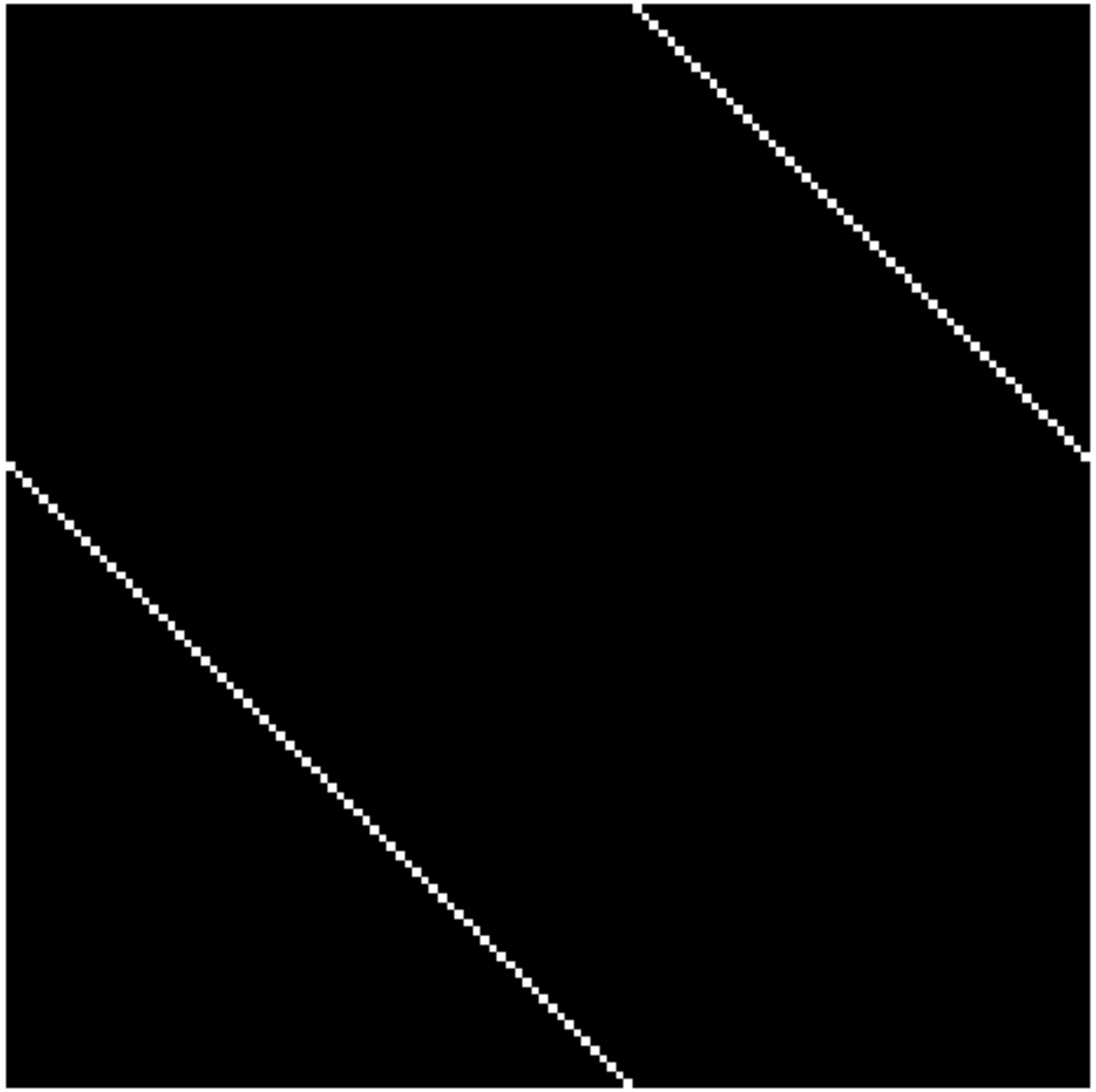


One “zero” eigenvalue

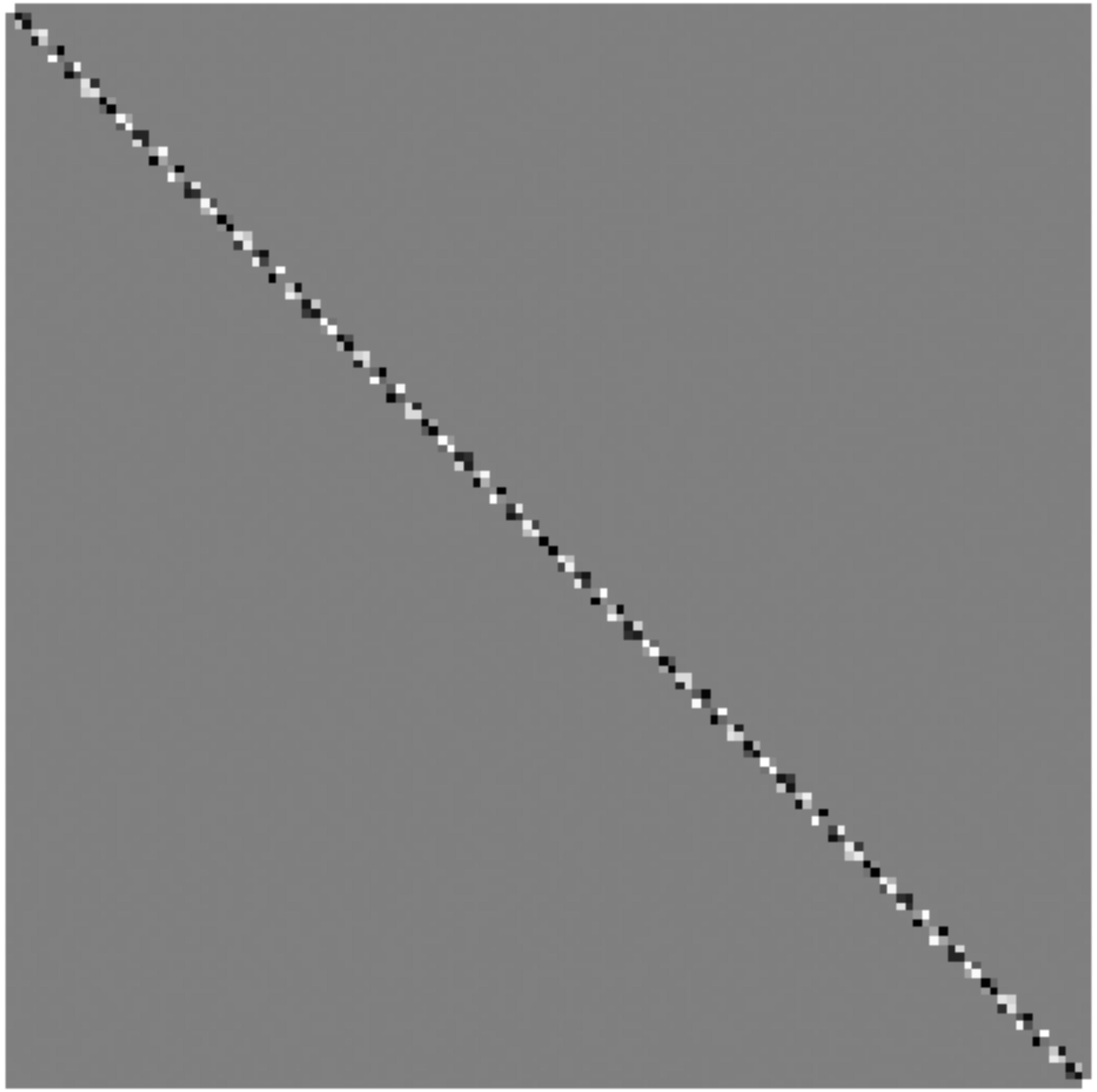
We just **derived** the Fourier Transform only from considering **shifts of functions**!

Recall: Circular Shift in Fourier Basis

$\mathbf{T}$

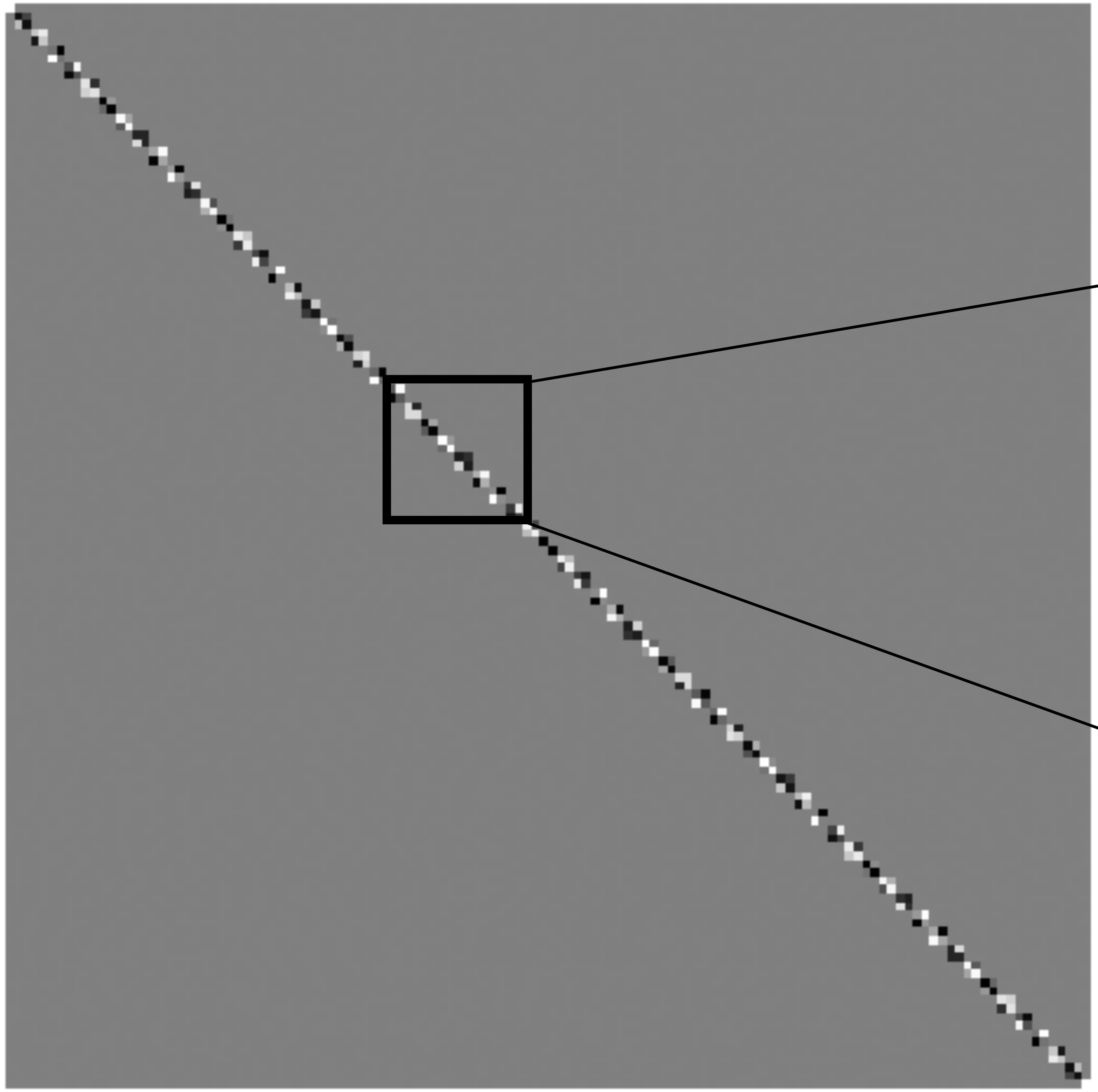


$\mathscr{F}\mathbf{T}\mathscr{F}^{-1}$



Recall: Circular Shifts in Fourier Basis

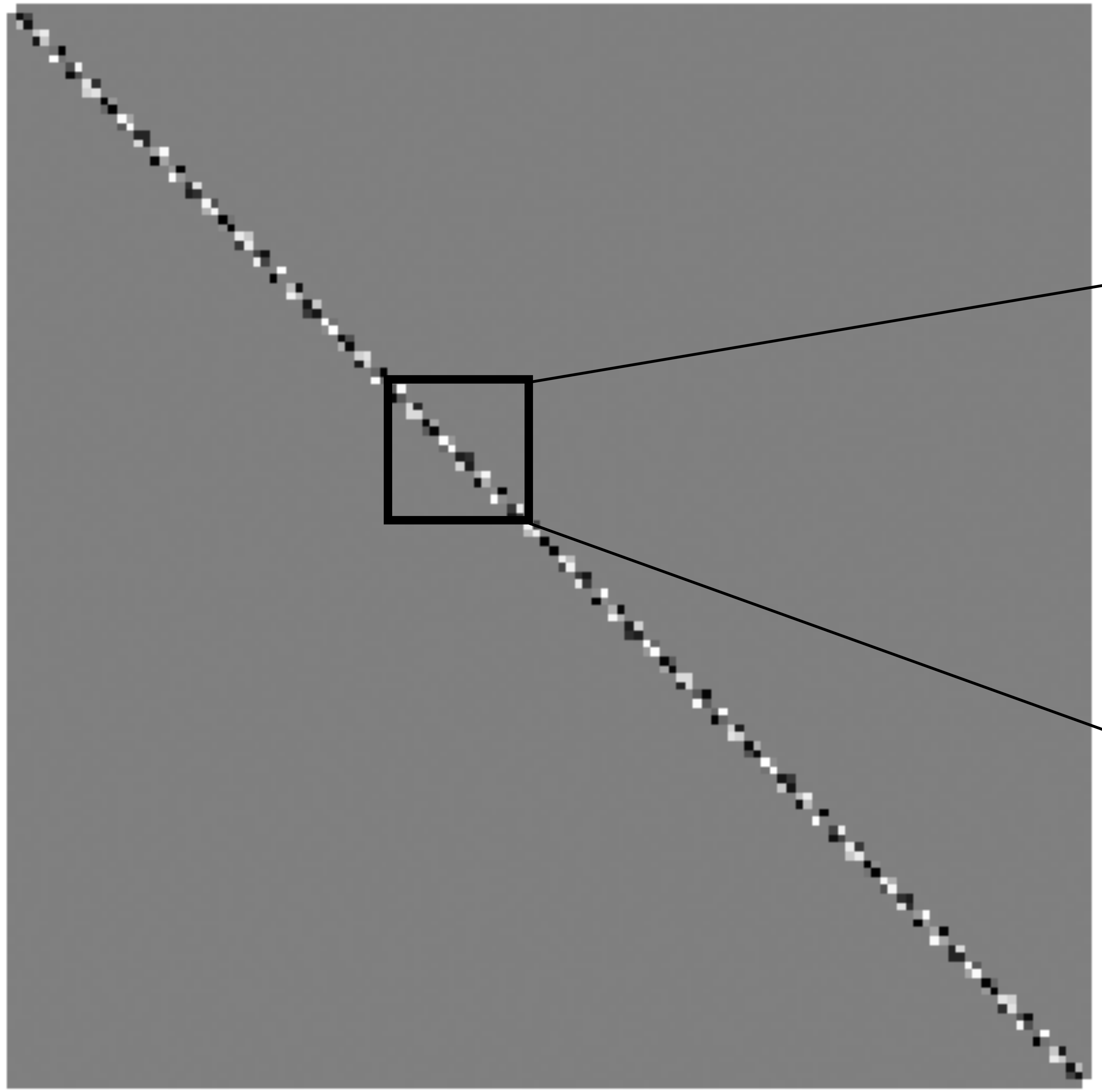
$$\mathscr{F}\mathbf{T}\mathscr{F}^{-1}$$



$$\begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \cos\theta_2 & -\sin\theta_2 & 0 & 0 \\ 0 & 0 & \sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos\theta_3 & -\sin\theta_3 \\ 0 & 0 & 0 & 0 & \sin\theta_3 & \cos\theta_3 \end{bmatrix}$$

Recall: Circular Shifts in Fourier Basis

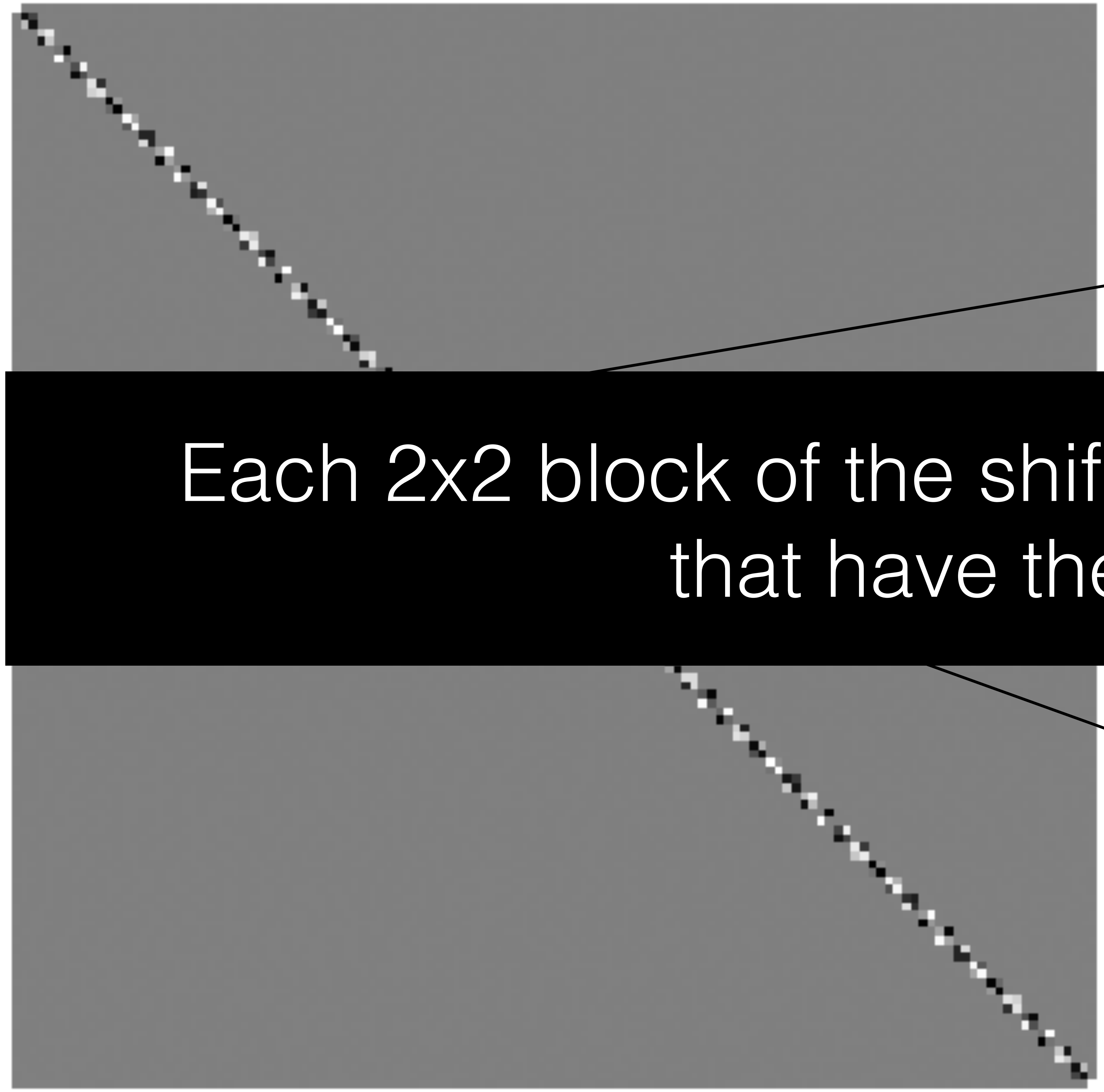
$$\mathscr{F}\mathbf{T}\mathscr{F}^{-1}$$



$$\begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \cos\theta_2 & -\sin\theta_2 & 0 & 0 \\ 0 & 0 & \sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos\theta_3 & -\sin\theta_3 \\ 0 & 0 & 0 & 0 & \sin\theta_3 & \cos\theta_3 \end{bmatrix}$$

Recall: Circular Shifts in Fourier Basis

$$\mathscr{F}\mathbf{T}\mathscr{F}^{-1}$$



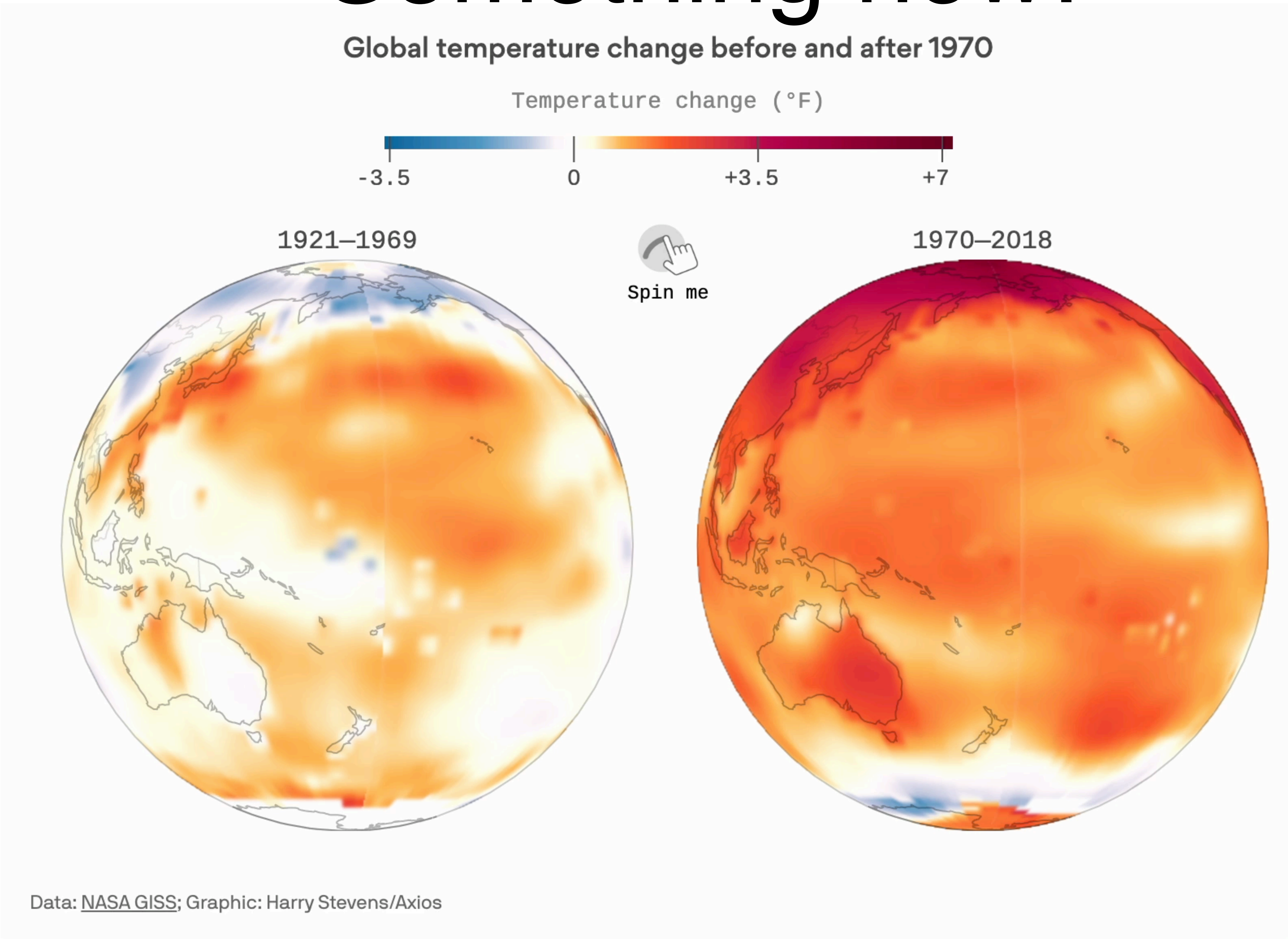
$$\begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sin\theta_3 & \cos\theta_3 \end{bmatrix}$$

Each 2x2 block of the shift operator only mixes coefficients that have the same eigenvalue!

Something new!

We will use our new-found knowledge to derive a steerable basis that you may not yet be familiar with, but which has broad applications in vision, graphics, and beyond.

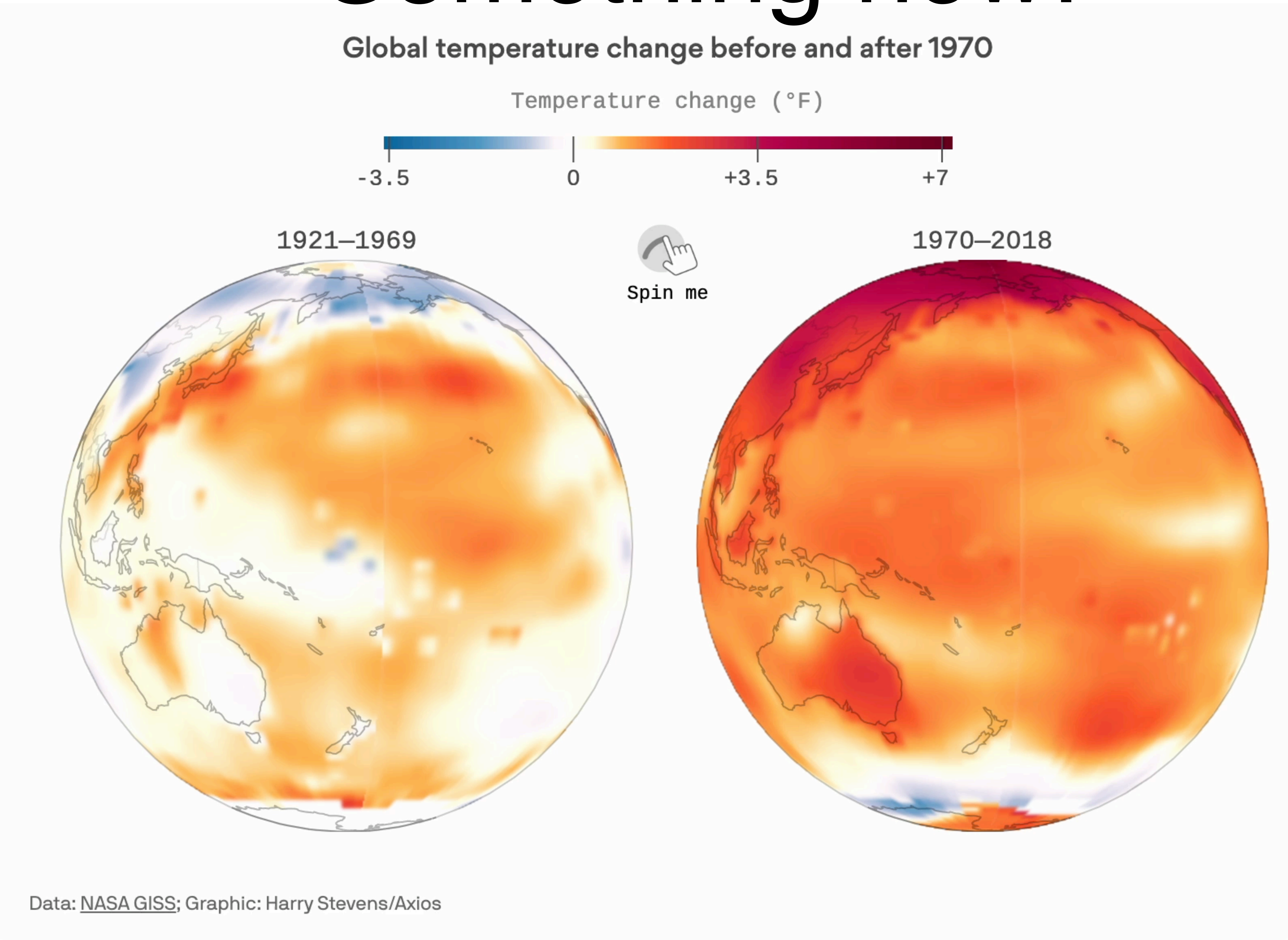
Something new!



Source: https://www.axios.com/2019/05/25/climate-change-earth-temperature-change

We will use our new-found knowledge to derive a steerable basis that you may not yet be familiar with, but which has broad applications in vision, graphics, and beyond.

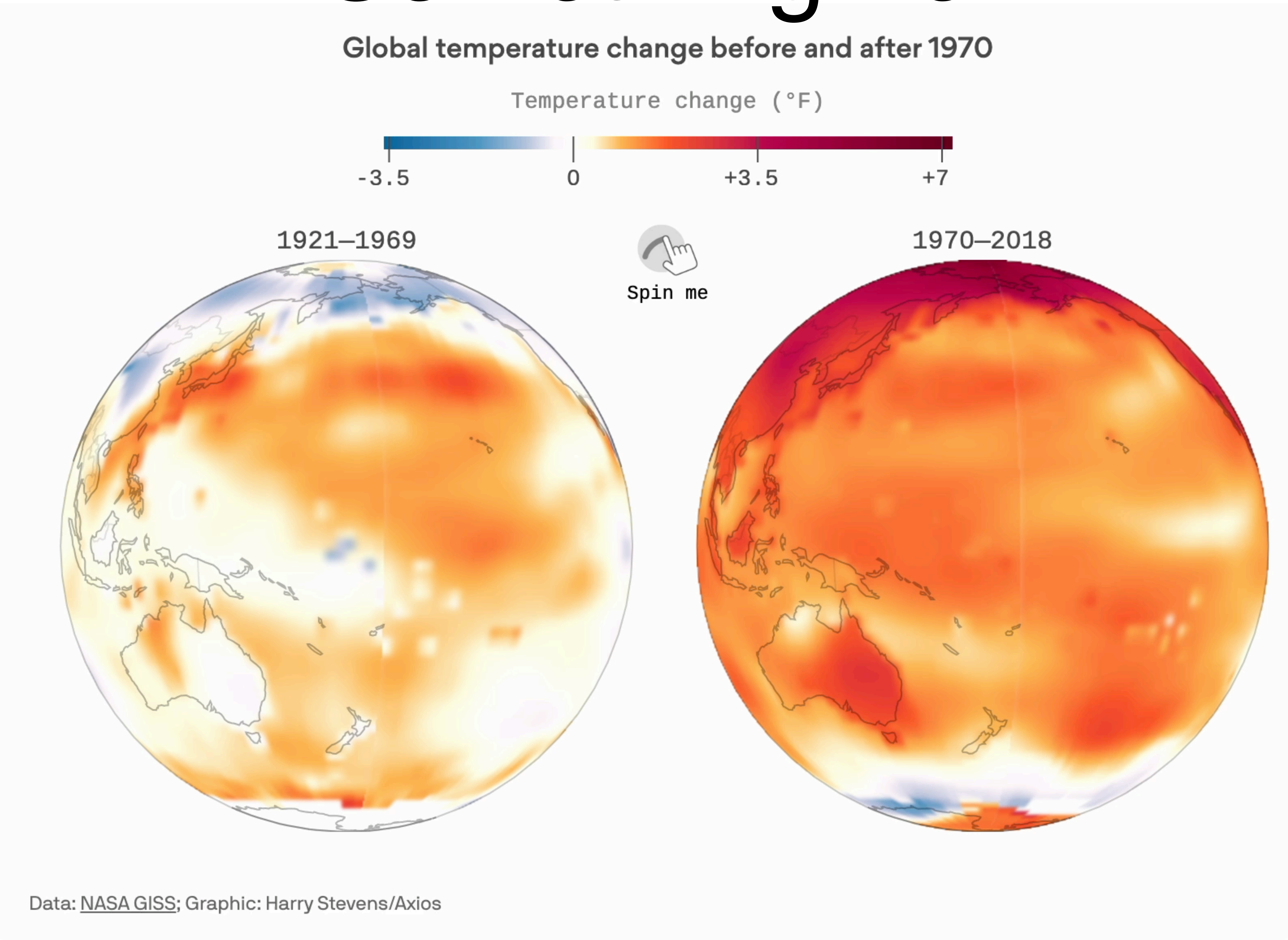
Something new!



Source: https://www.axios.com/2019/05/25/climate-change-earth-temperature-change

We will use our new-found knowledge to derive a steerable basis that you may not yet be familiar with, but which has broad applications in vision, graphics, and beyond.

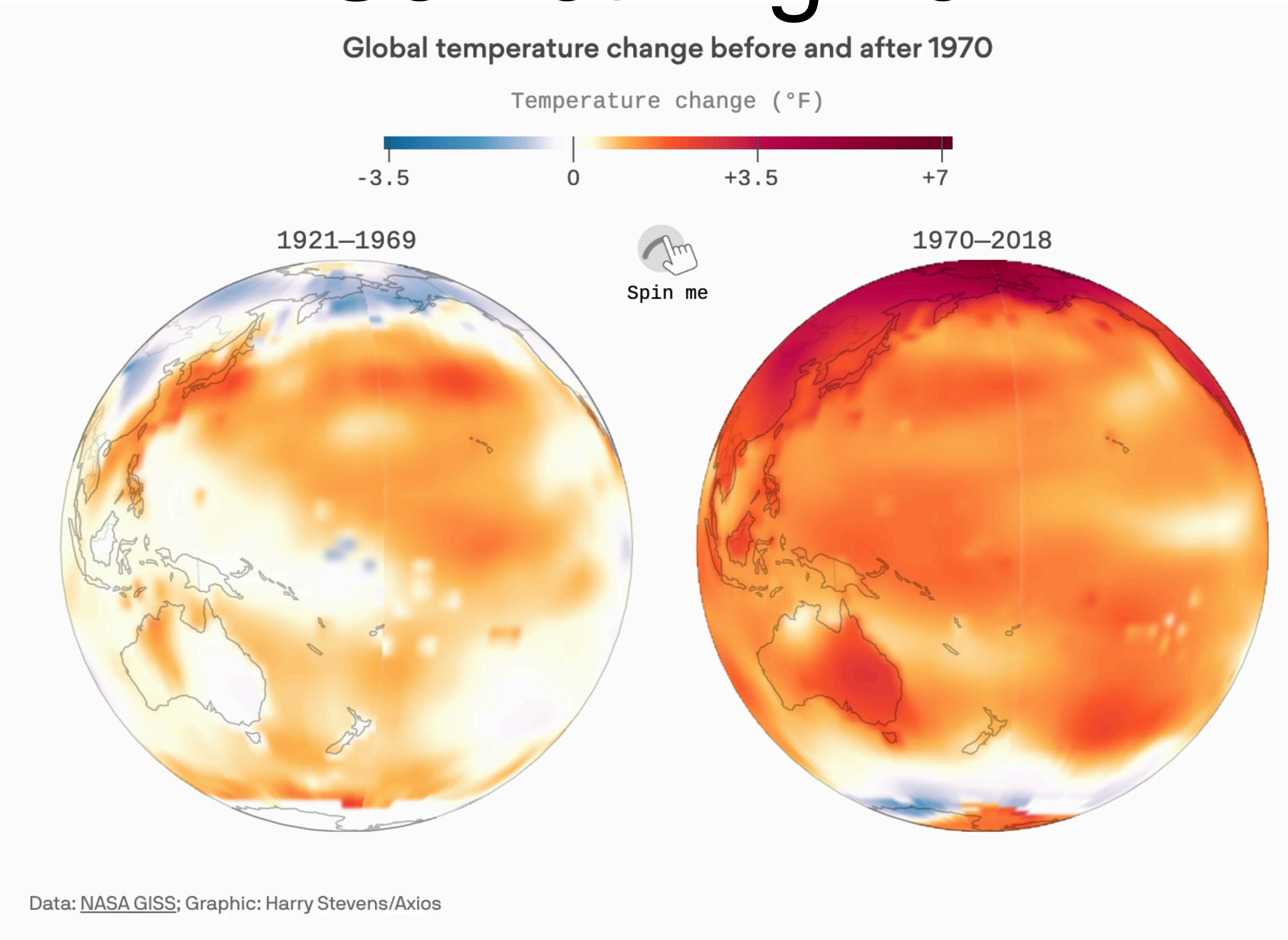
Something new!



Source: https://www.axios.com/2019/05/25/climate-change-earth-temperature-change

Consider functions on the sphere - like here, the texture of the sphere (the map) or the temperature at every point on the sphere.

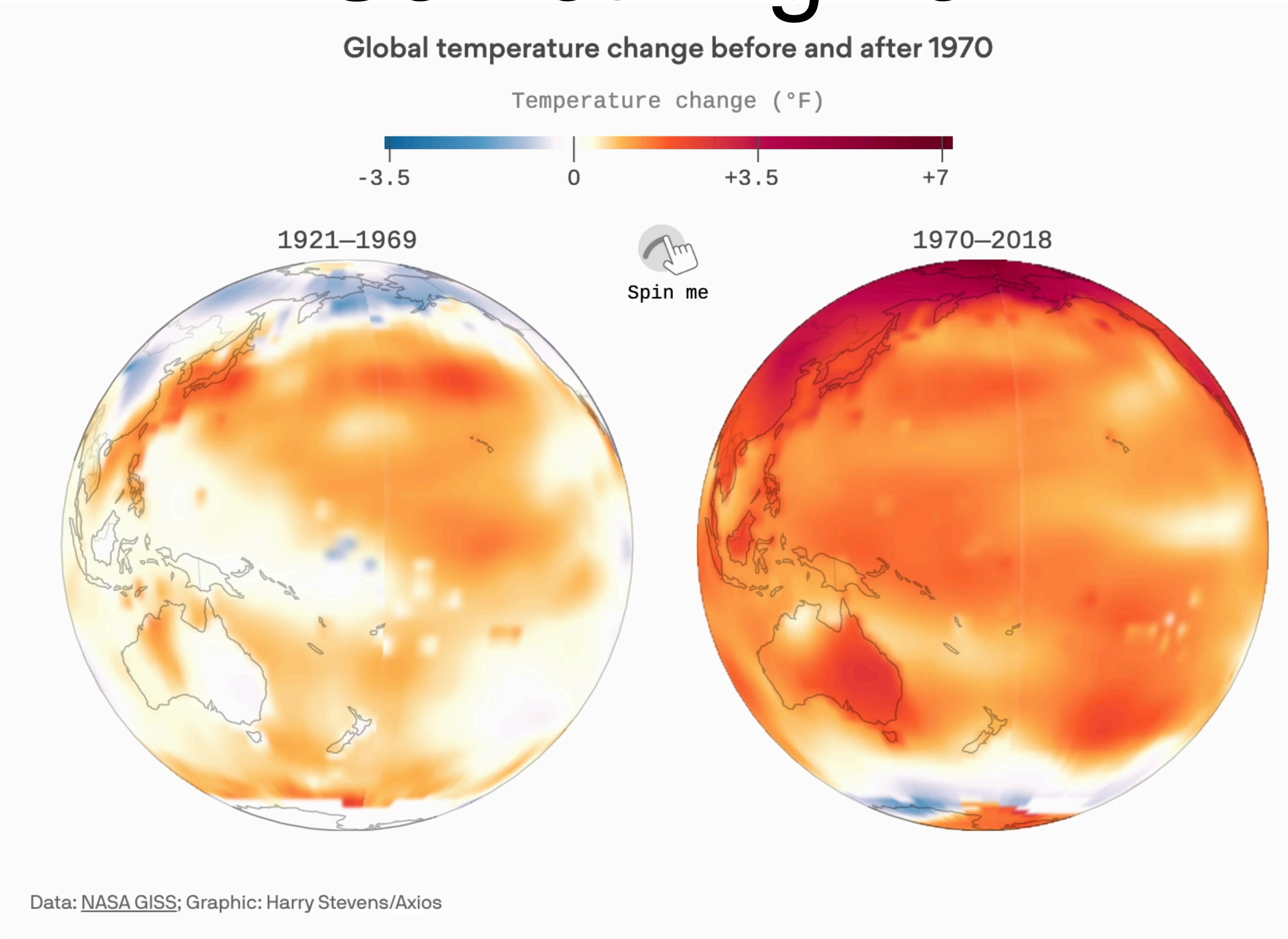
Something new!



Source: https://www.axios.com/2019/05/25/climate-change-earth-temperature-change

Consider functions on the sphere - like here, the texture of the sphere (the map) or the temperature at every point on the sphere.

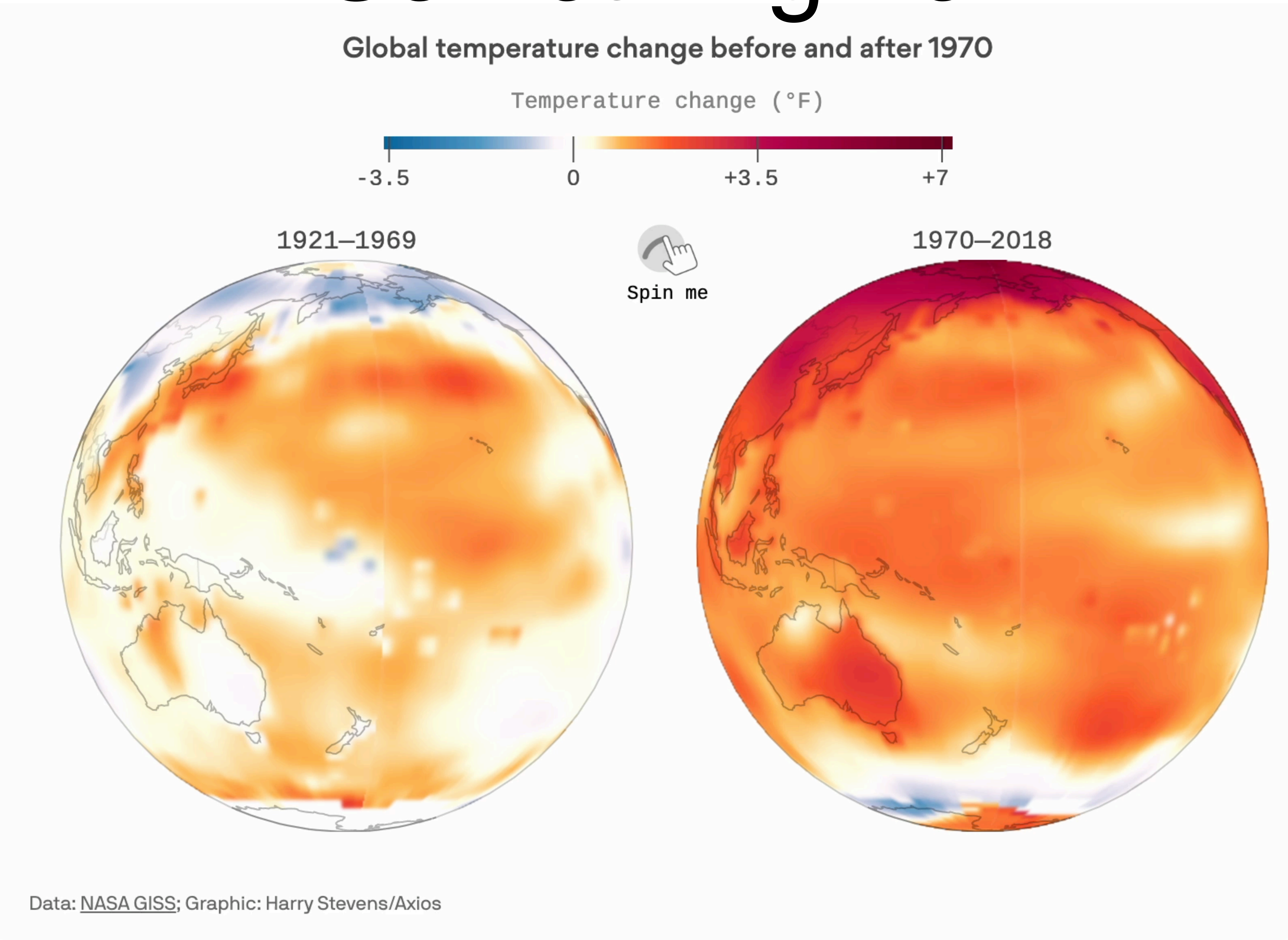
Something new!



Source: https://www.axios.com/2019/05/25/climate-change-earth-temperature-change

Is there a *rotation-steerable basis* for these functions?

Something new!



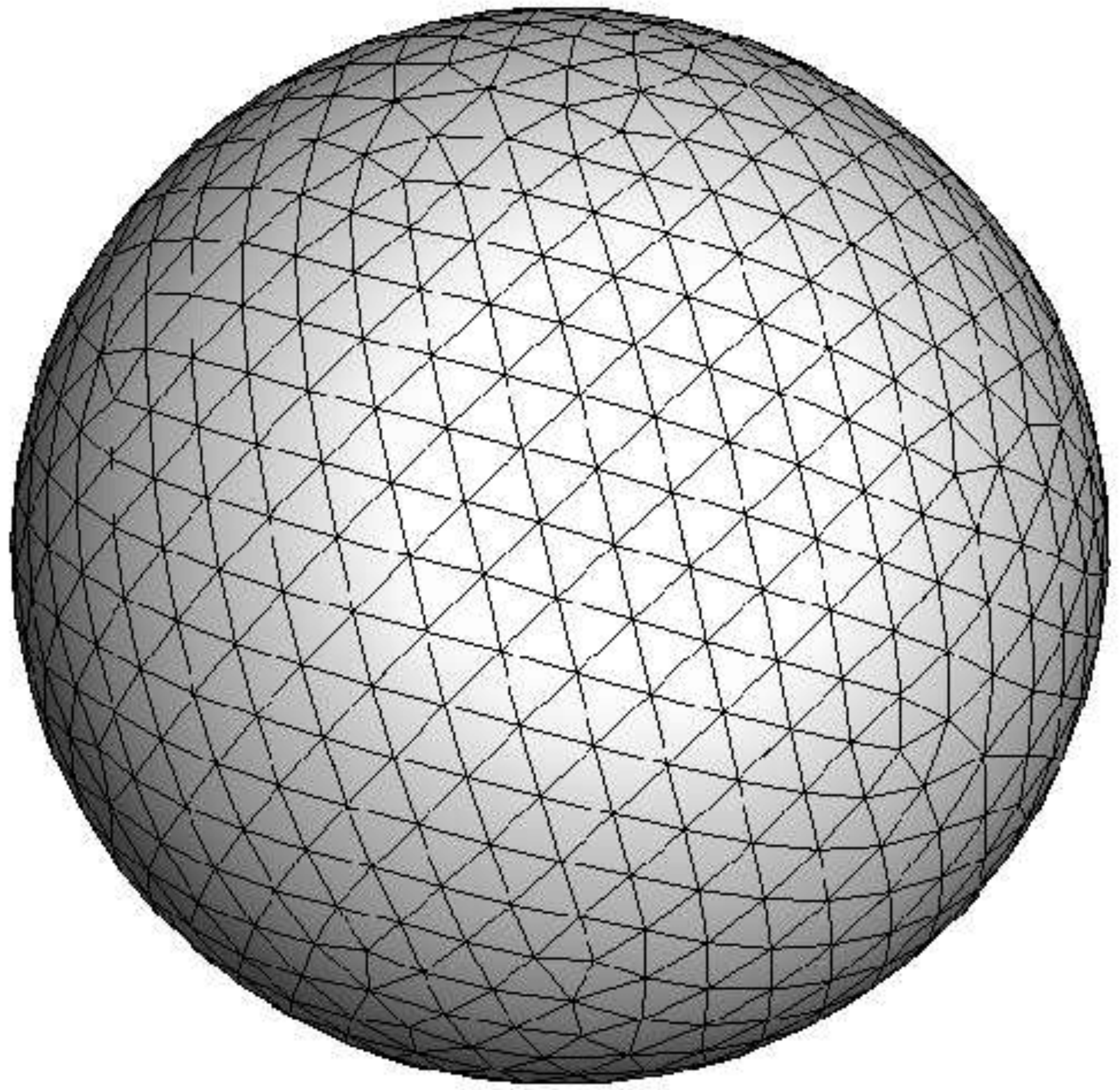
Source: https://www.axios.com/2019/05/25/climate-change-earth-temperature-change

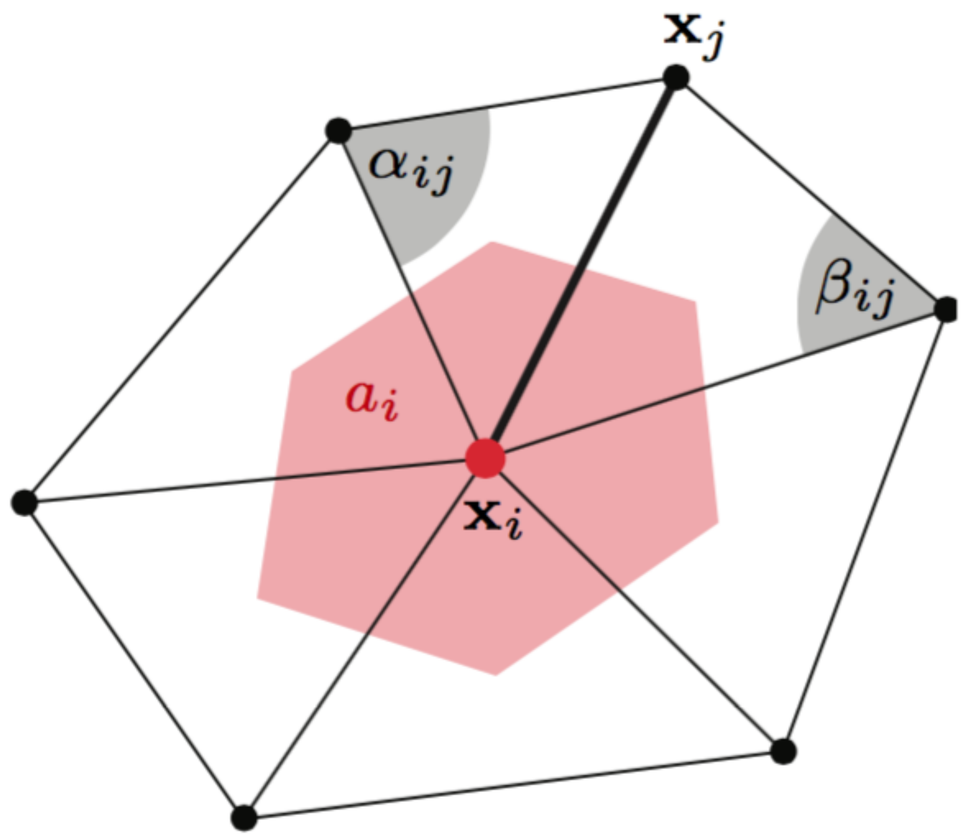
Is there a *rotation-steerable basis* for these functions?

Recipe for finding a steerable basis

1. Find a *linear operator* such that…
    1. …it *commutes* with the group action. This means that its eigenspace is unchanged under the group action.
    2. …its eigenvectors are guaranteed to be a complete basis for the function space (it is self-adjoint).
2. Compute its Eigendecomposition.
3. The eigenfunctions then form a *steerable basis* for the group transformation!

Cutting to the Chase: The Laplace-Beltrami Operator





$$(\Delta u)_i \approx \underbrace{\frac{1}{2\alpha_i}}_{M} \underbrace{\sum_{j \in \mathcal{N}(i)} (\cot \alpha_{ij} + \cot \beta_{ij})(\mathbf{x}_i - \mathbf{x}_j)}_{C}$$

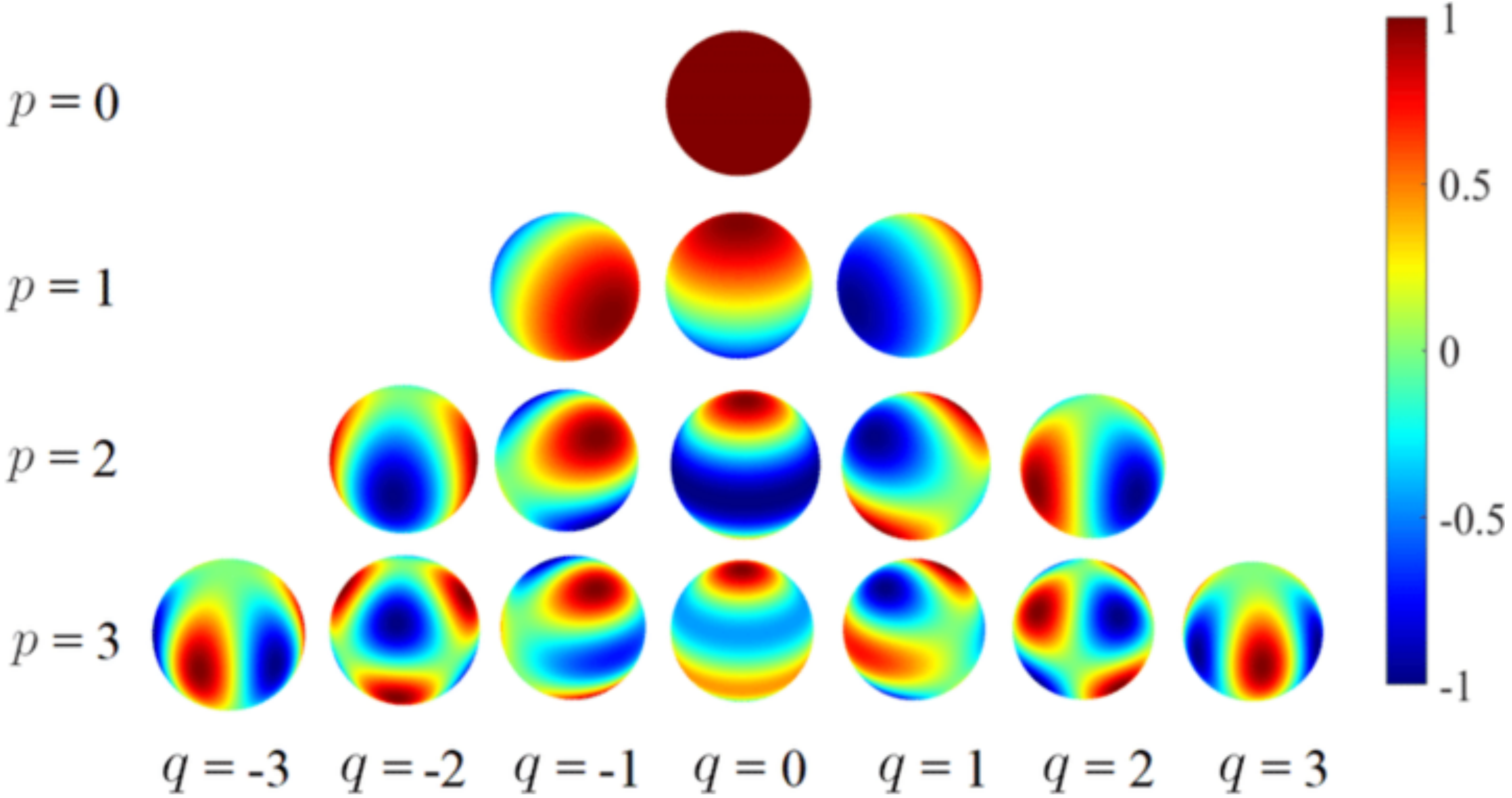
$$L = MC$$

Swiss Army Laplacian, Solomon, Crane and Vouga, https://www.cs.cmu.edu/~kmcrane/Projects/Other/SwissArmyLaplacian.pdf

user10024395 (https://math.stackexchange.com/users/139772/user10024395), How to intepret cotangent laplacian?, URL (version: 2017-11-09): https://math.stackexchange.com/q/2512014

The Spherical Harmonics

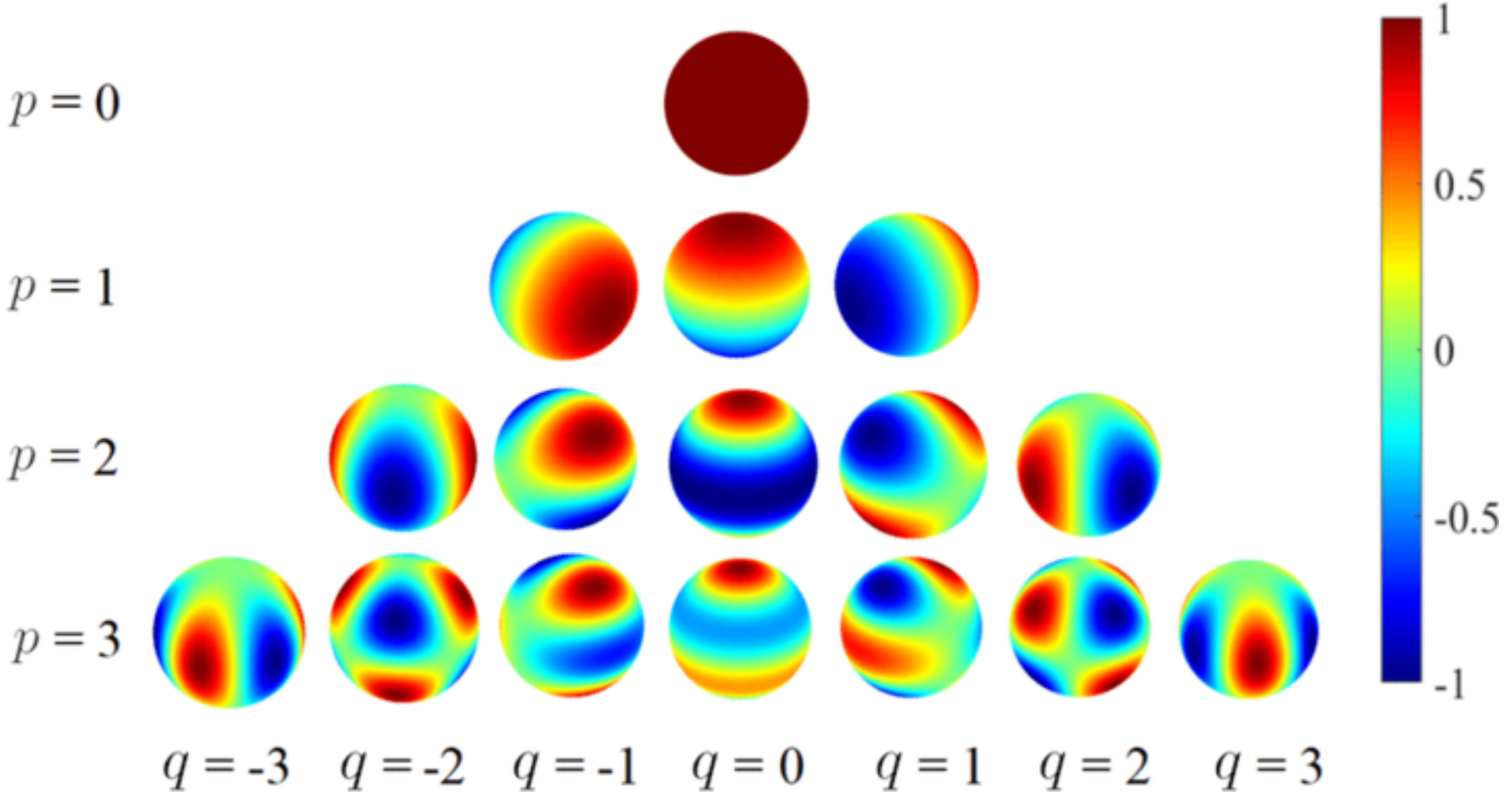
1. A Fourier Basis for functions on the sphere!
2. Are they rotation-steerable…?

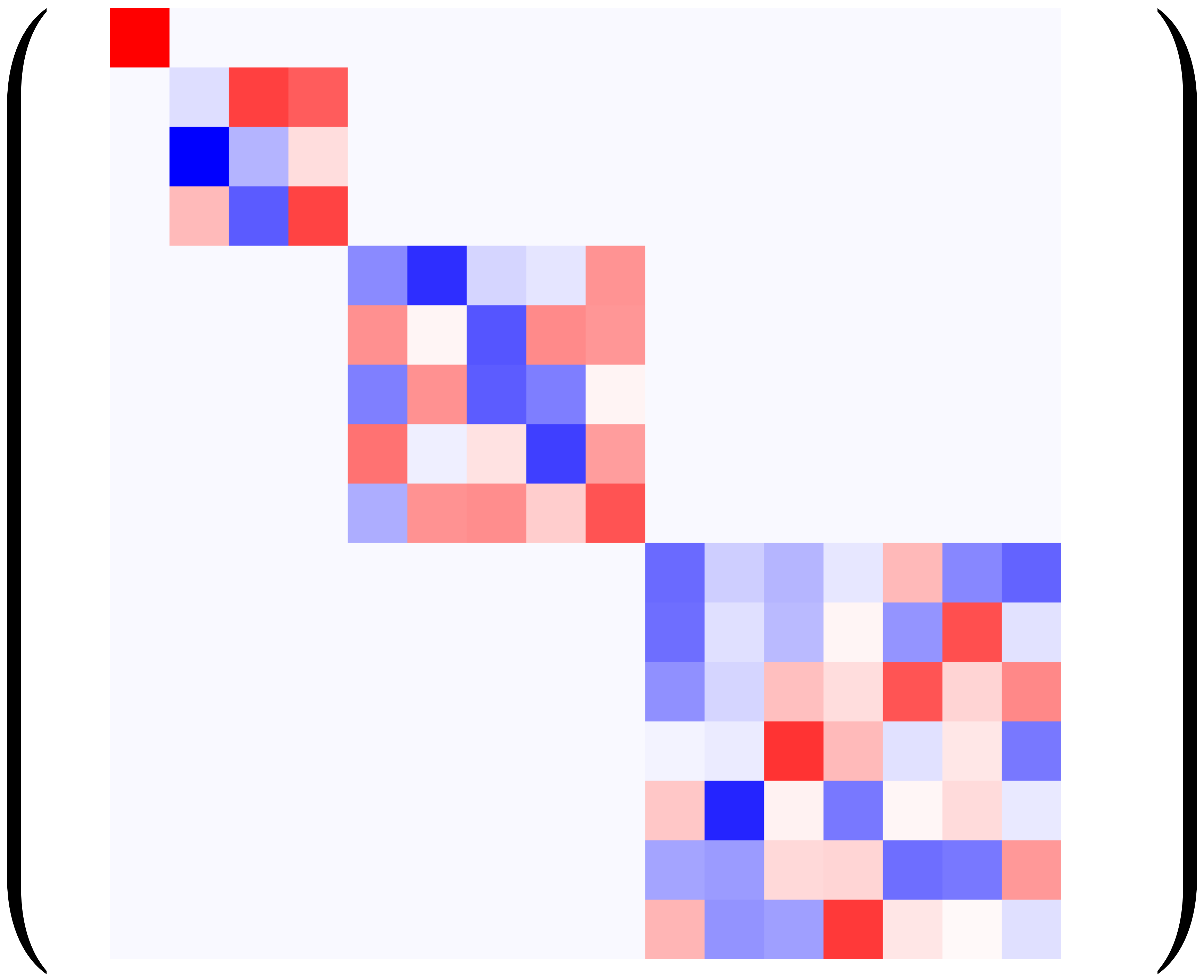


Link to Colab

Steering the spherical harmonics basis via the "Wigner-D Matrices"

$$f(\mathbf{R}\mathbf{n}) = [\mathscr{F}_{S^2}^{-1} \ \mathbf{D}(\mathbf{R}) \ \mathscr{F}_{S^2} \ f]](\mathbf{n})$$

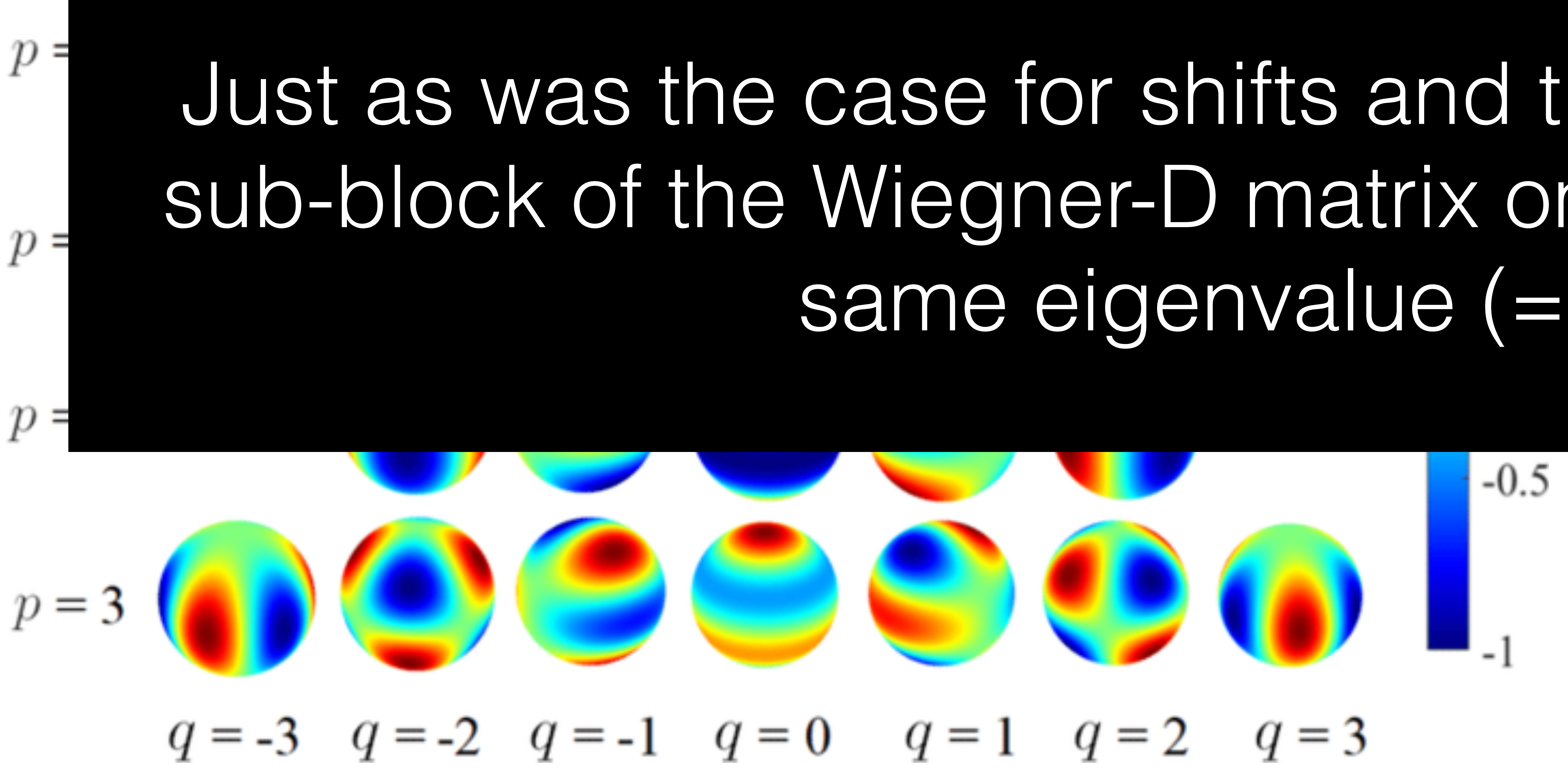


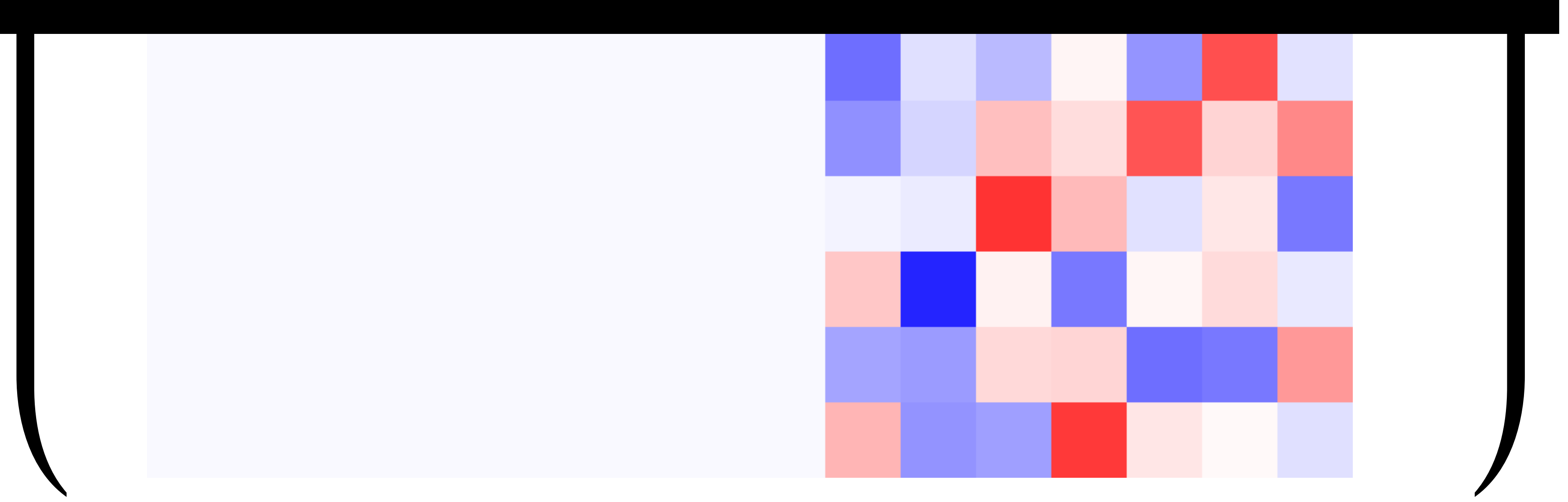


Steering the spherical harmonics basis via the "Wigner-D Matrices"

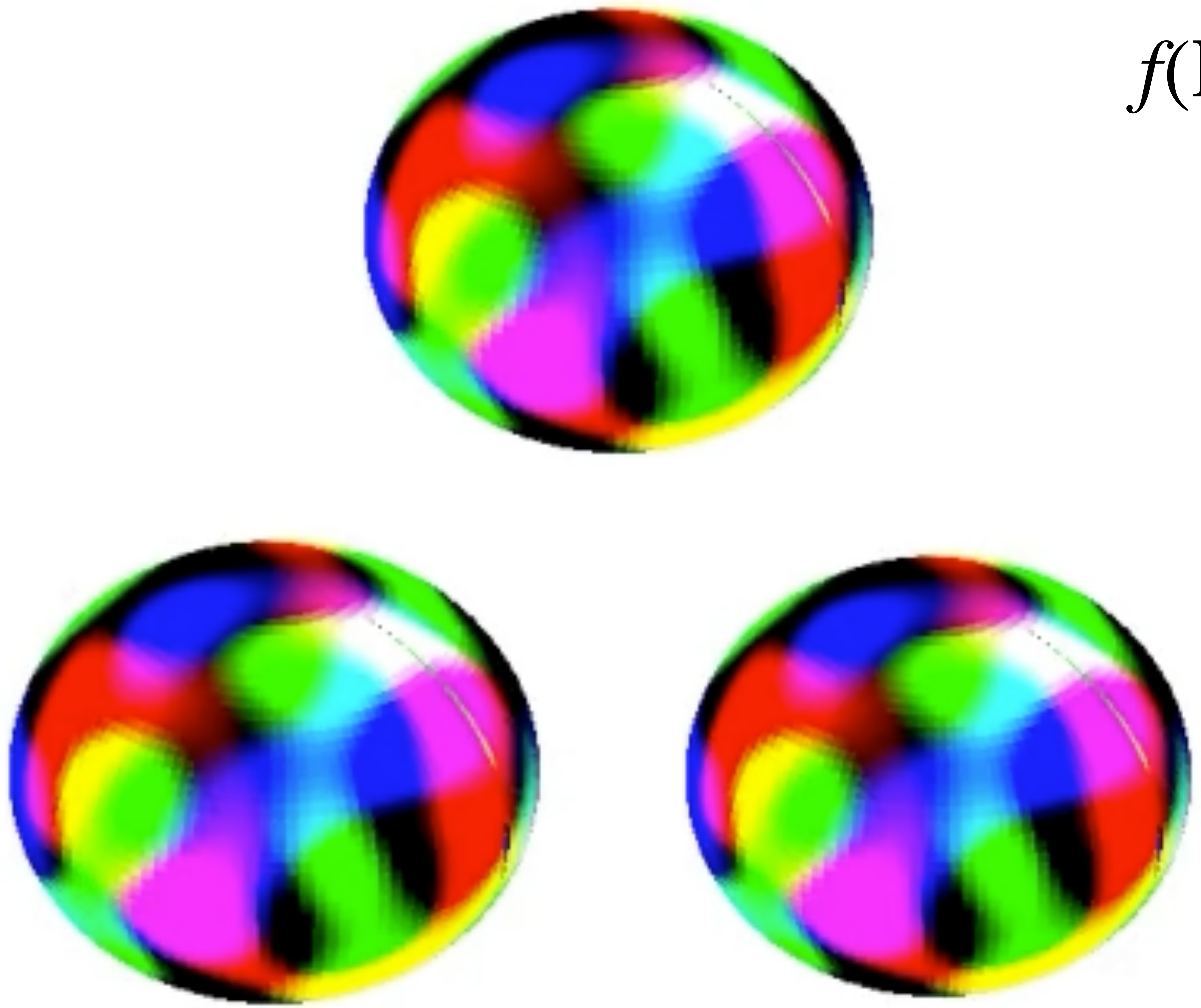
$$f(\mathbf{Rn}) = [\mathscr{F}_{S^2}^{-1}\ \mathbf{D}(\mathbf{R})\ \mathscr{F}_{S^2}\ f]](\mathbf{n})$$

Just as was the case for shifts and the Fourier Transform, each sub-block of the Wiegner-D matrix only mixes coefficients of the same eigenvalue (=frequency)

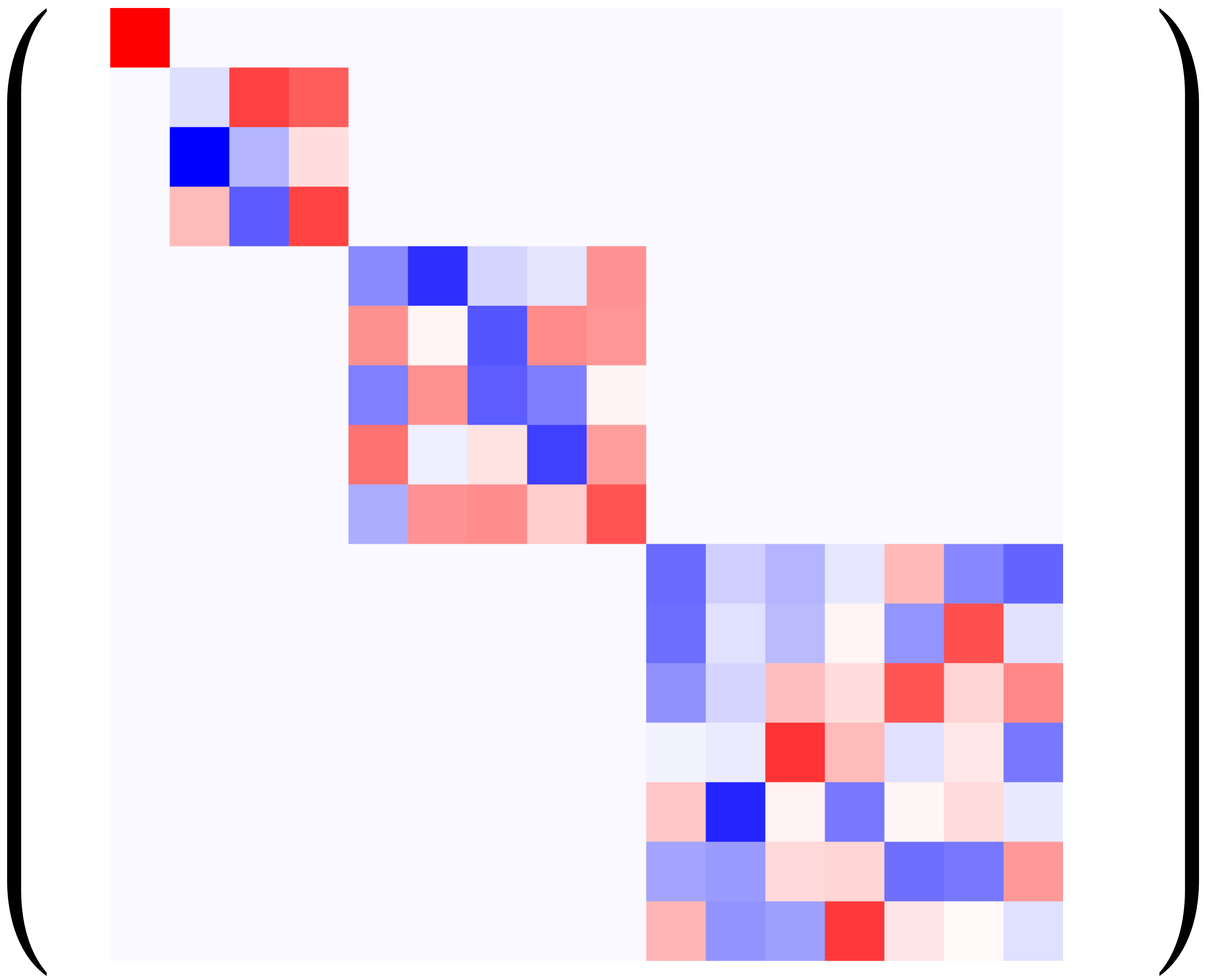




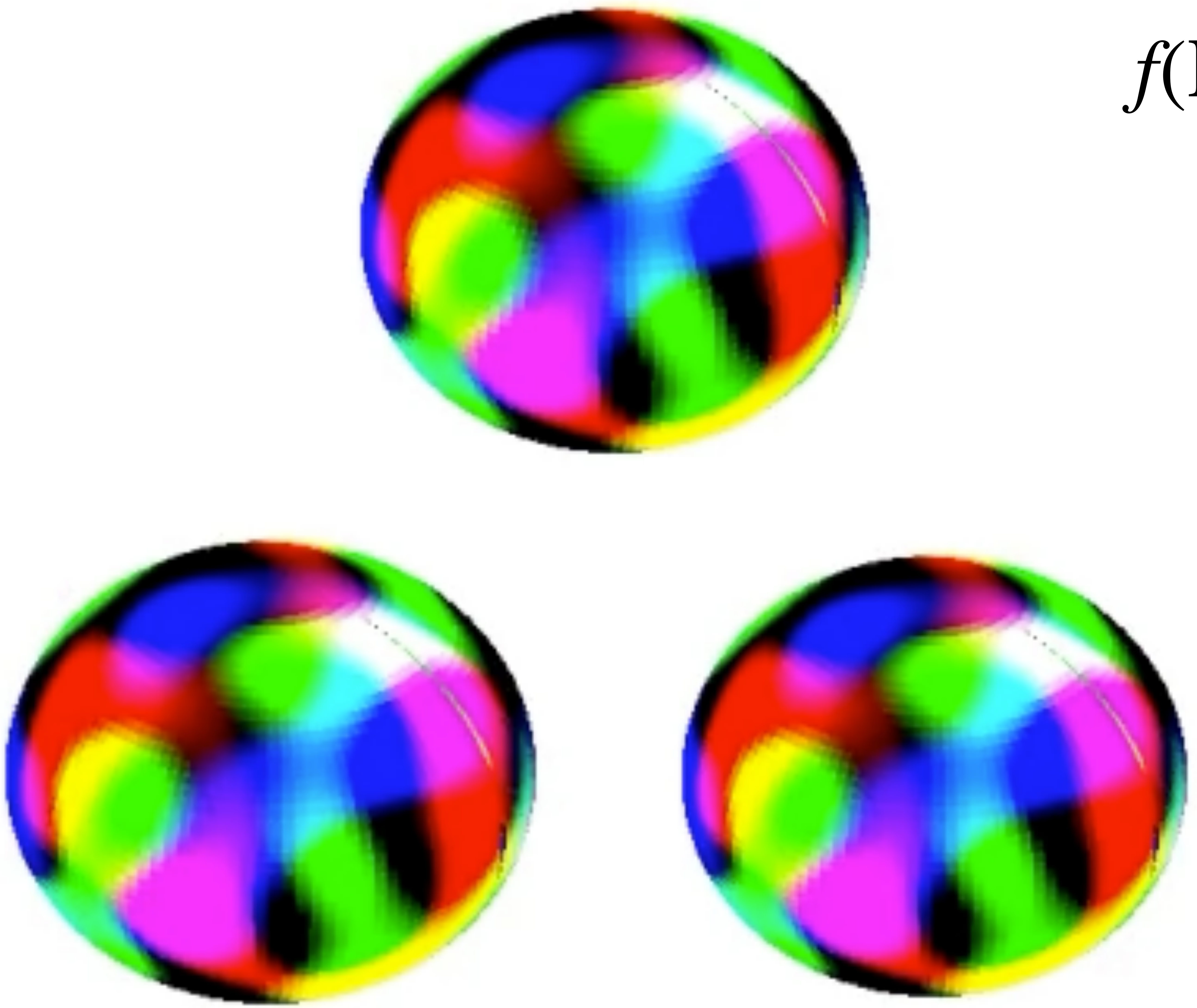
Steering the spherical harmonics basis via the "Wigner-D Matrices"



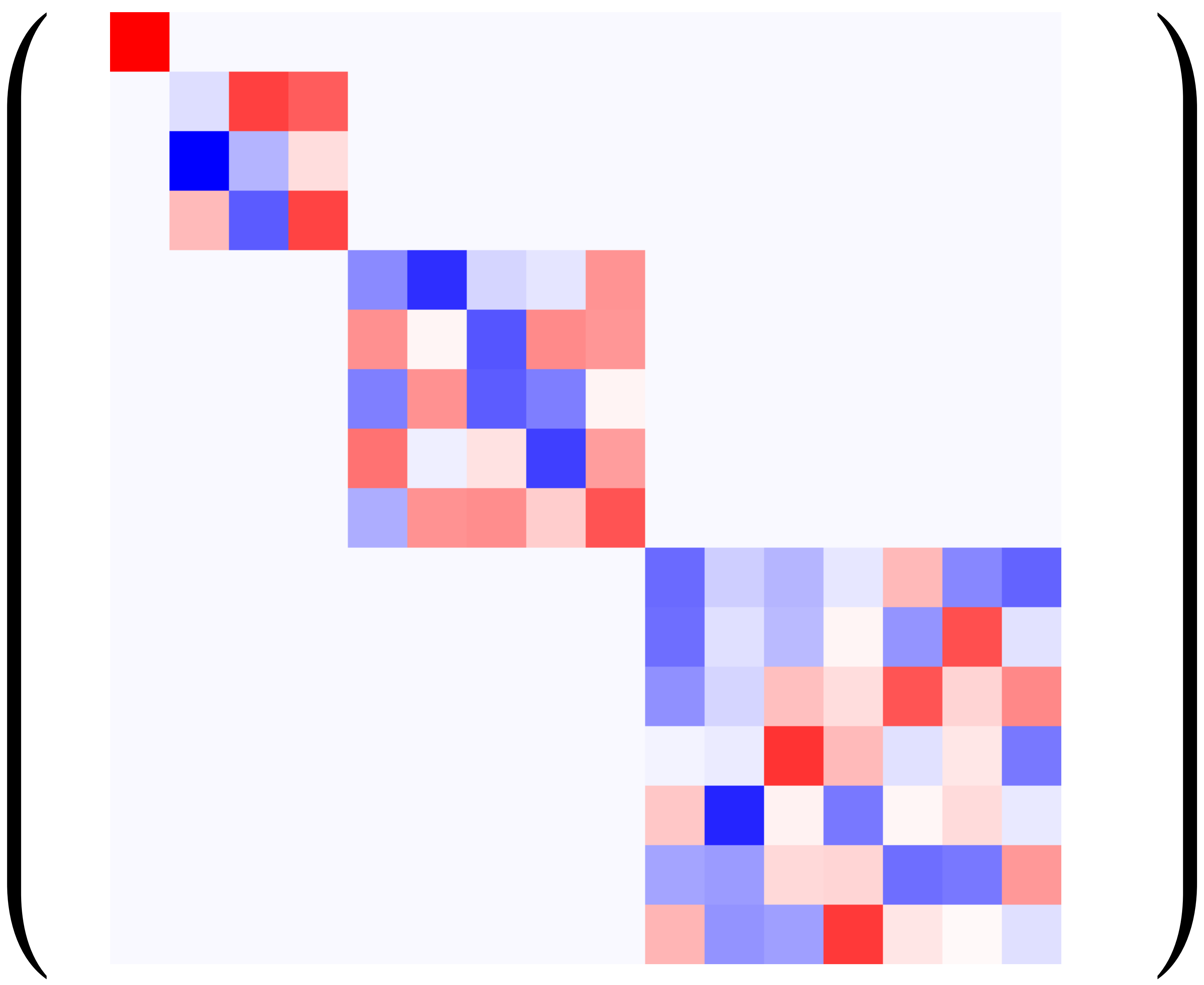
$$f(\mathbf{R}\mathbf{n}) = [\mathscr{F}_{S^2}^{-1} \ \mathbf{D}(\mathbf{R}) \ \mathscr{F}_{S^2} \ f]](\mathbf{n})$$



Steering the spherical harmonics basis via the "Wigner-D Matrices"

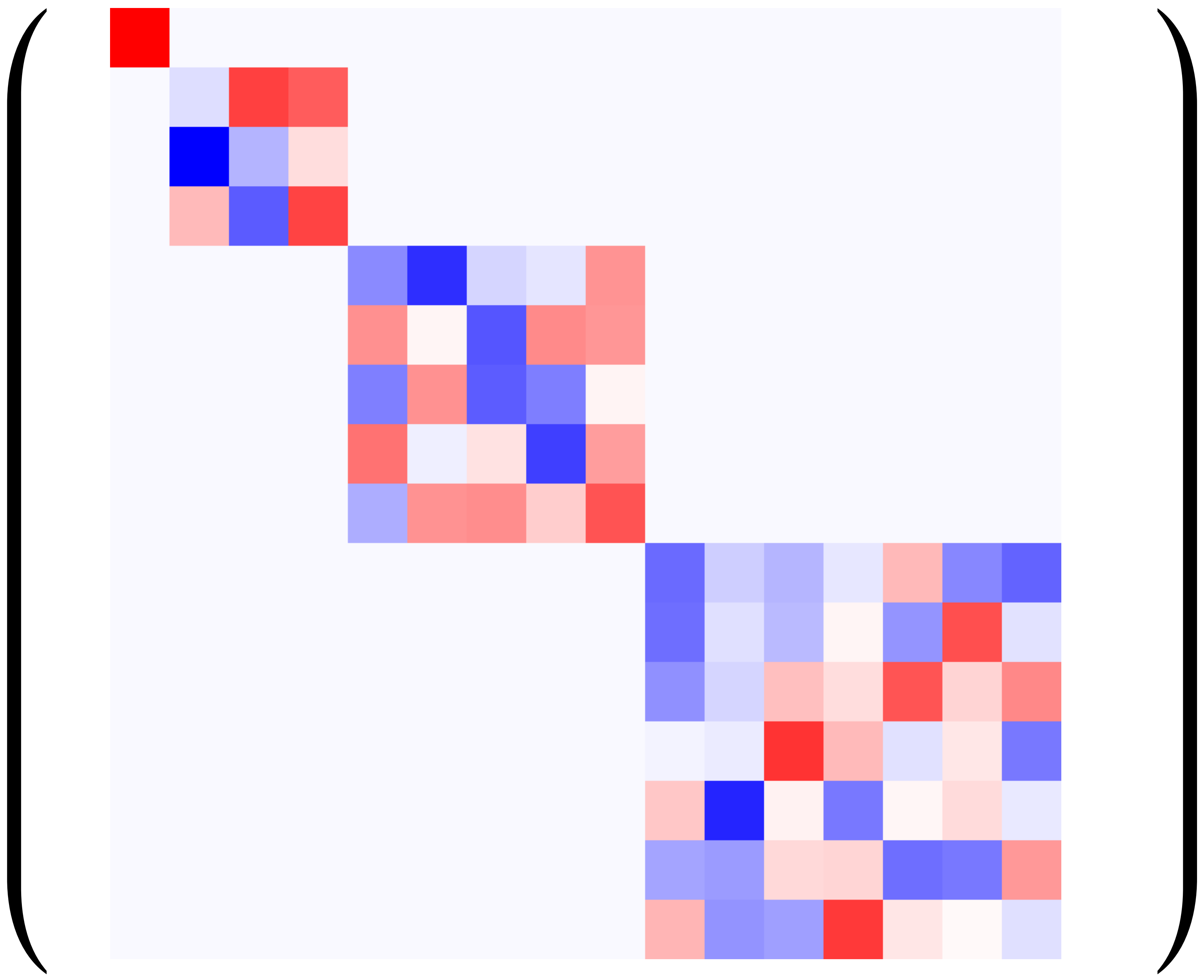


$$f(\mathbf{Rn}) = [\mathscr{F}_{S^2}^{-1}\ \mathbf{D}(\mathbf{R})\ \mathscr{F}_{S^2}\ f]](\mathbf{n})$$



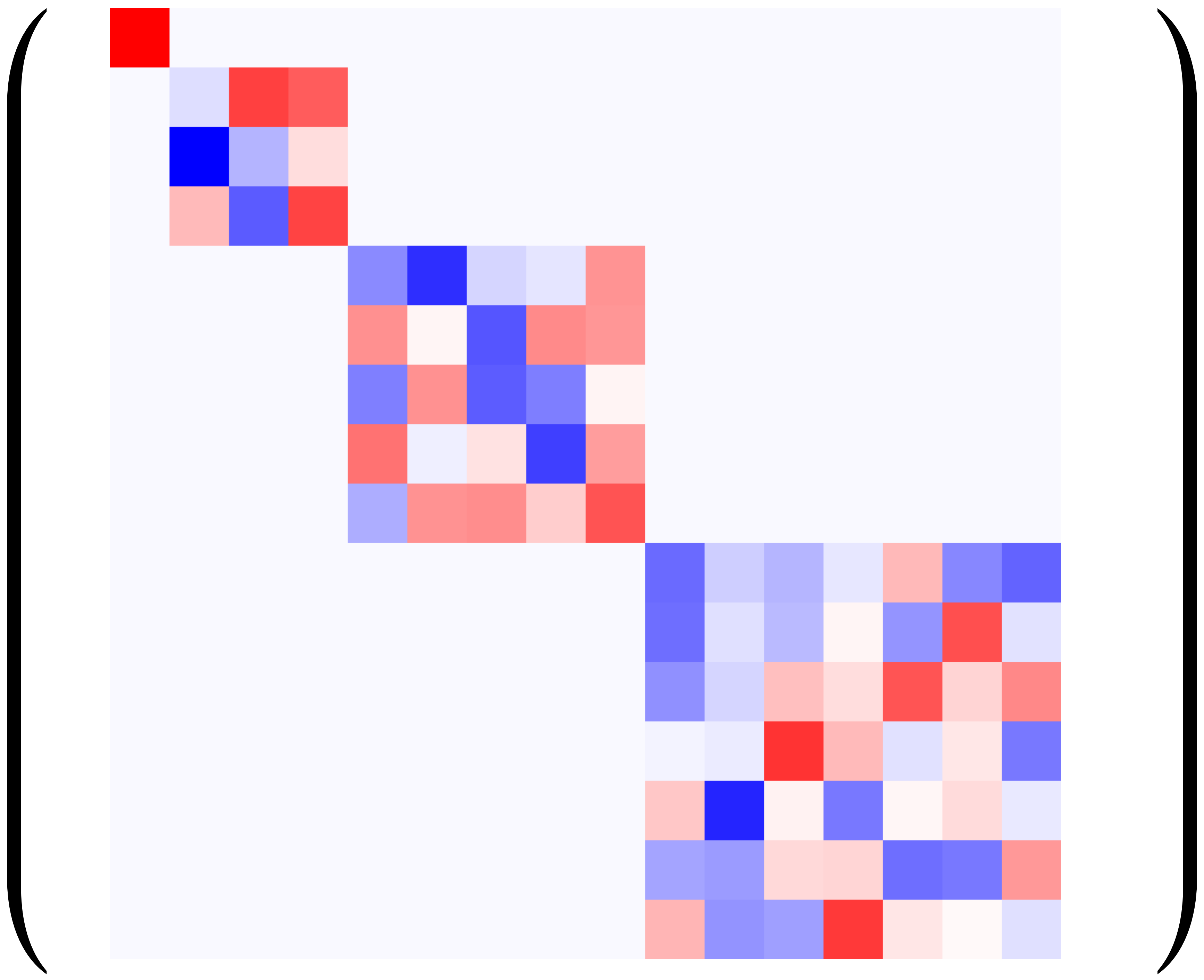
Steering the spherical harmonics basis via the "Wigner-D Matrices"

$$(\mathbf{Rn}) = [\mathcal{F}_{S^2}^{-1}\ \ \mathbf{D}(\mathbf{R})\ \ \mathcal{F}_{S^2}\ \ f]](\mathbf{n})$$



Steering the spherical harmonics basis via the "Wigner-D Matrices"

$$(\mathbf{Rn}) = [\mathcal{F}_{S^2}^{-1} \ \mathbf{D}(\mathbf{R}) \ \mathcal{F}_{S^2} \ f]](\mathbf{n})$$



PlenOctrees for Real-time Rendering of Neural Radiance Fields

Alex Yu UC Berkeley
Ruilong Li UC Berkeley & USC ICT
Matt Tancik UC Berkeley
Hao Li Pinscreen & UC Berkeley
Ren Ng UC Berkeley
Angjoo Kanazawa UC Berkeley

Project website & online demo: alexyu.net/plenoctrees

PlenOctrees for Real-time Rendering of Neural Radiance Fields

Alex Yu UC Berkeley

Ruilong Li UC Berkeley & USC ICT

Matt Tancik UC Berkeley

Hao Li Pinscreen & UC Berkeley

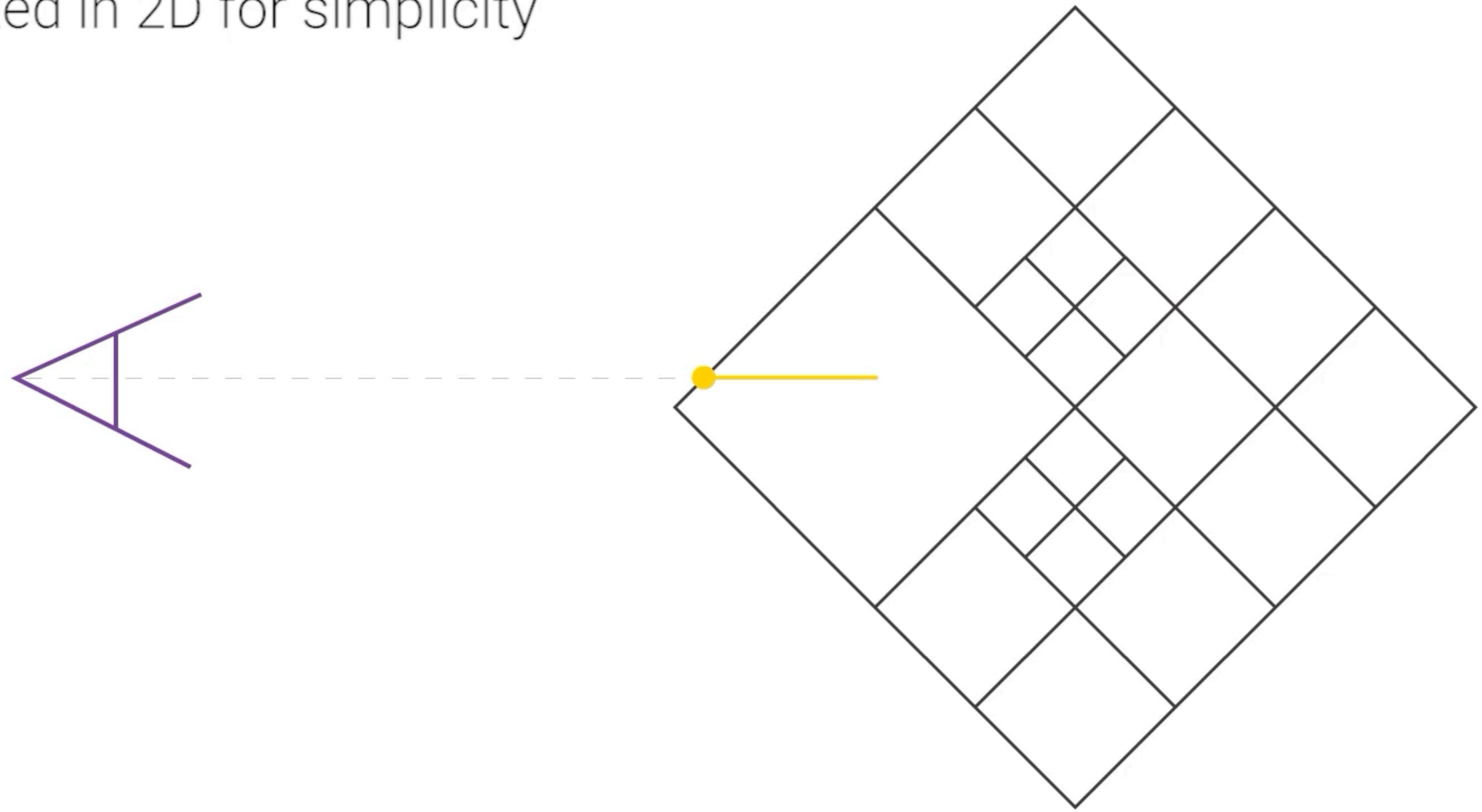
Ren Ng UC Berkeley

Angjoo Kanazawa UC Berkeley

Project website & online demo: alexyu.net/plenoctrees

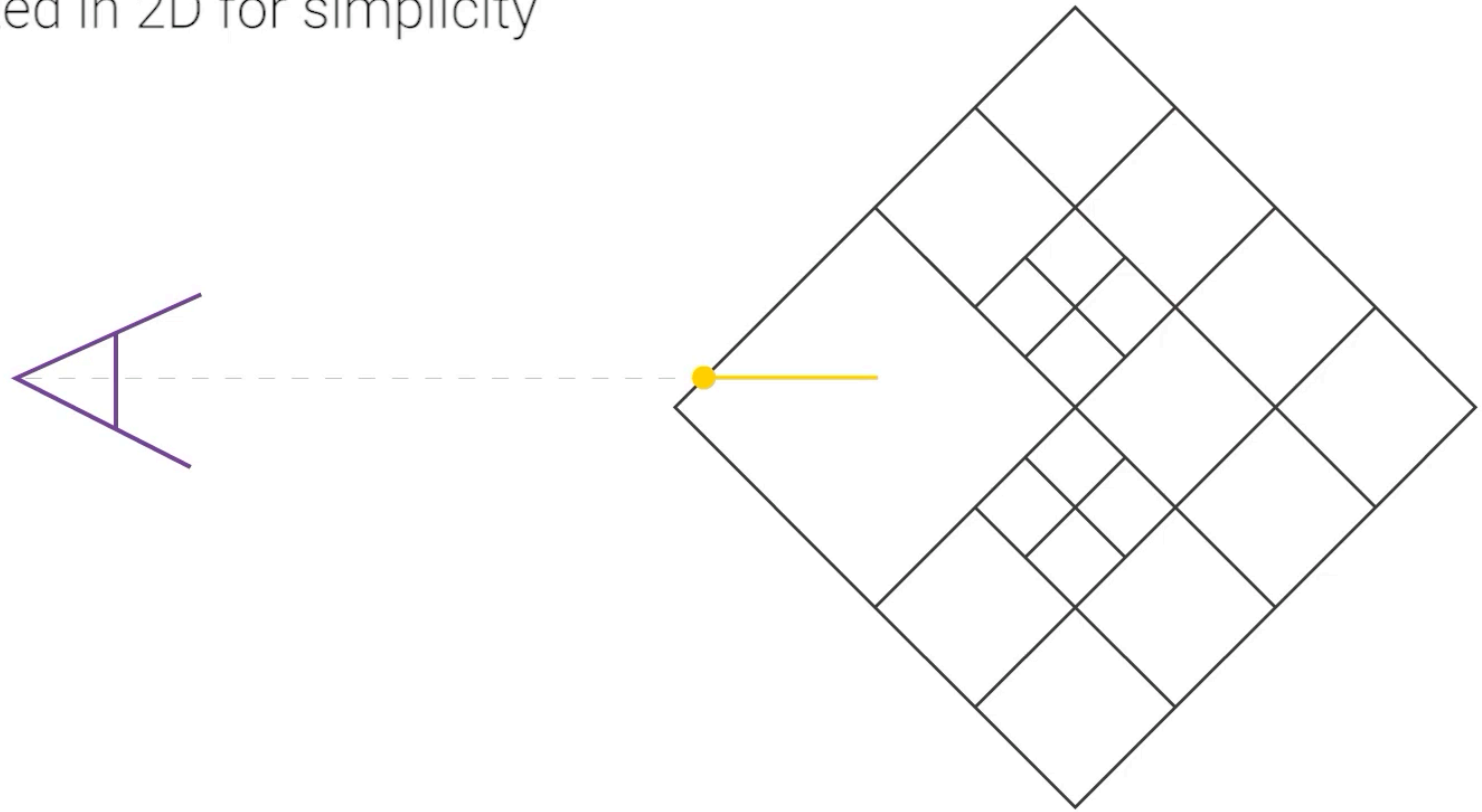
PlenOctree Rendering

* Illustrated in 2D for simplicity

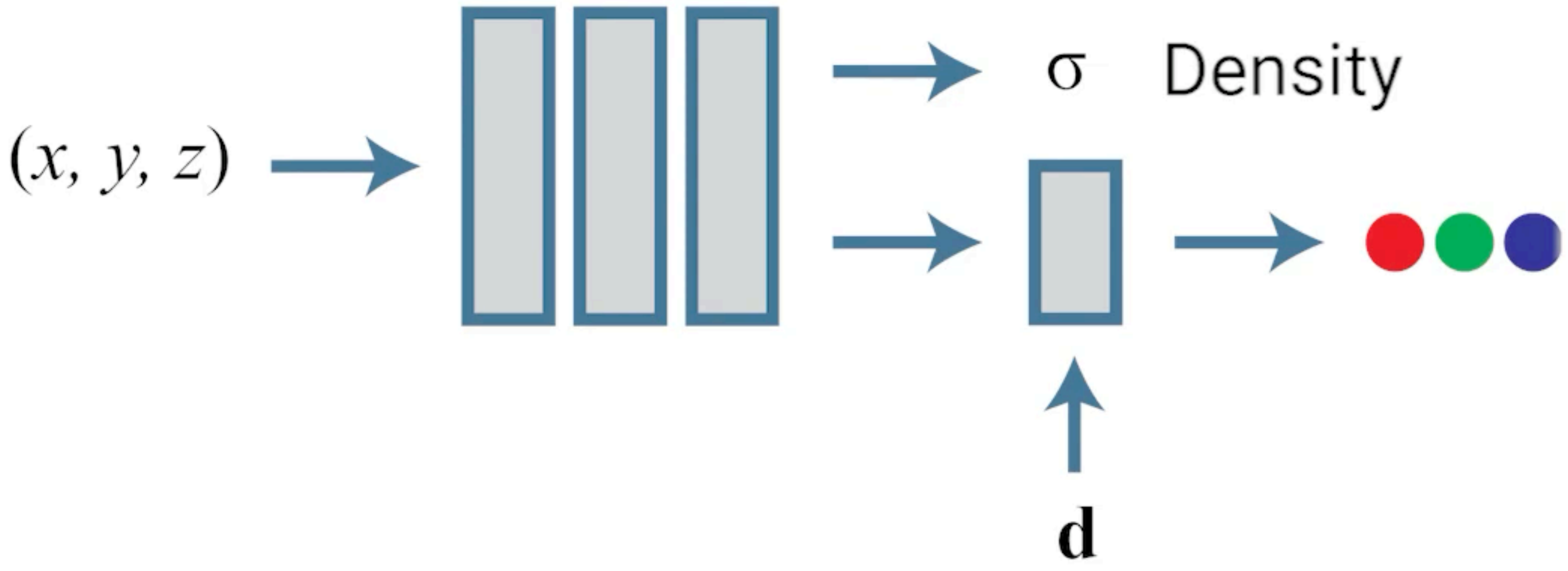


PlenOctree Rendering

* Illustrated in 2D for simplicity

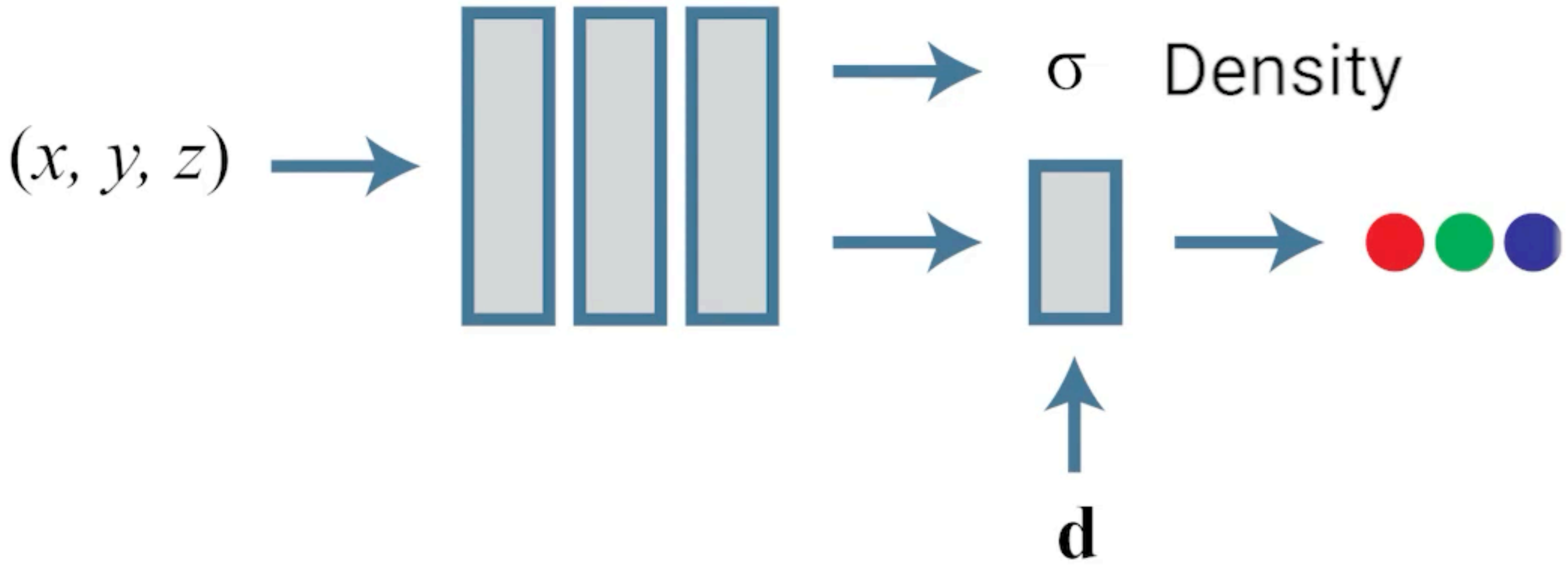


NeRF



NeRF with Spherical Harmonics (NeRF-SH)

NeRF



NeRF with Spherical Harmonics (NeRF-SH)

PlenOctree Viewer

https://alexyu.net/plenoctrees/demo/?load=../data/materials.npz

Search

Other Bookmarks

PlenOctree Viewer   Options

PlenOctree Viewer

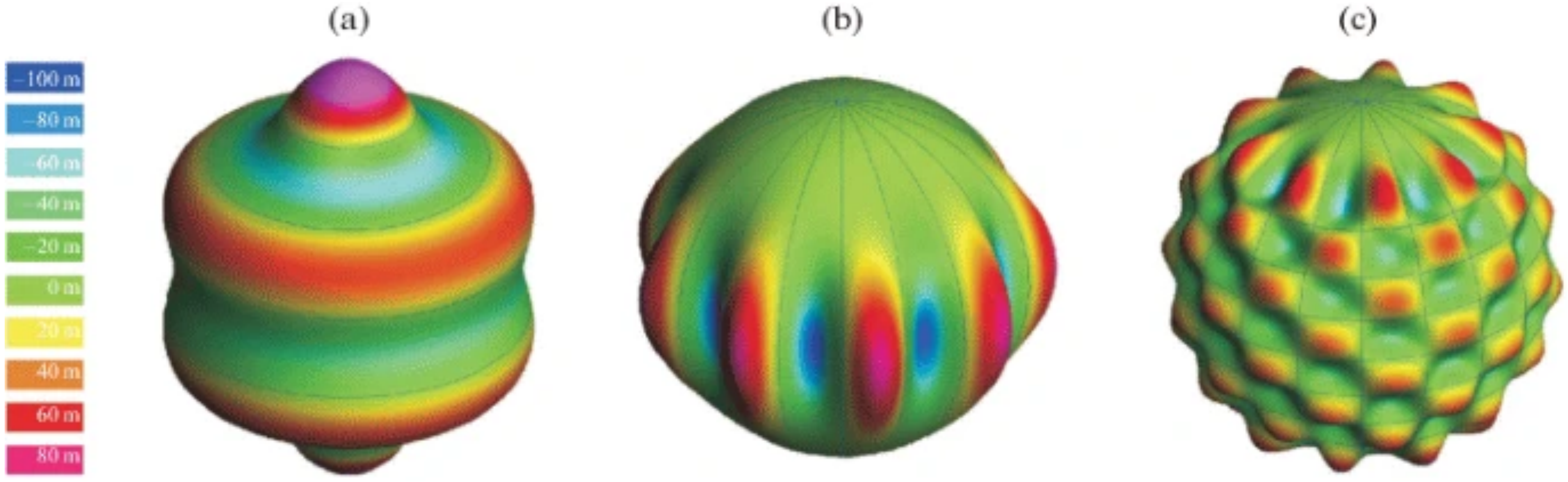
https://alexyu.net/plenoctrees/demo/?load=../data/materials.npz

Search

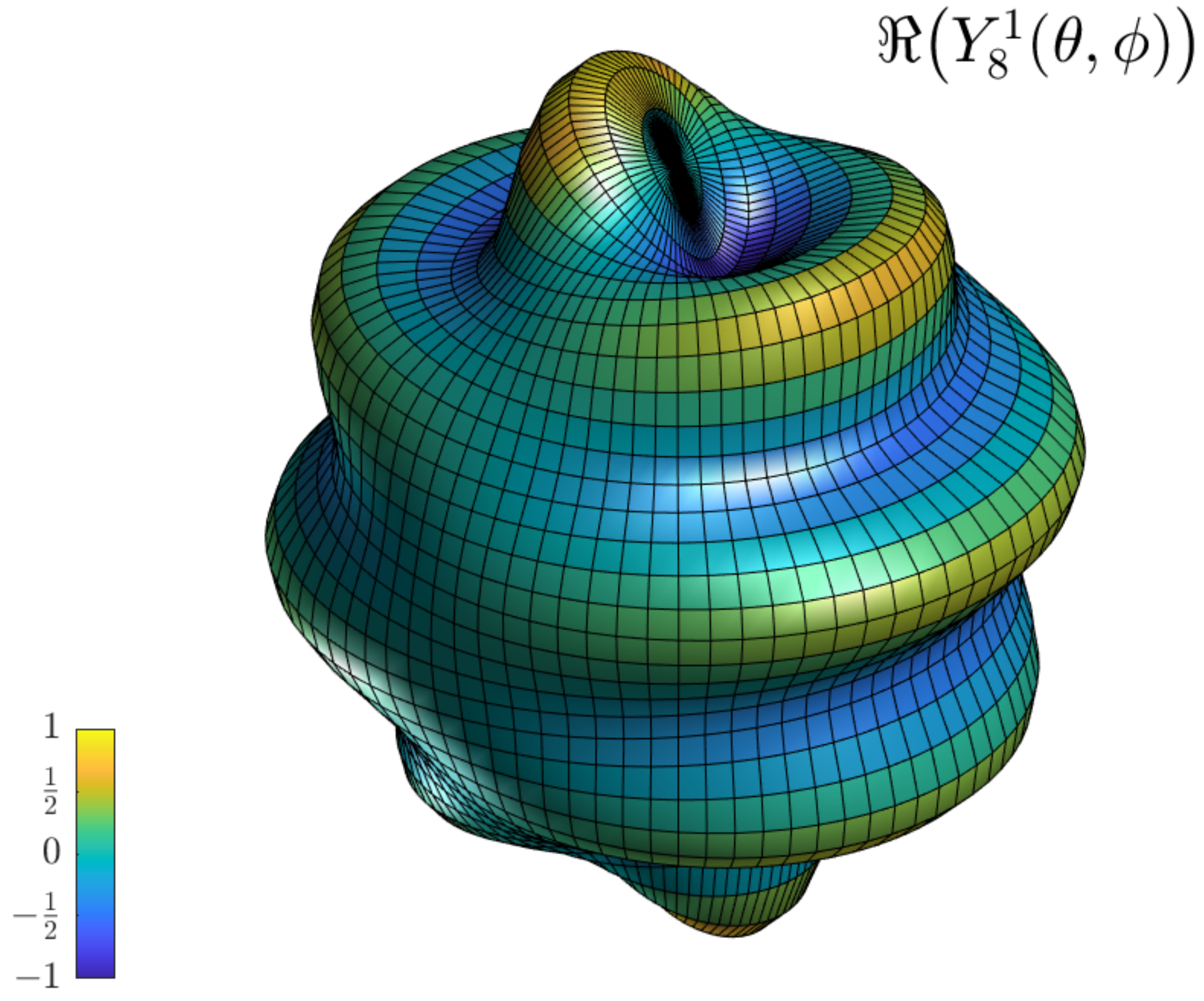
Other Bookmarks

PlenOctree Viewer   Options

Also just very pretty :)



Mikhailov et al. High-Degree Models of the Earth’s Gravity Field: History of Development, Assessment of Prospects and Resolution. Seism. Instr. 57, 446–461 (2021).



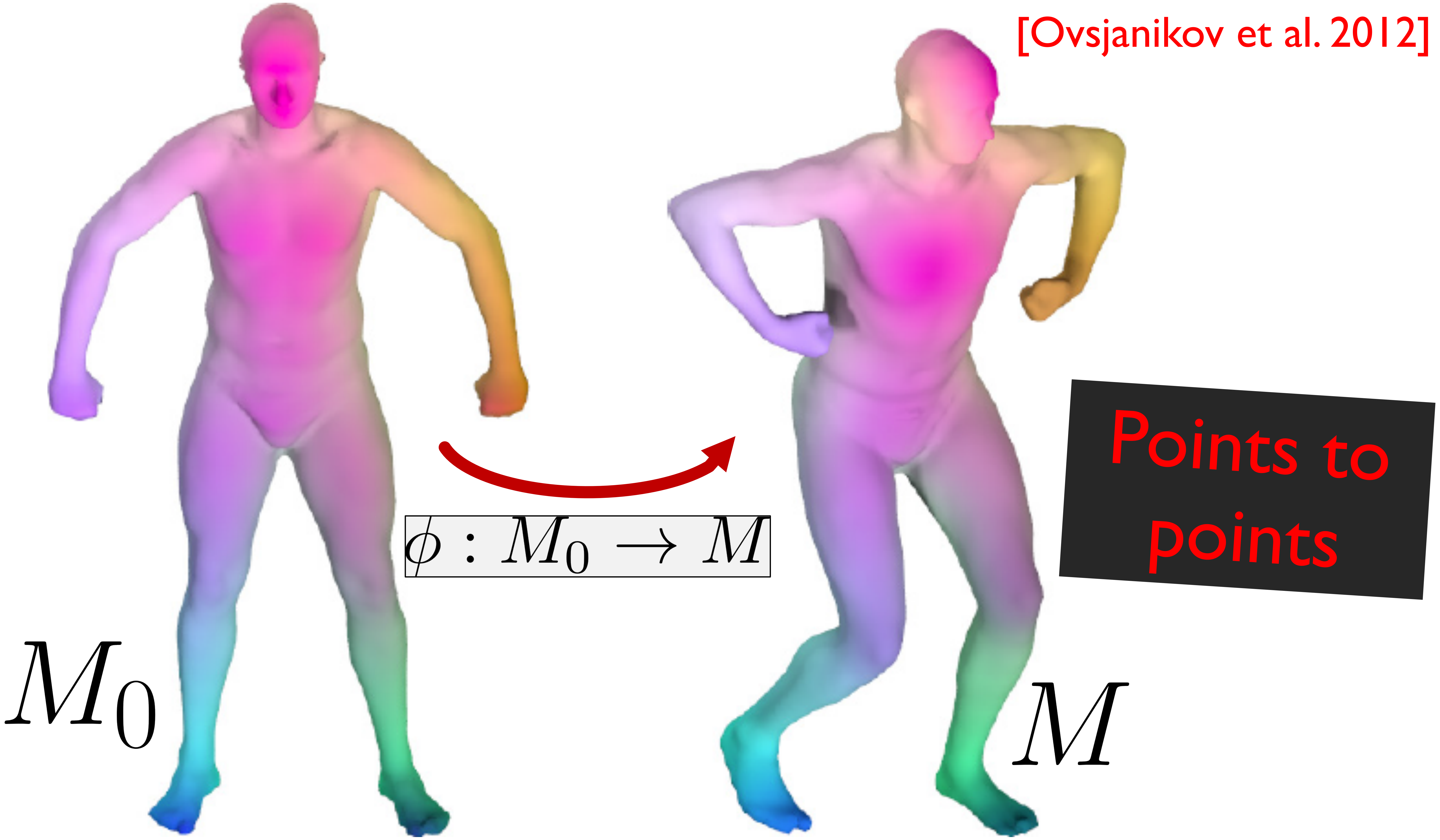
https://x.com/00aleph00/status/1297675678456918016

Similar approach works for correspondences on meshes - called “Functional Maps”

Functional Maps for Shape-to-Shape Correspondence

Slide Credit: Justin Solomon

[Ovsjanikov et al. 2012]

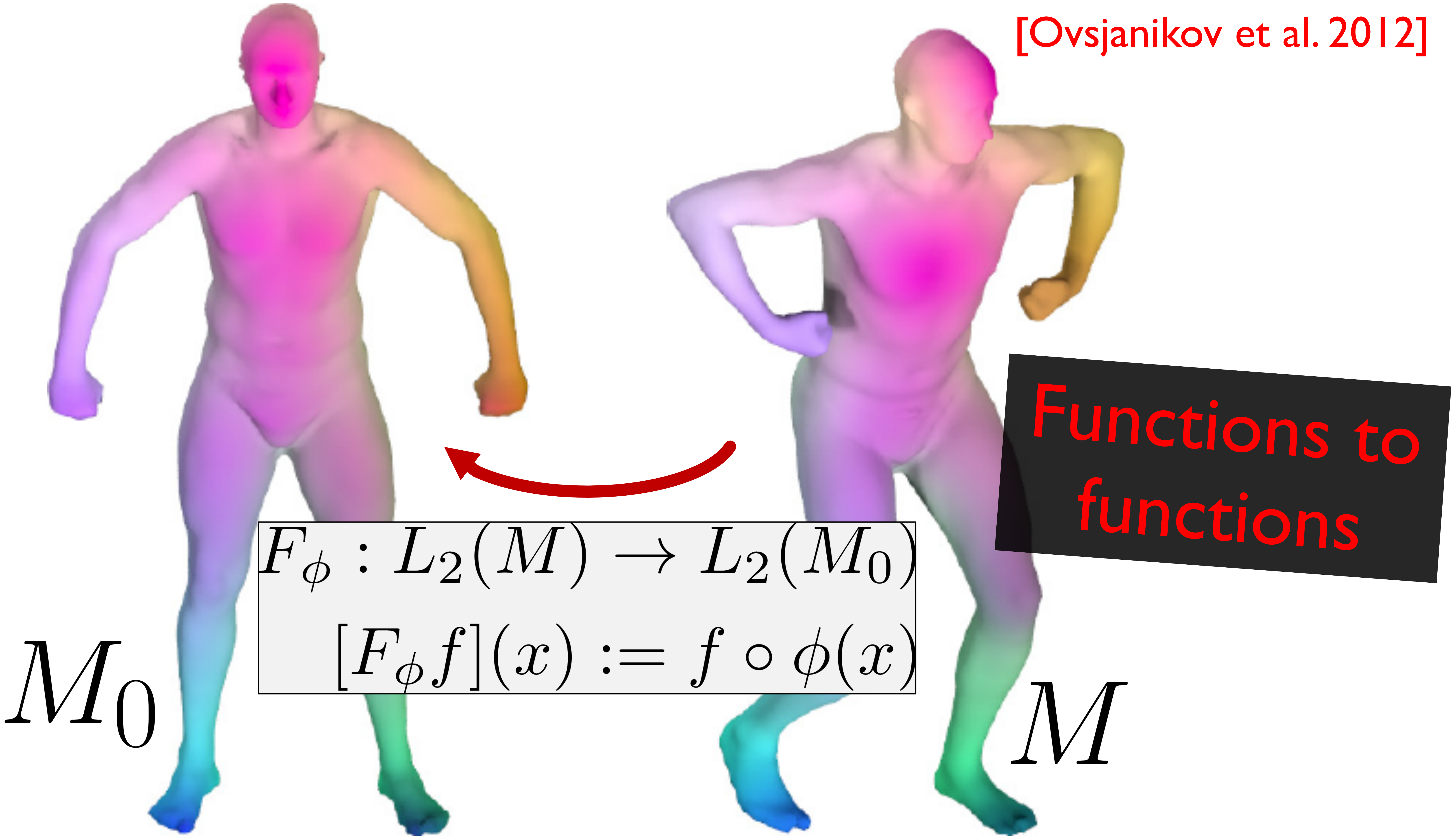


Points on M_0 to points on M

Functional Maps for Shape-to-Shape Correspondence

Slide Credit: Justin Solomon

[Ovsjanikov et al. 2012]



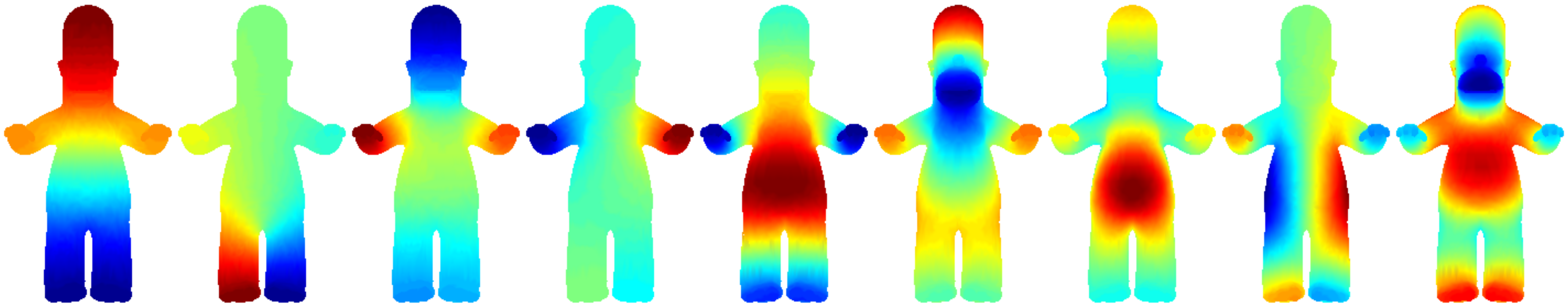
$$F_\phi : L_2(M) \to L_2(M_0)$$
$$[F_\phi f](x) := f \circ \phi(x)$$

Functions on M to functions on M_0

Functional Maps for Shape-to-Shape Correspondence

Slide Credit: Justin Solomon

[Ovsjanikov et al. 2012]



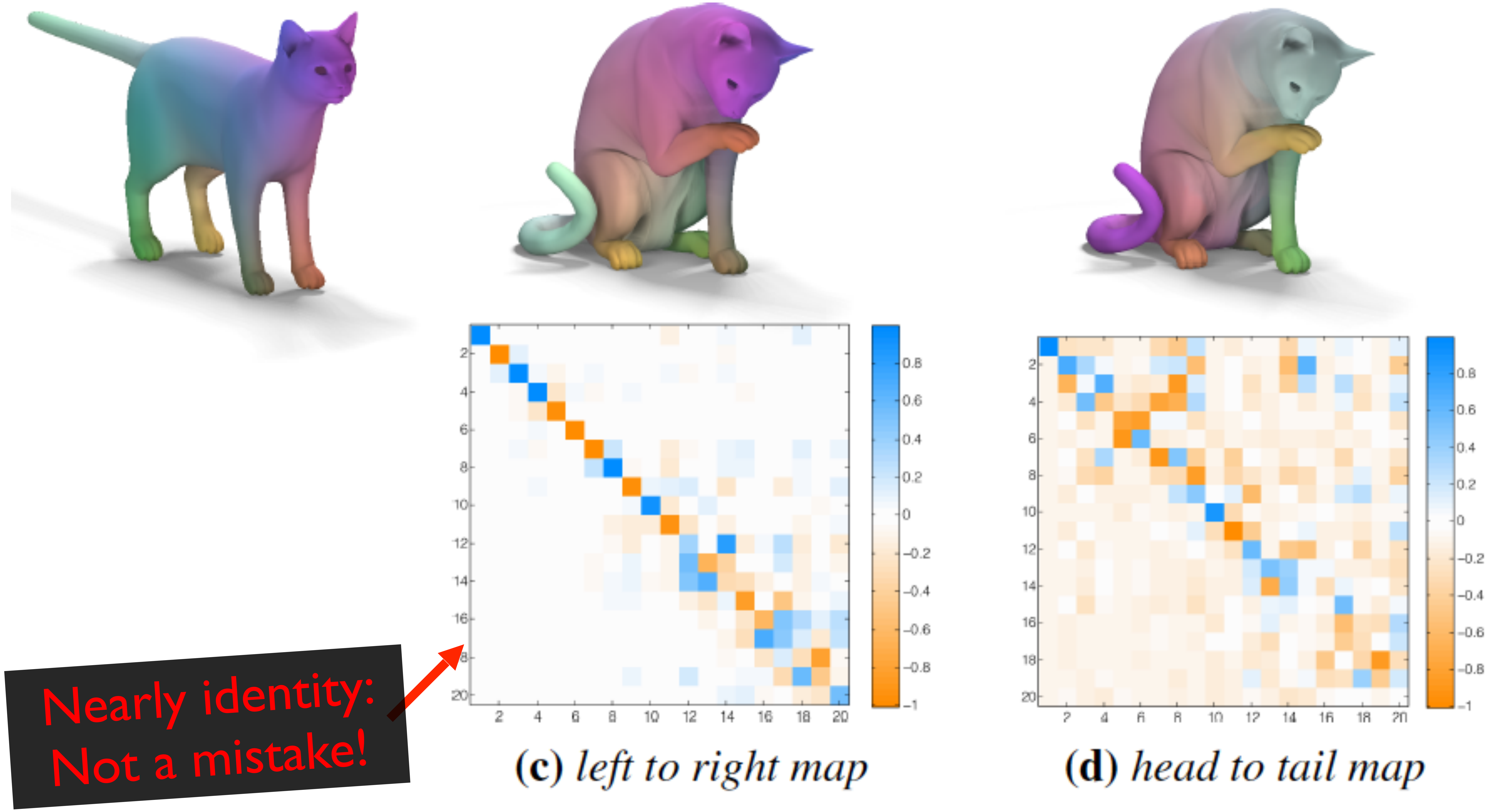
$$f(x) = \sum_i a_i \psi_i(x)$$

Functional map:

Matrix taking Laplace-Beltrami (Fourier) coefficients on M to coefficients on M_0

Example Maps

Slide Credit: Justin Solomon



(c) *left to right map*

(d) *head to tail map*

Adapted from slides by Q. Huang, V. Kim

Functional Maps for Shape-to-Shape Correspondence

Slide Credit: Justin Solomon

- Simple Algorithm
  - Compute some geometric functions to be preserved: A, B
  - Solve in least-squares sense for C: B = C A
- Additional Considerations
  - Favor commutativity
  - Favor orthonormality (if shapes are isometric)
  - Efficiently getting point-to-point correspondences

Adapted from slides by Q. Huang, V. Kim

Ovsjanikov'12

Useful Survey

Slide Credit: Justin Solomon

Computing and Processing Correspondences with Functional Maps

SIGGRAPH 2017 Course Notes

Organizers & Lecturers:

Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, Alex Bronstein

Deep Functional Maps

Slide Credit: Justin Solomon

Many extensions:
- Incorporate commutativity
- Learn shape features
- Self-supervision

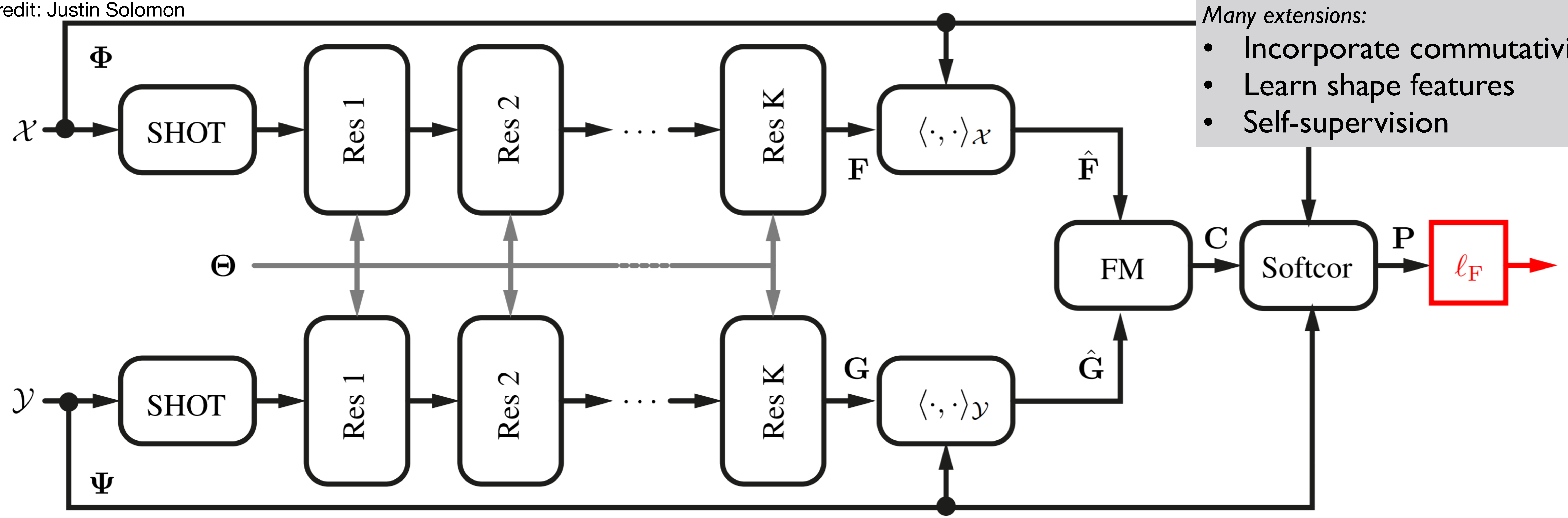
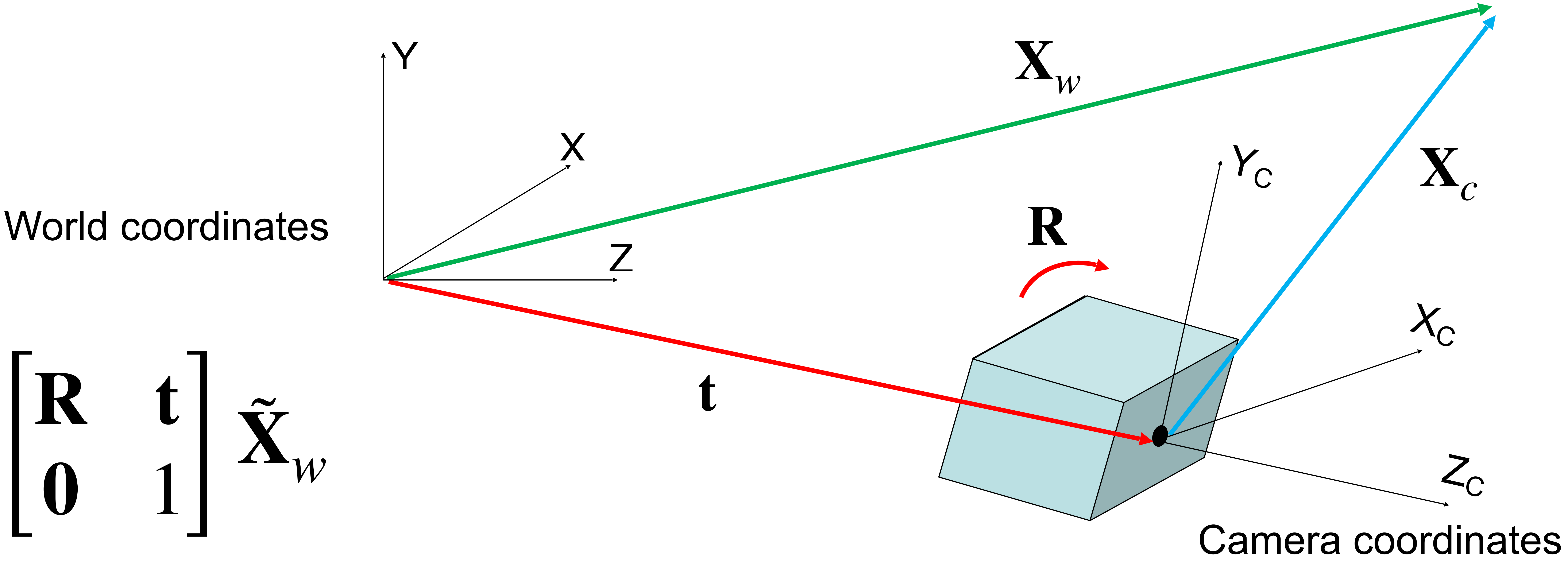


Figure 3. **FMNet architecture.** Input point-wise descriptors (SHOT [38] in this paper) from a pair of shapes are passed through an identical sequence of operations (with shared weights), resulting in refined descriptors $\mathbf{F}, \mathbf{G}$. These, in turn, are projected onto the Laplacian eigenbases $\mathbf{\Phi}, \mathbf{\Psi}$ to produce the spectral representations $\hat{\mathbf{F}}, \hat{\mathbf{G}}$. The functional map (FM) and soft correspondence (Softcor) layers, implementing Equations (3) and (6) respectively, are not parametric and are used to set up the geometrically structured loss ℓ_{F} (5).

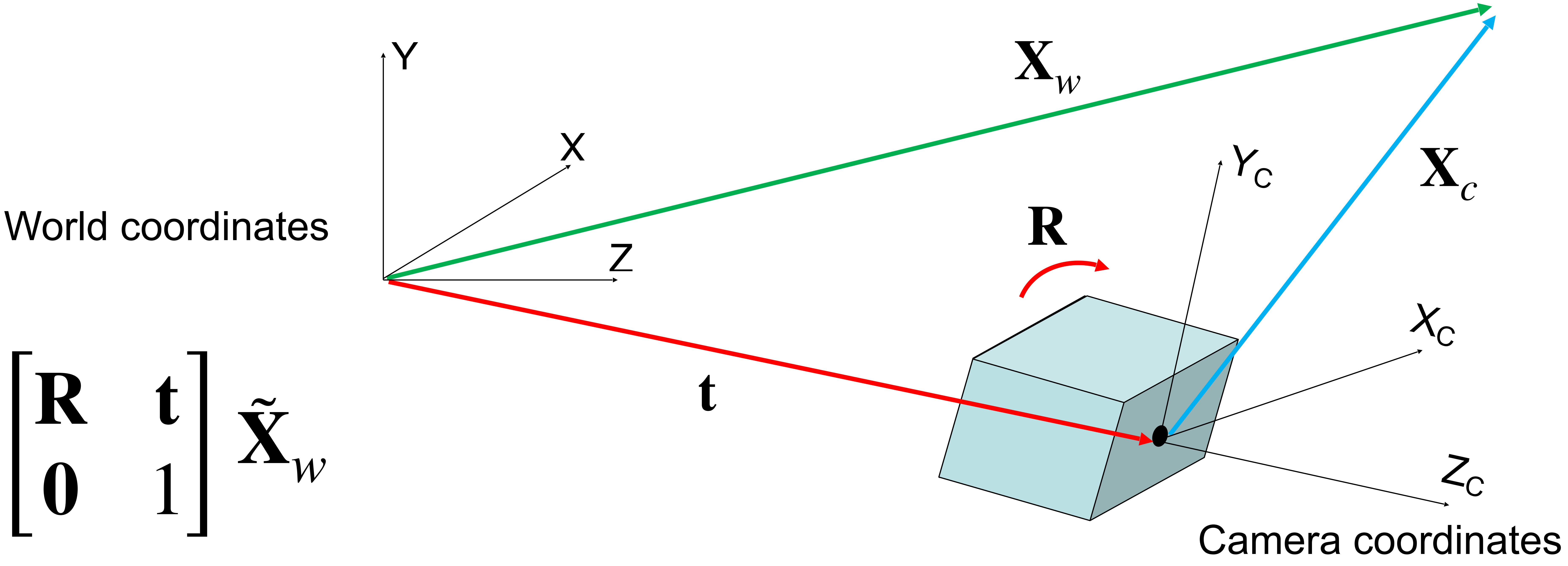
"Deep functional maps: Structured prediction for dense shape correspondence" (Litany et al. 2017)

What about the group of Camera Transformations, SE(3)?



$$\tilde{\mathbf{X}}_C = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tilde{\mathbf{X}}_w$$

What about the group of Camera Transformations, SE(3)?

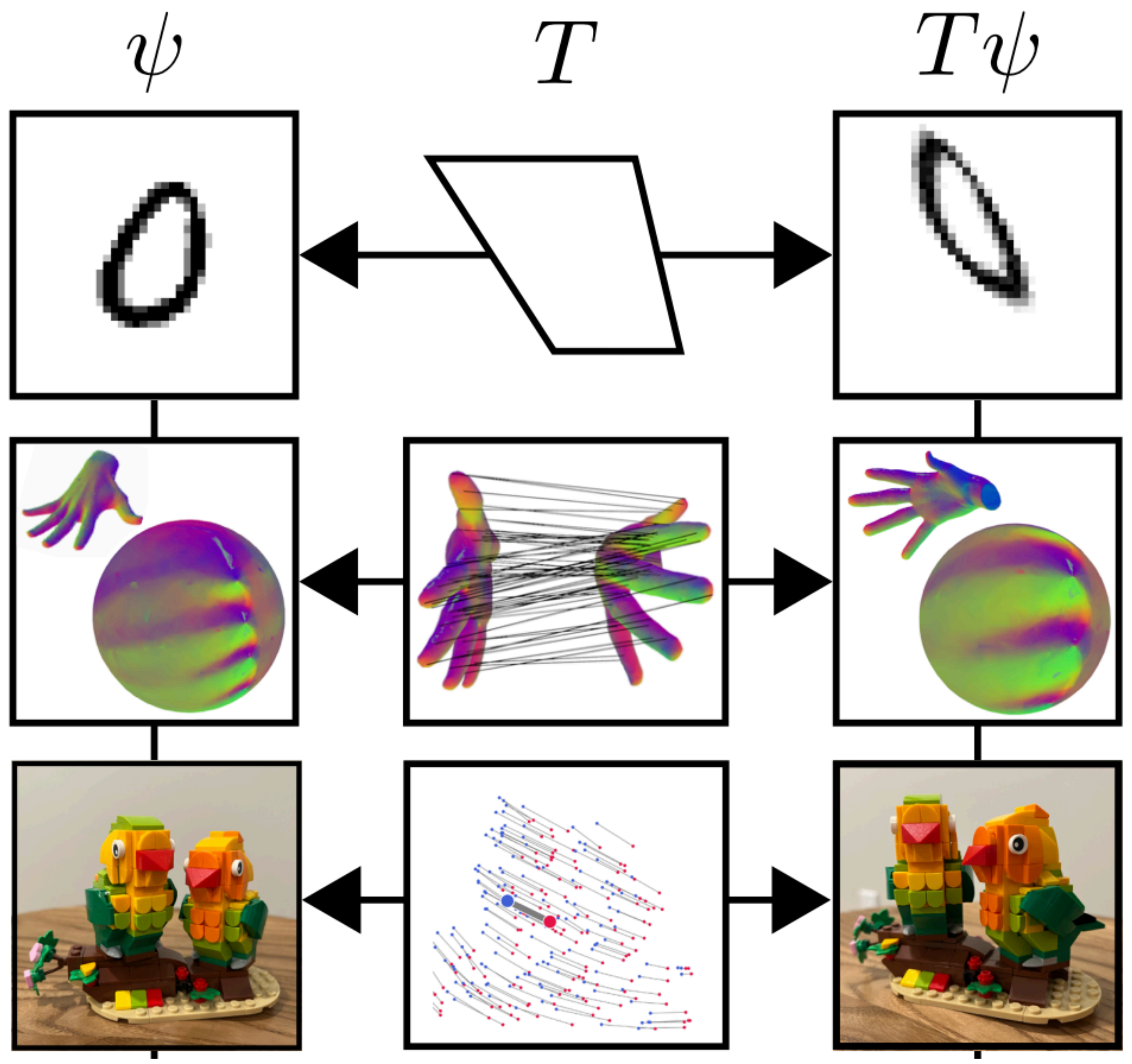


$$\tilde{\mathbf{X}}_C = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tilde{\mathbf{X}}_w$$

The recipe still works! But everything gets more complicated. SE(3) is not compact, its matrix representations for functions over $\mathbb{R}^3$ are infinite-dimensional, and so is the steerable basis…

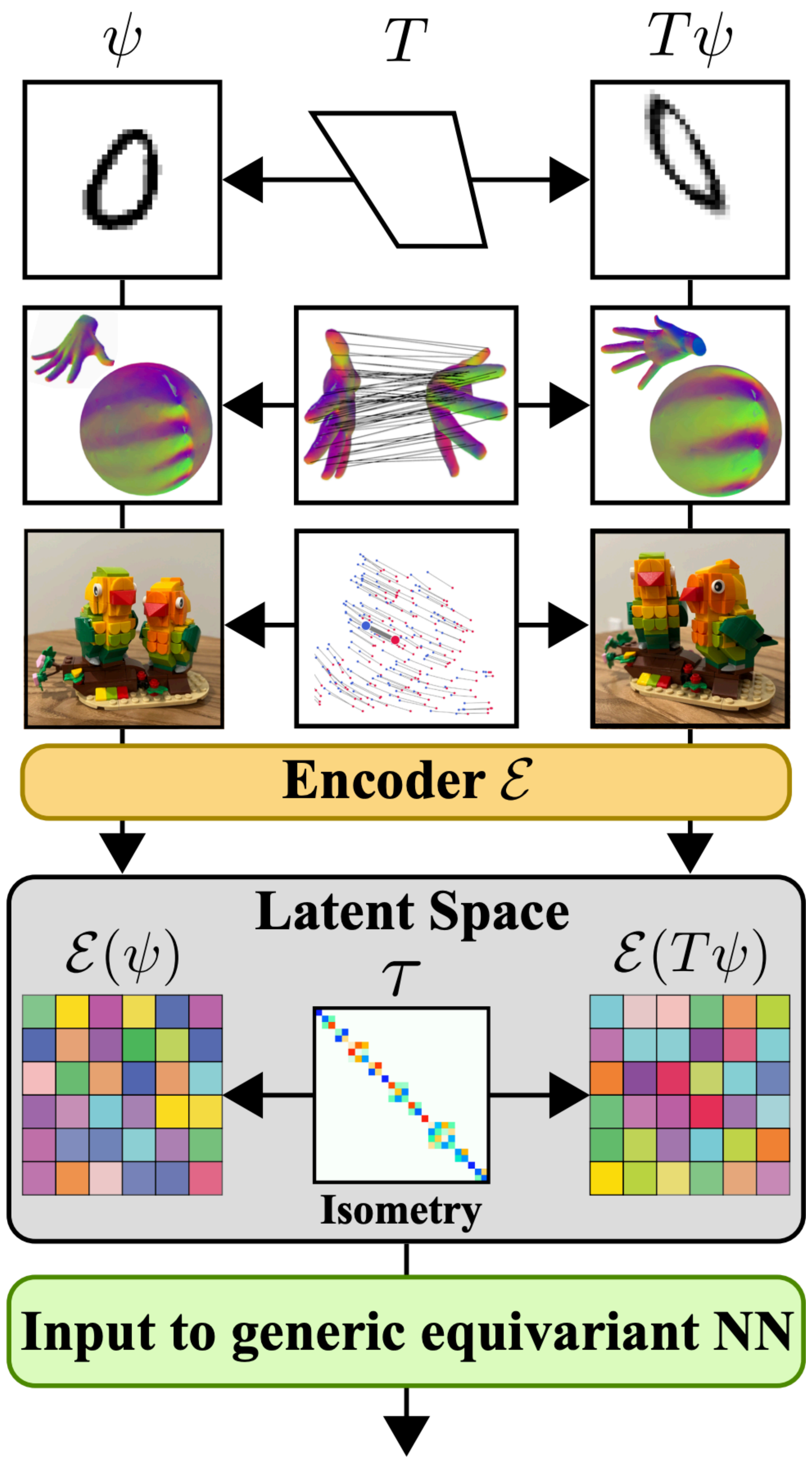
Papers!

Neural Isometries



Real-World Transformations are difficult in observation space…

Neural Isometries



Real-World Transformations are difficult in observation space…

…but there must exist a latent space where they are simple

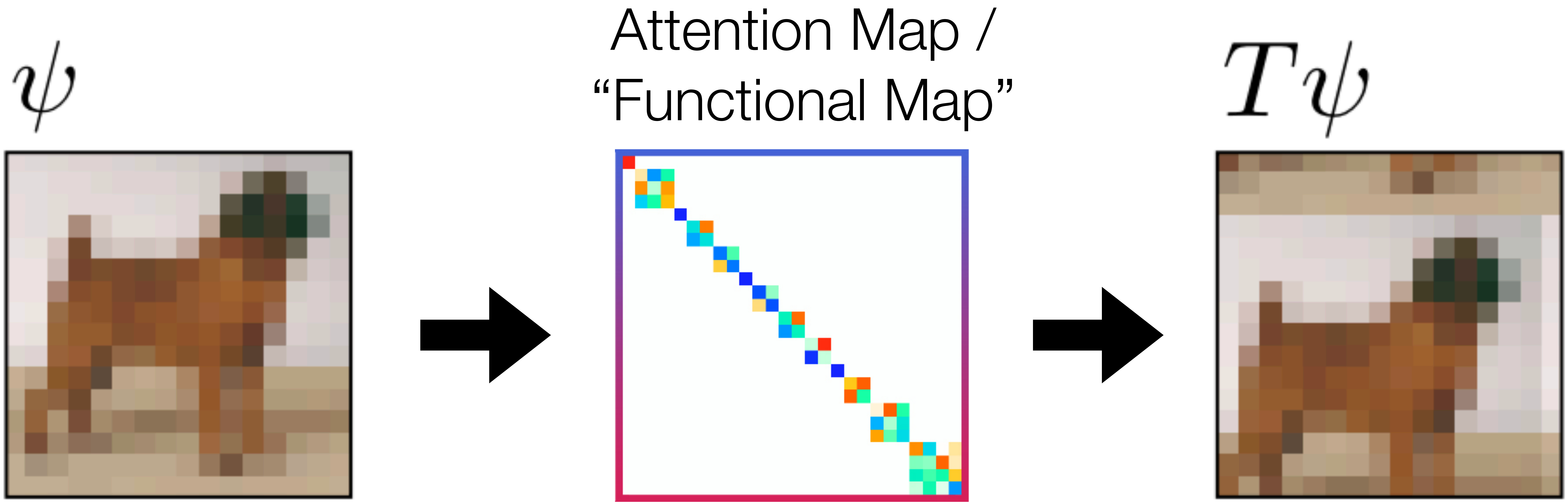
Toy example: 2D Shifts!

ψ

$T\psi$

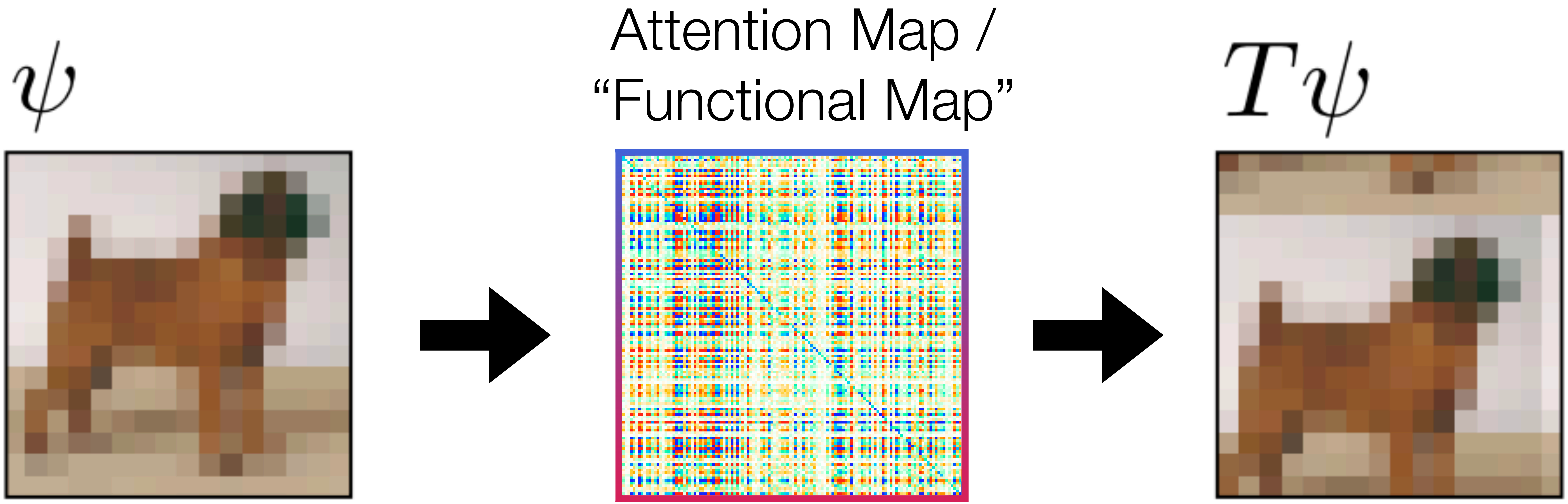
How to parameterize the space of all maps between these images?

Toy example: 2D Shifts!



Express every pixel as weighted sum of pixels in other image.

Toy example: 2D Shifts!

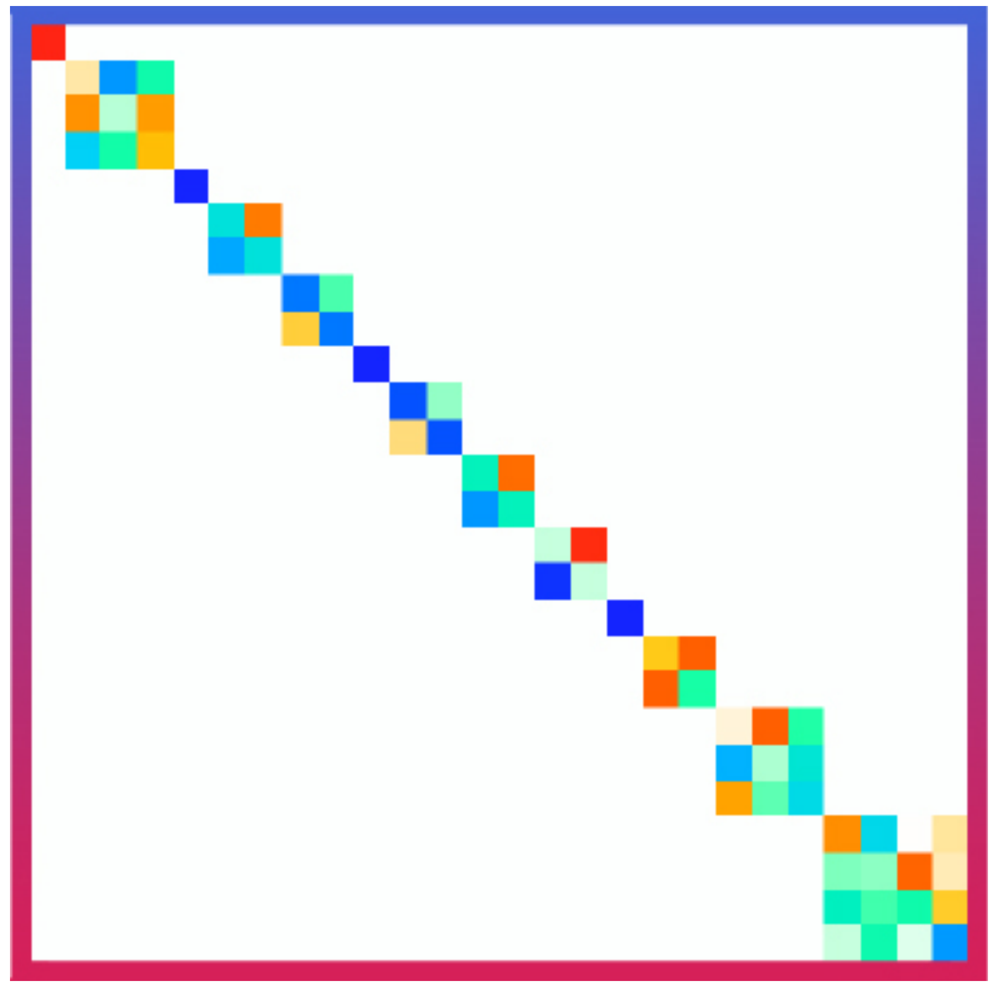


Express every pixel as weighted sum of pixels in other image.

Toy example: 2D Shifts!

ψ

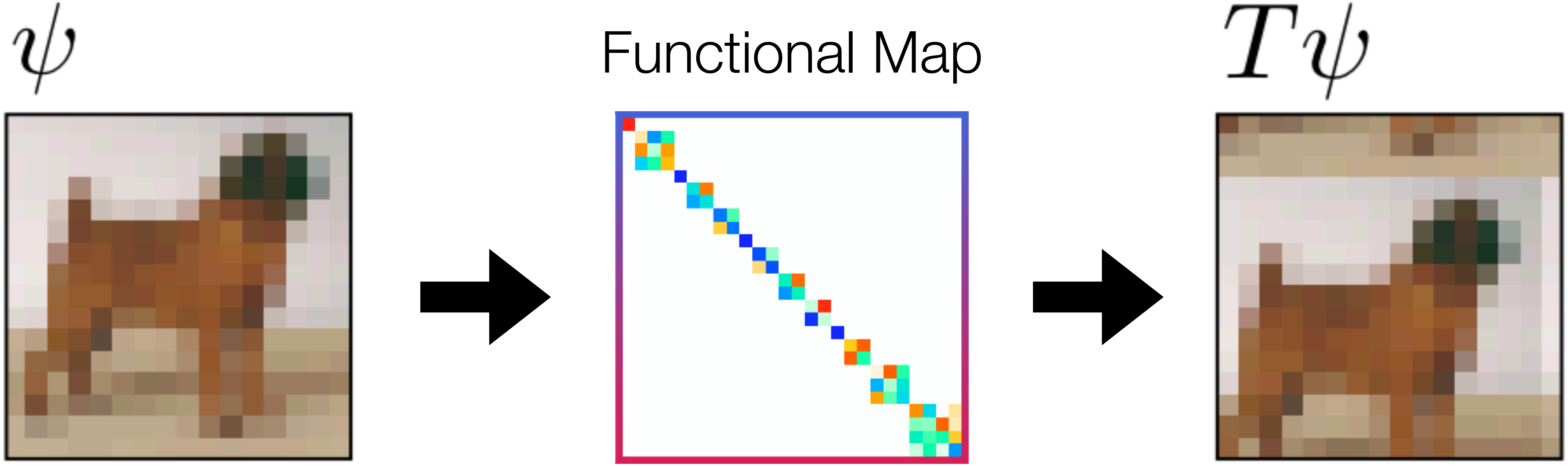
Functional Map



$T\psi$

How to find the *correct* map?

Toy example: 2D Shifts!

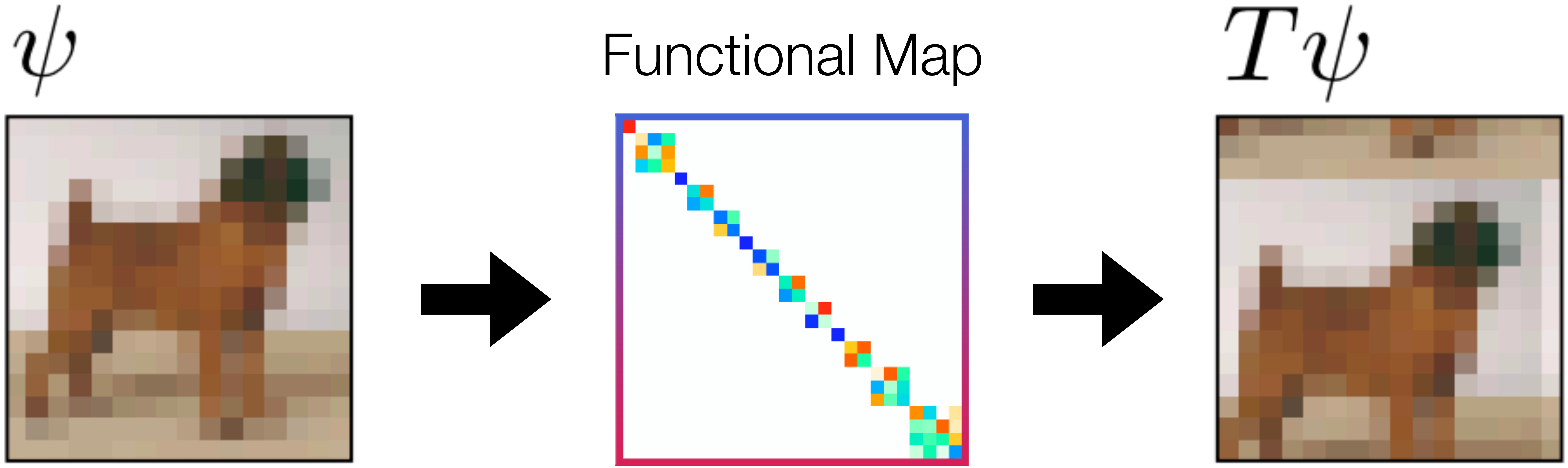


How to find the *correct* map?

Assumption 1: Our transformations form a *group*.

Assumption 2: Our group can be well-approximated by a *compact group.*

Toy example: 2D Shifts!

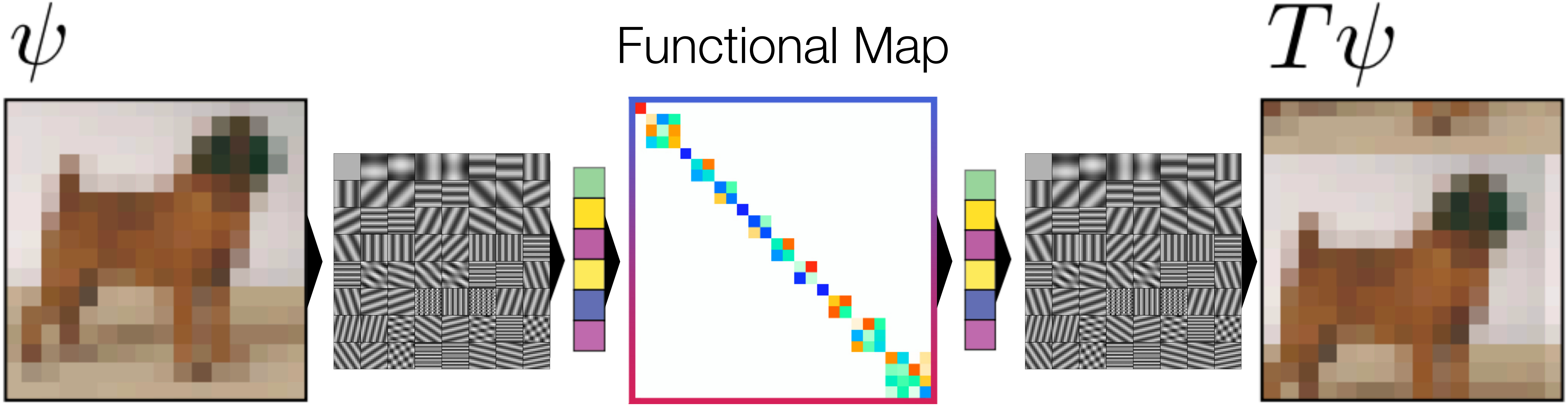


How to find the *correct* map?

Assumption 1: Our transformations form a *group*.
Assumption 2: Our group can be well-approximated by a *compact group.*

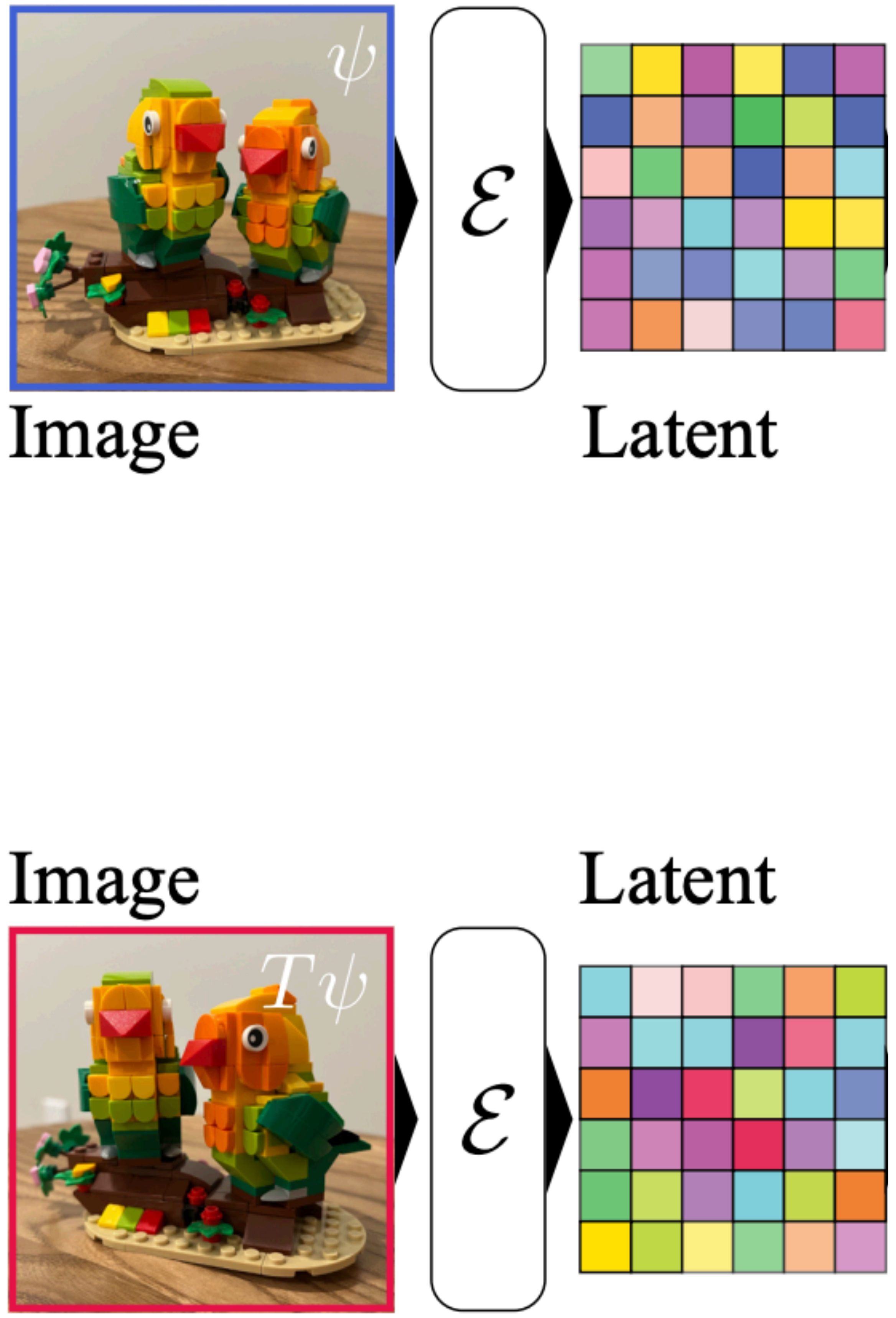
Peter-Weyl Theorem: There exists a basis for images in which our transformations will be block-diagonal!

Learn Function Basis and Solve For Transformation in that Basis

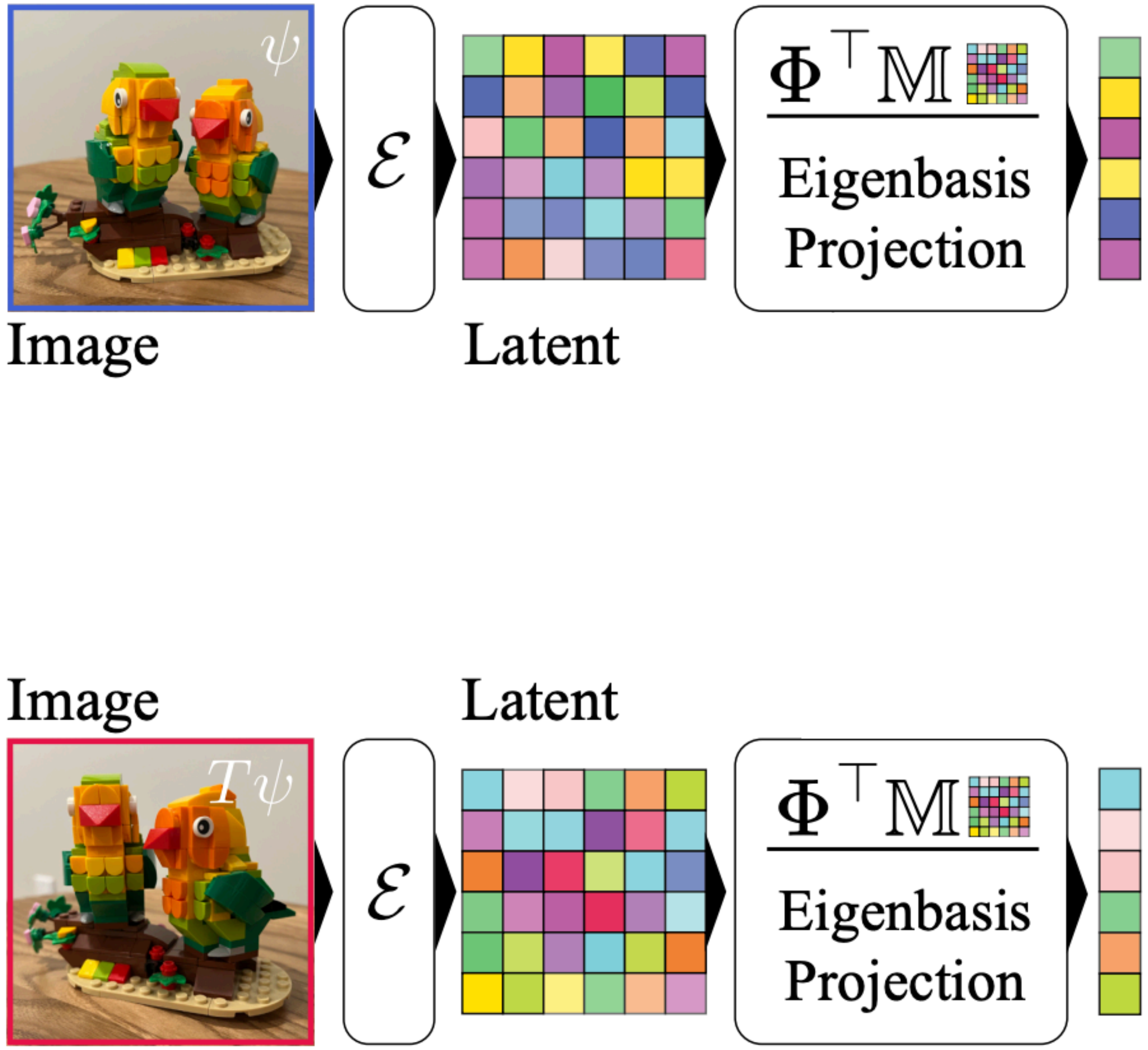


For shifts, you all know what function basis it is: The Fourier basis!

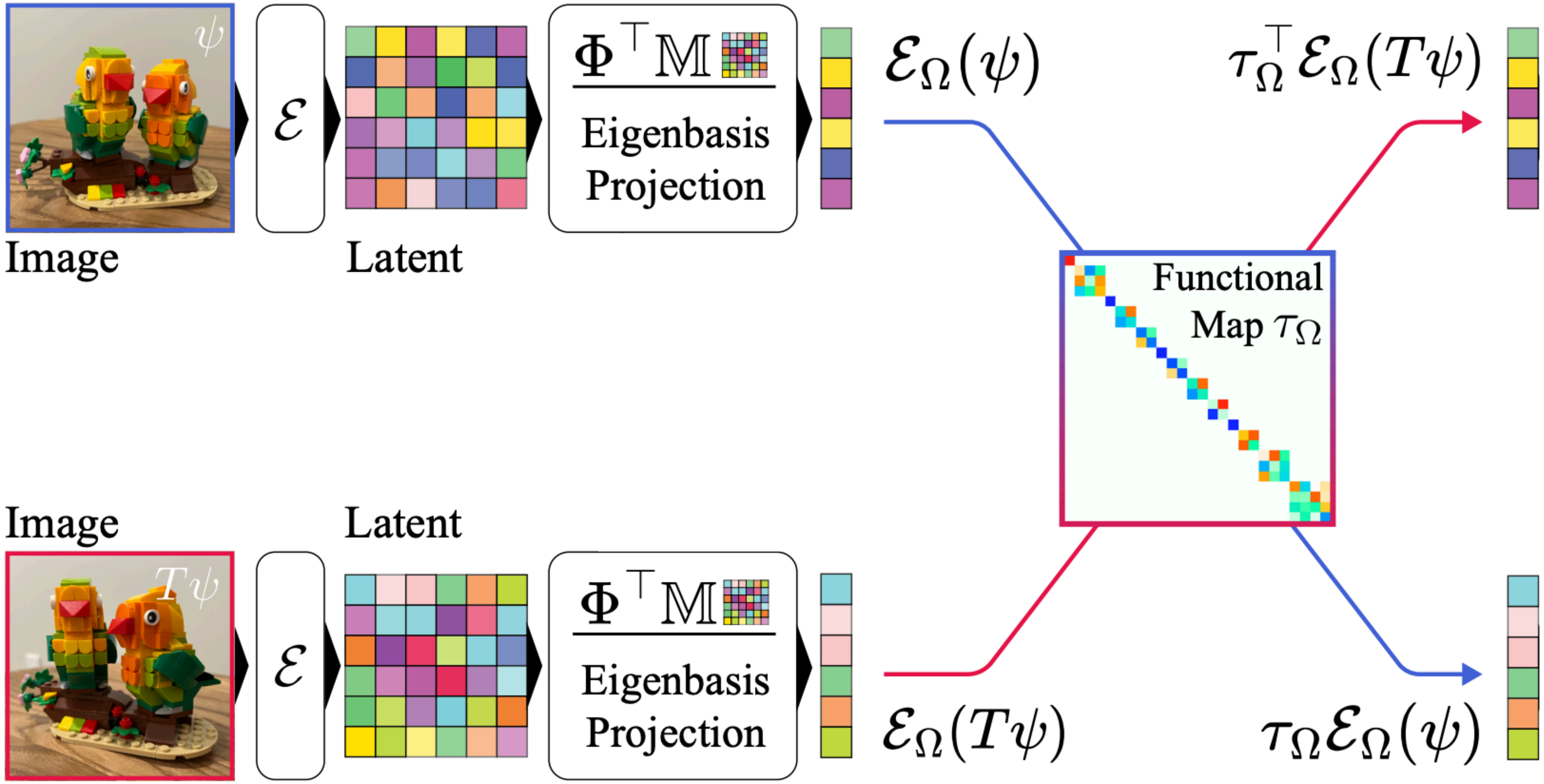
Neural Isometries



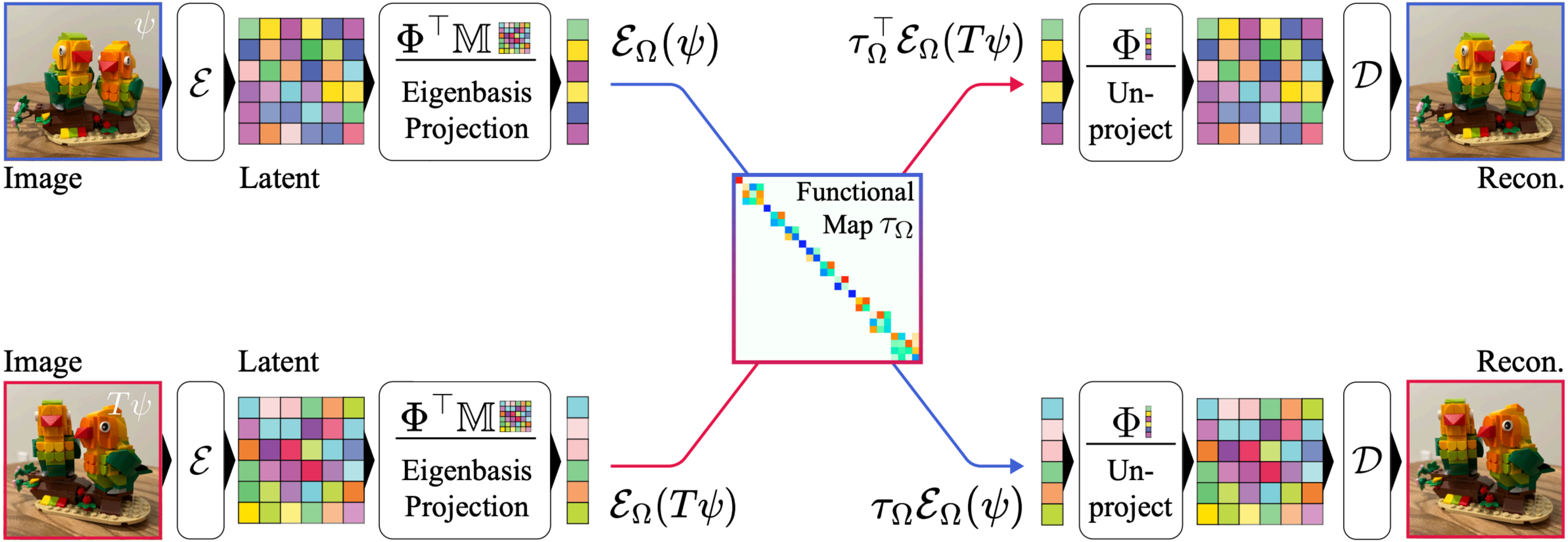
Neural Isometries



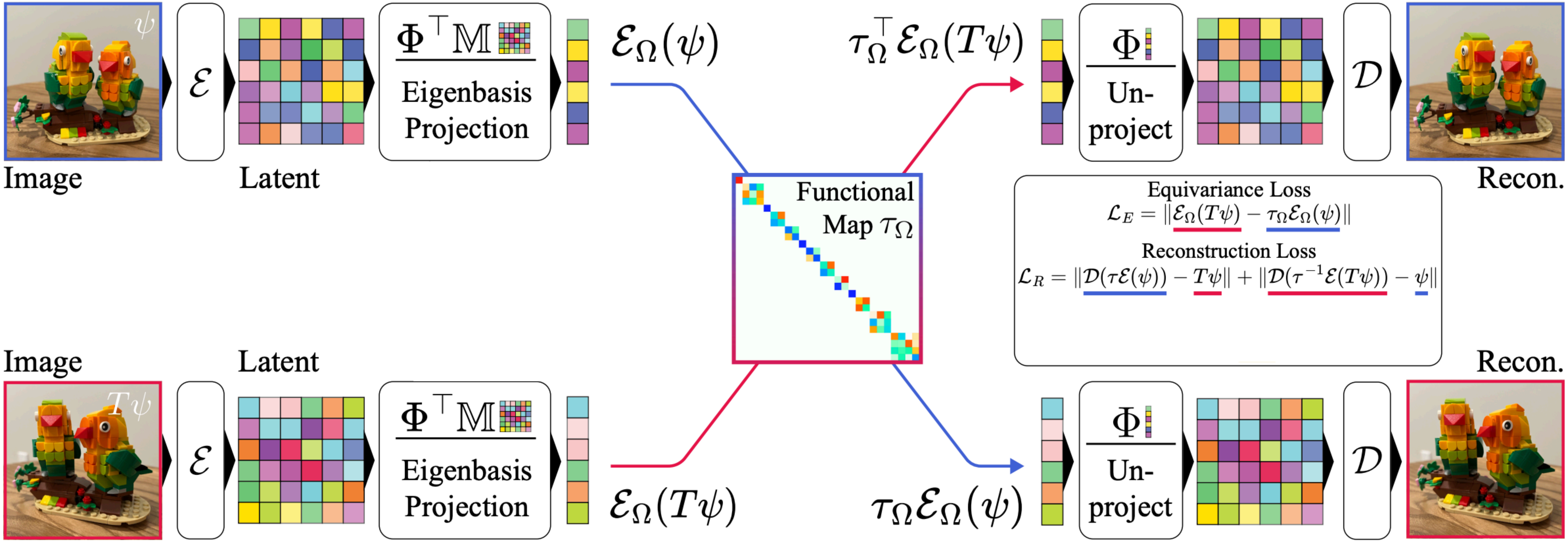
Neural Isometries



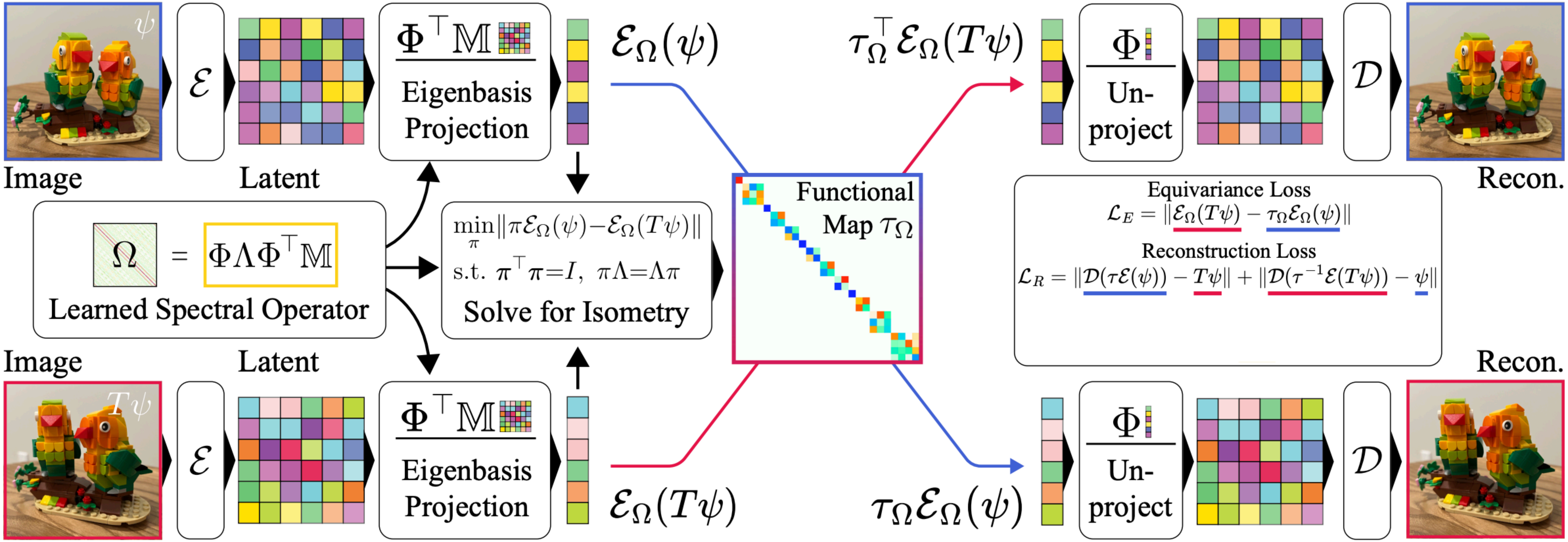
Neural Isometries



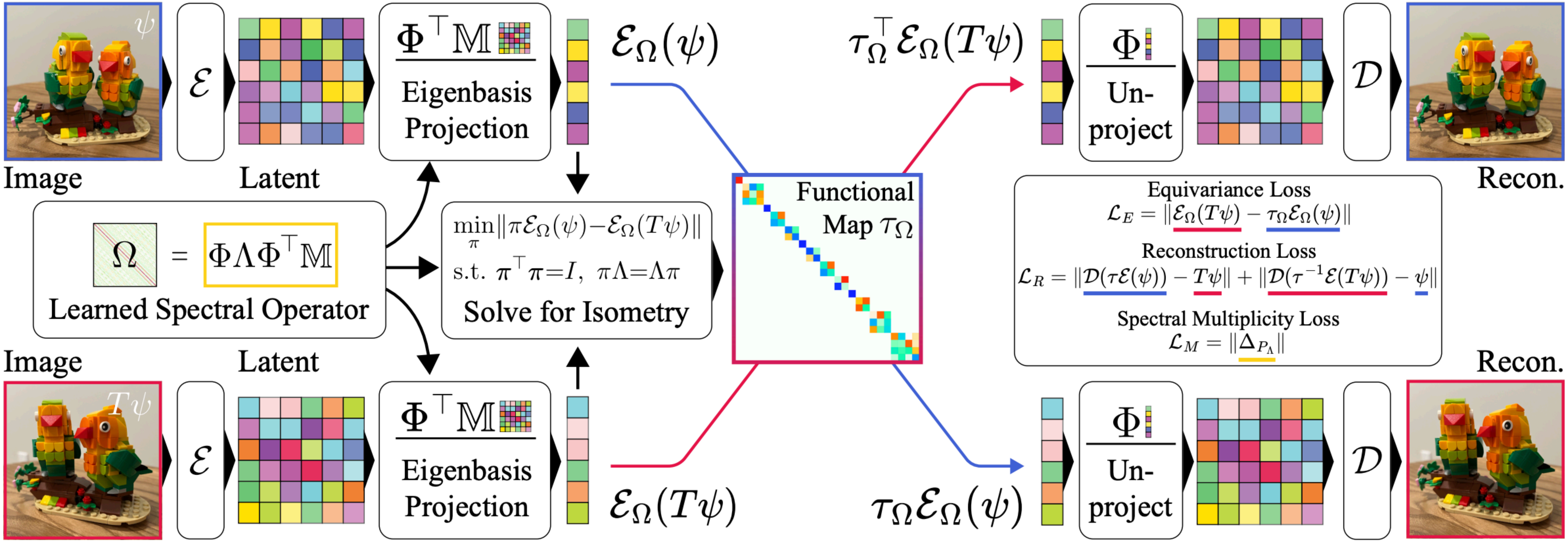
Neural Isometries



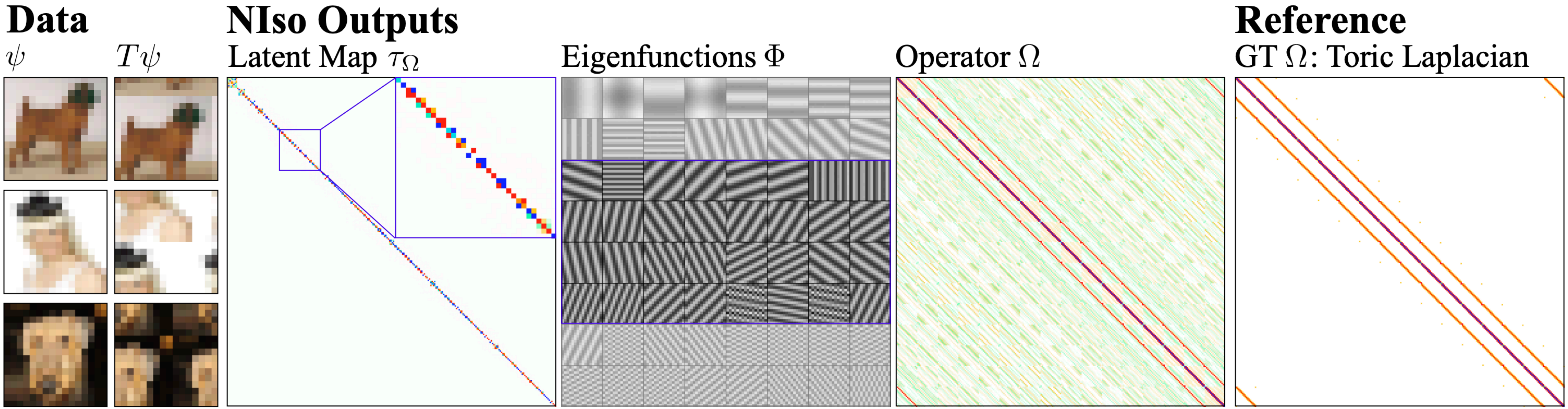
Neural Isometries



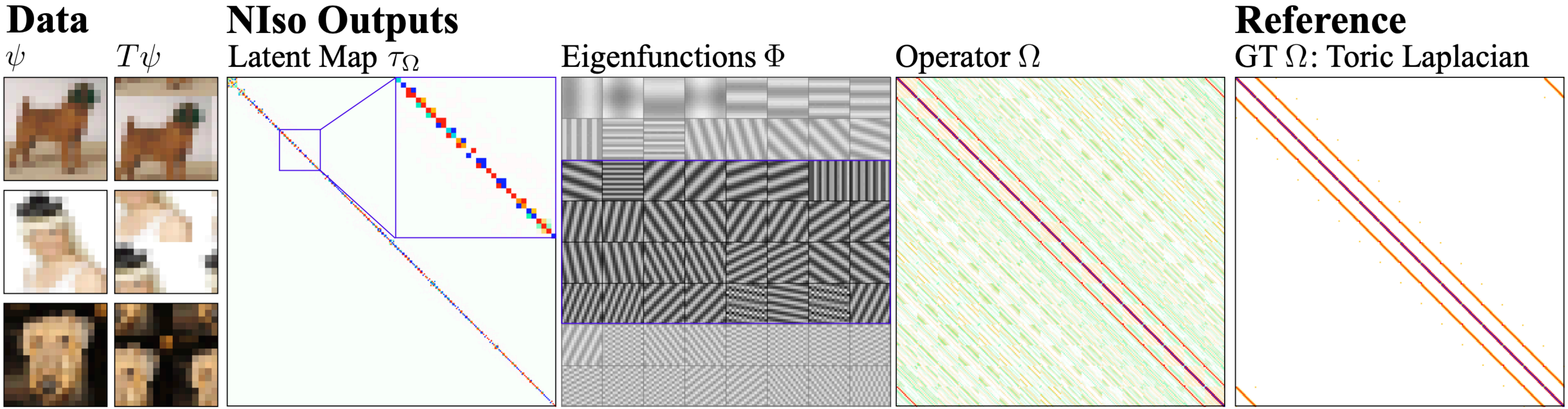
Neural Isometries



Toy example: Discovering the Fourier Transform

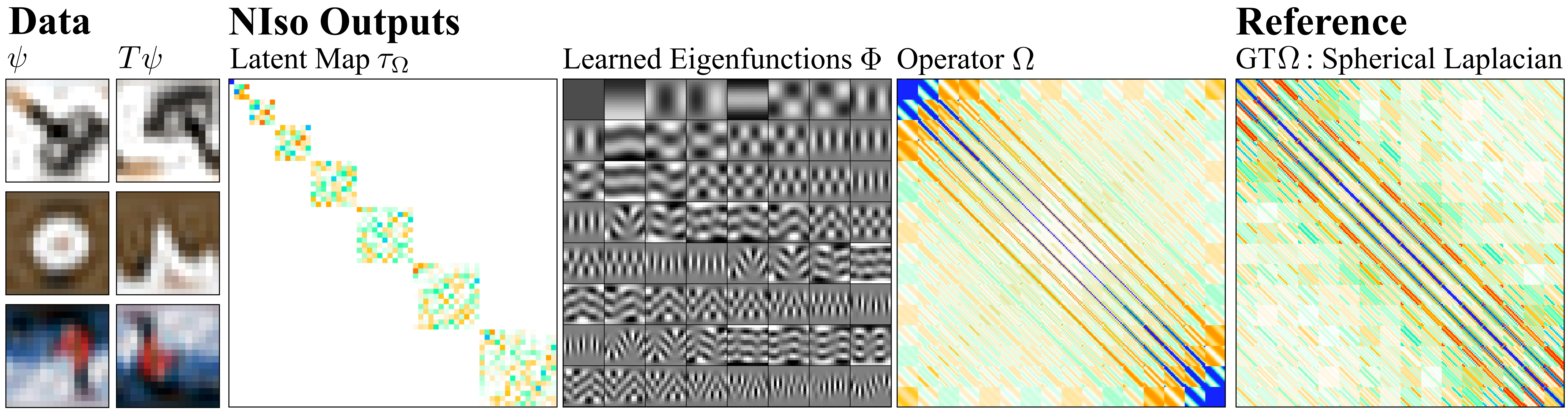


Toy example: Discovering the Fourier Transform

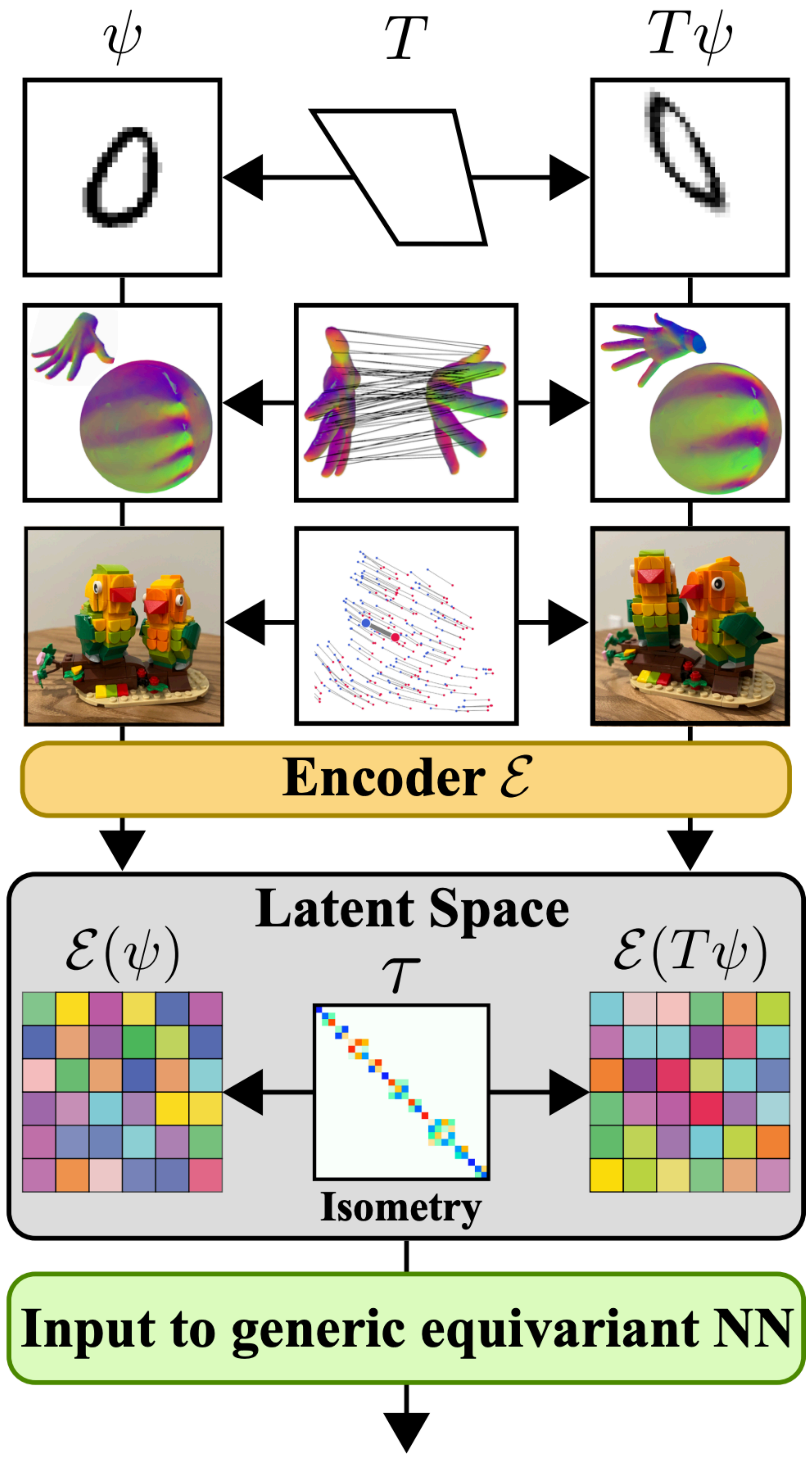


We jointly learned shift-equivariant basis functions (sines and cosines) and the shift operator purely by saying "find me a basis such that the transform is block-diagonal"

Toy example 2: Discovering the Spherical Harmonics

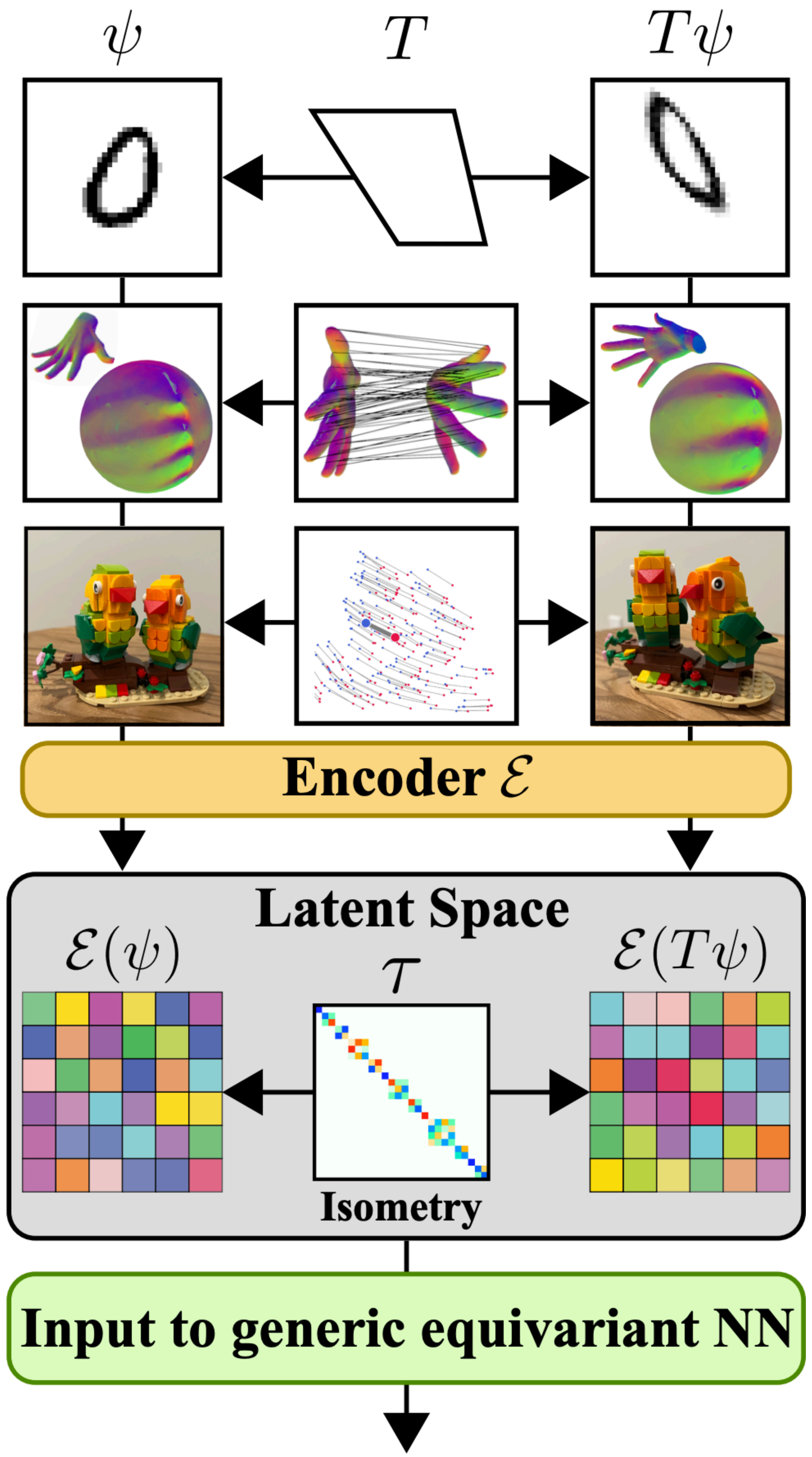


Latent Space is Equivariant to Learned Transformation!



[1] C. Deng, et. Al, “Vector Neurons: A General Framework for SO(3)-Equivariant Neural Networks”, ICCV 2021

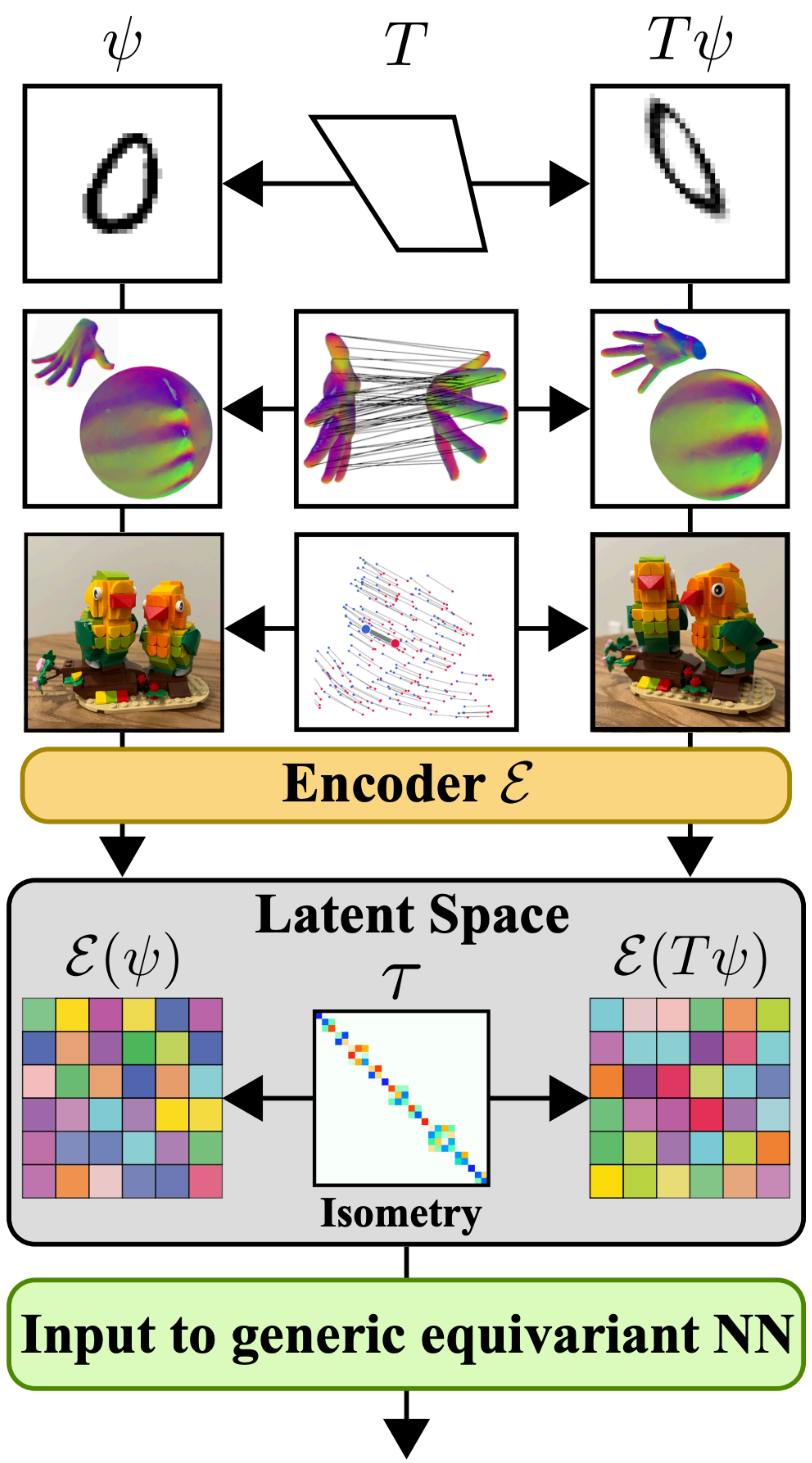
Latent Space is Equivariant to Learned Transformation!



We use a modified version of Vector Neurons [1], which are SO(3) equivariant, but can be modified for our purposes!

[1] C. Deng, et. Al, "Vector Neurons: A General Framework for SO(3)-Equivariant Neural Networks", ICCV 2021

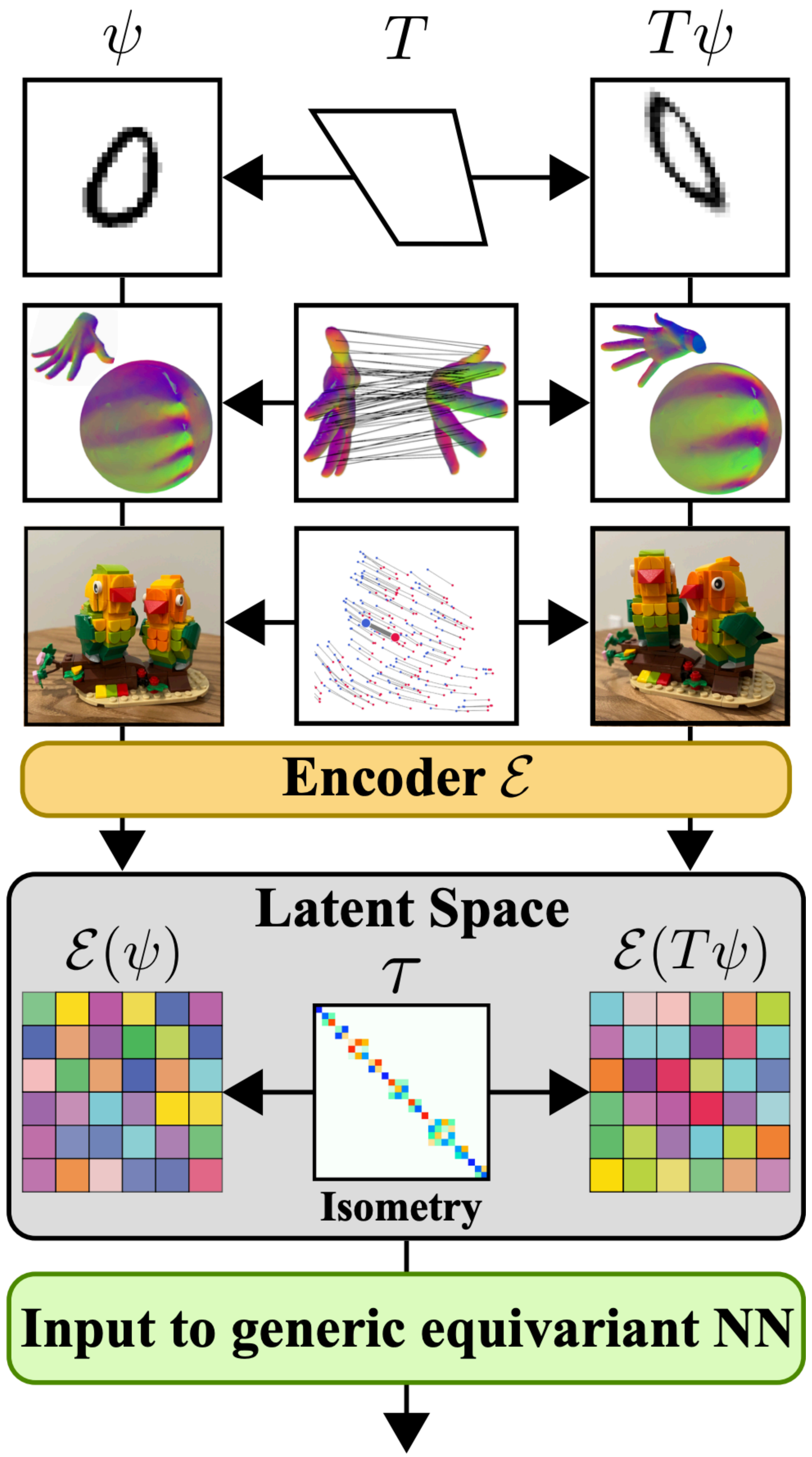
Latent Space is Equivariant to Learned Transformation!



We use a modified version of Vector Neurons [1], which are SO(3) equivariant, but can be modified for our purposes!

[1] C. Deng, et. Al, "Vector Neurons: A General Framework for SO(3)-Equivariant Neural Networks", ICCV 2021

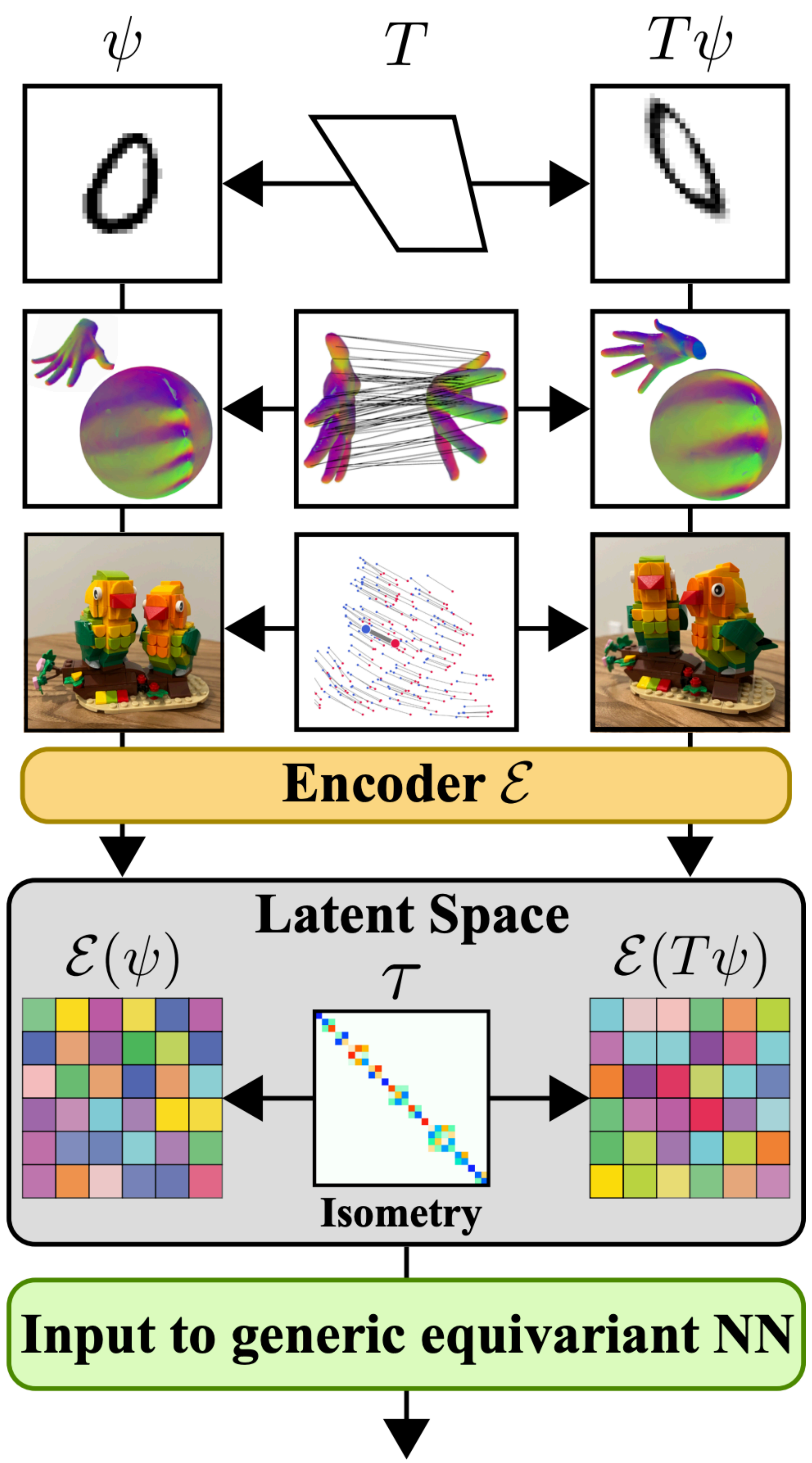
Latent Space is Equivariant to Learned Transformation!



We use a modified version of Vector Neurons [1], which are SO(3) equivariant, but can be modified for our purposes!

[1] C. Deng, et. Al, "Vector Neurons: A General Framework for SO(3)-Equivariant Neural Networks", ICCV 2021

Latent Space is Equivariant to Learned Transformation!

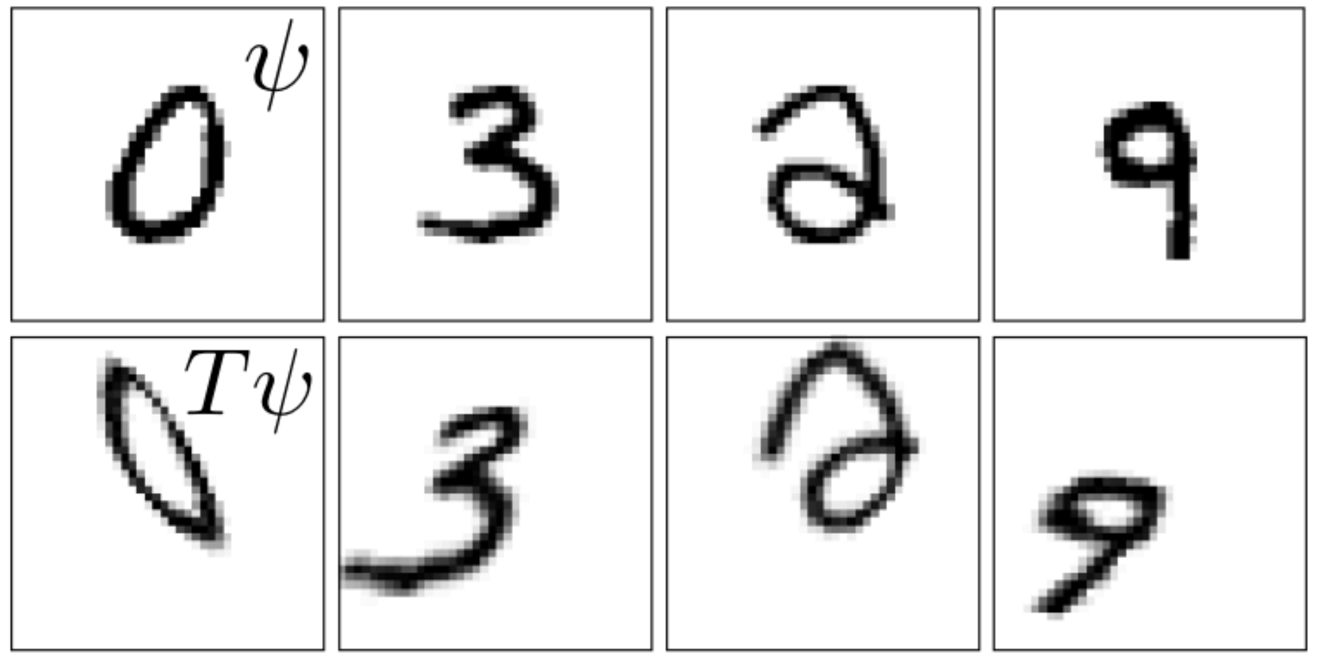


We use a modified version of Vector Neurons [1], which are SO(3) equivariant, but can be modified for our purposes!

[1] C. Deng, et. Al, "Vector Neurons: A General Framework for SO(3)-Equivariant Neural Networks", ICCV 2021

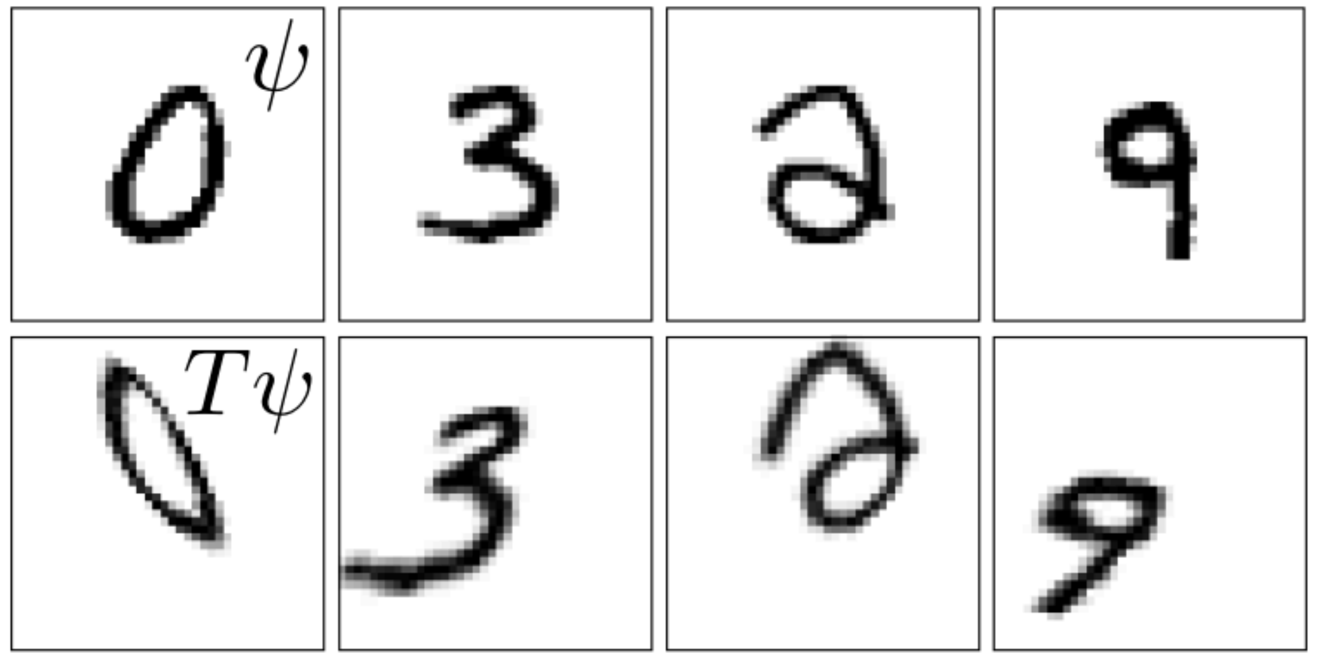
Performs on Par With Hand-Crafted Equivariant Architectures!

Homographic MNIST



Performs on Par With Hand-Crafted Equivariant Architectures!

Homographic MNIST

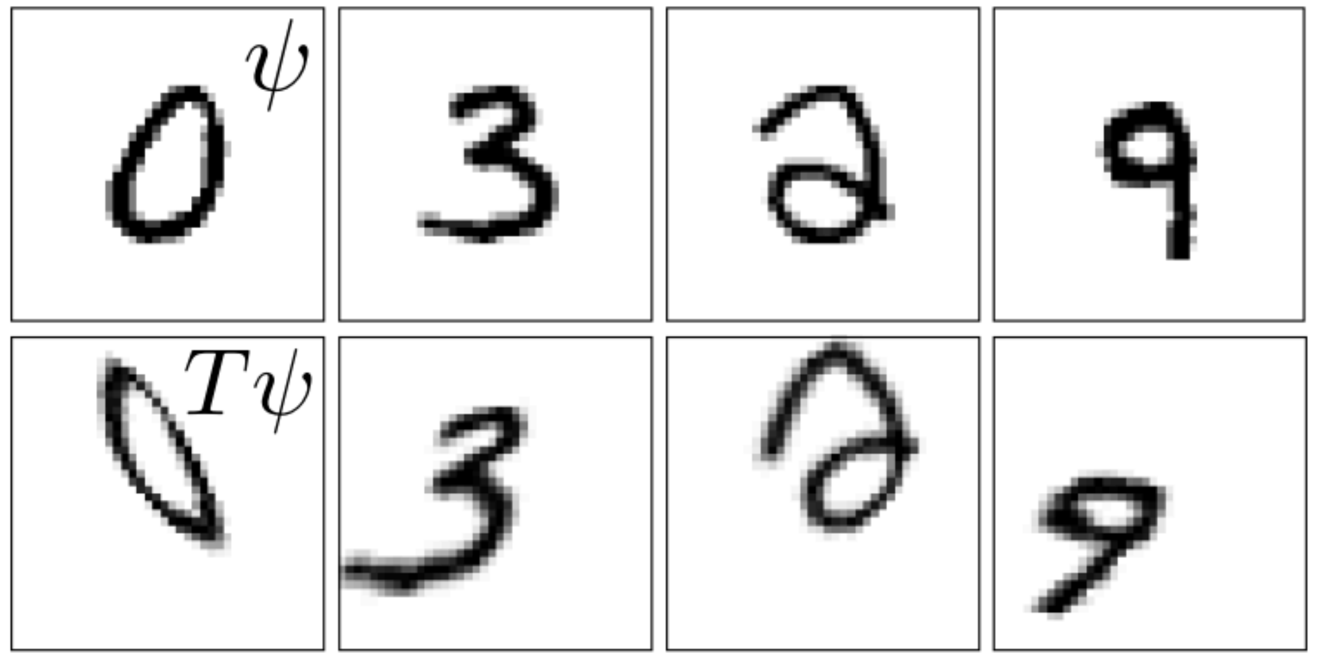


	Acc.
NIso	92.52 ($\pm$ 0.91)
w/ triplet	97.38 ($\pm$ 0.23)
w/o $\mathcal{L}_E$	77.30 ($\pm$ 2.56)
w/o $\mathcal{L}_M$	45.27 ($\pm$ 1.20)
NFT [35]	41.93 ($\pm$ 0.84)
w/ triplet	67.15 ($\pm$ 1.10)
AE Baseline	46.64 ($\pm$ 0.41)
homConv [6]	95.71 ($\pm$ 0.09)
LieDecomp [43]	**98.30** ($\pm$ **0.10**)

Table 1: **Hom. MNIST.**

Performs on Par With Hand-Crafted Equivariant Architectures!

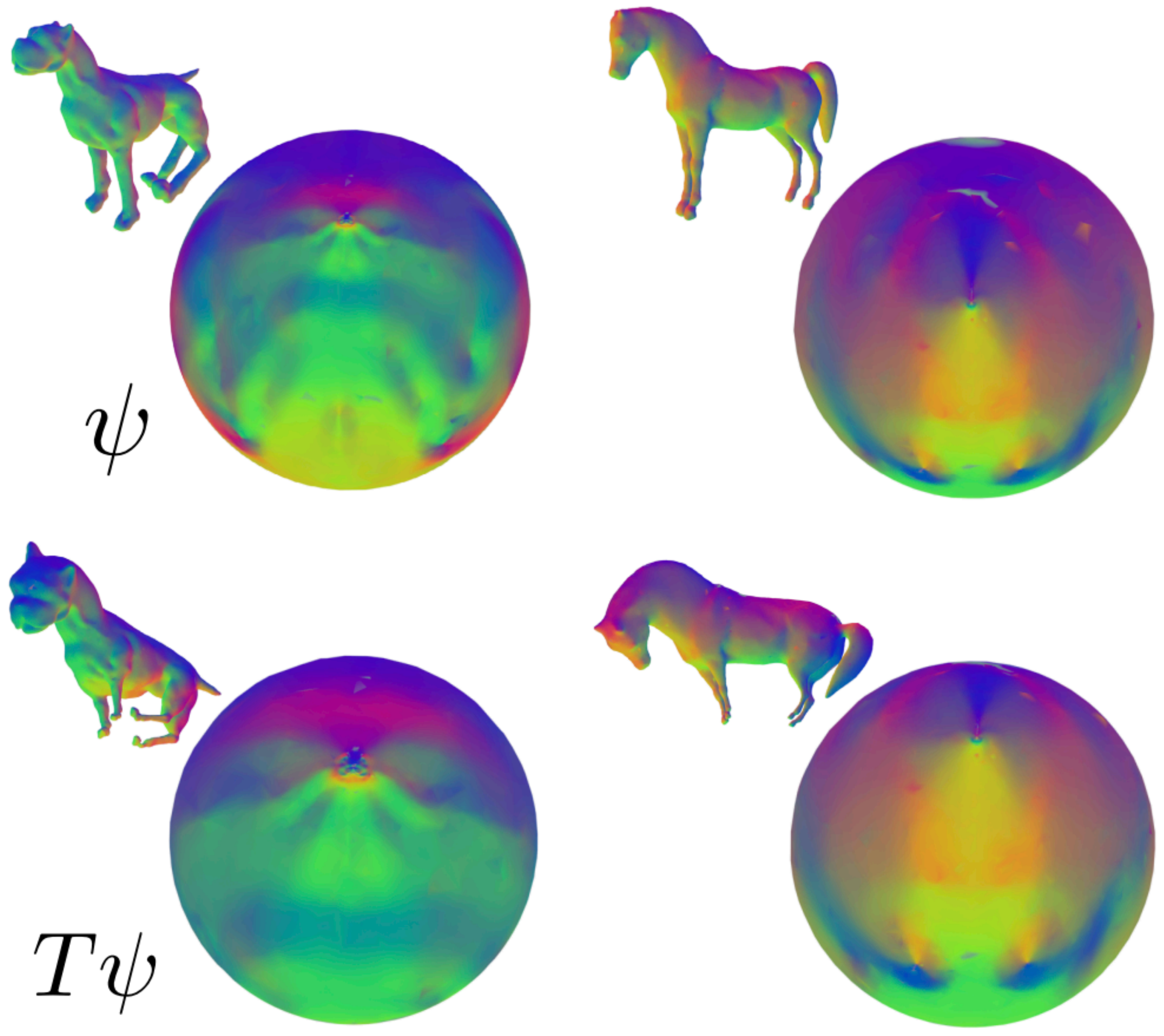
Homographic MNIST



	Acc.
NIso	92.52 ($\pm$ 0.91)
w/ triplet	97.38 ($\pm$ 0.23)
w/o $\mathcal{L}_E$	77.30 ($\pm$ 2.56)
w/o $\mathcal{L}_M$	45.27 ($\pm$ 1.20)
NFT [35]	41.93 ($\pm$ 0.84)
w/ triplet	67.15 ($\pm$ 1.10)
AE Baseline	46.64 ($\pm$ 0.41)
homConv [6]	95.71 ($\pm$ 0.09)
LieDecomp [43]	**98.30** ($\pm$ **0.10**)

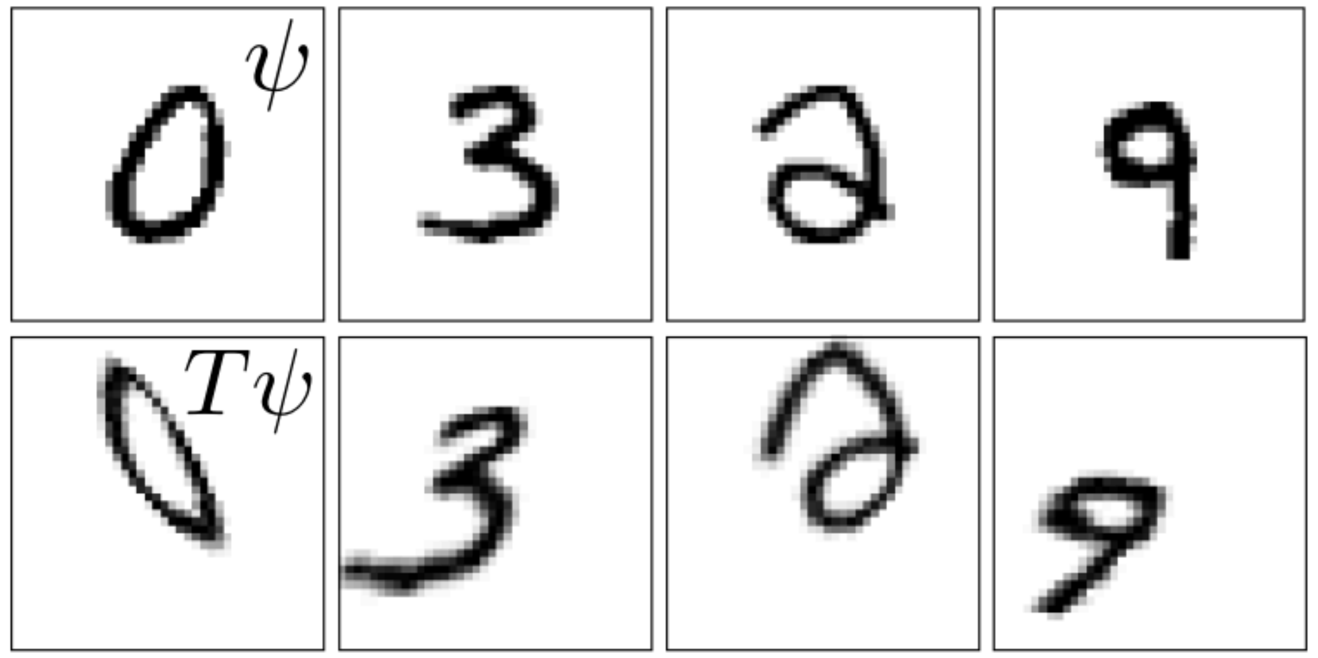
Table 1: **Hom. MNIST.**

Conformal Shape Classification



Performs on Par With Hand-Crafted Equivariant Architectures!

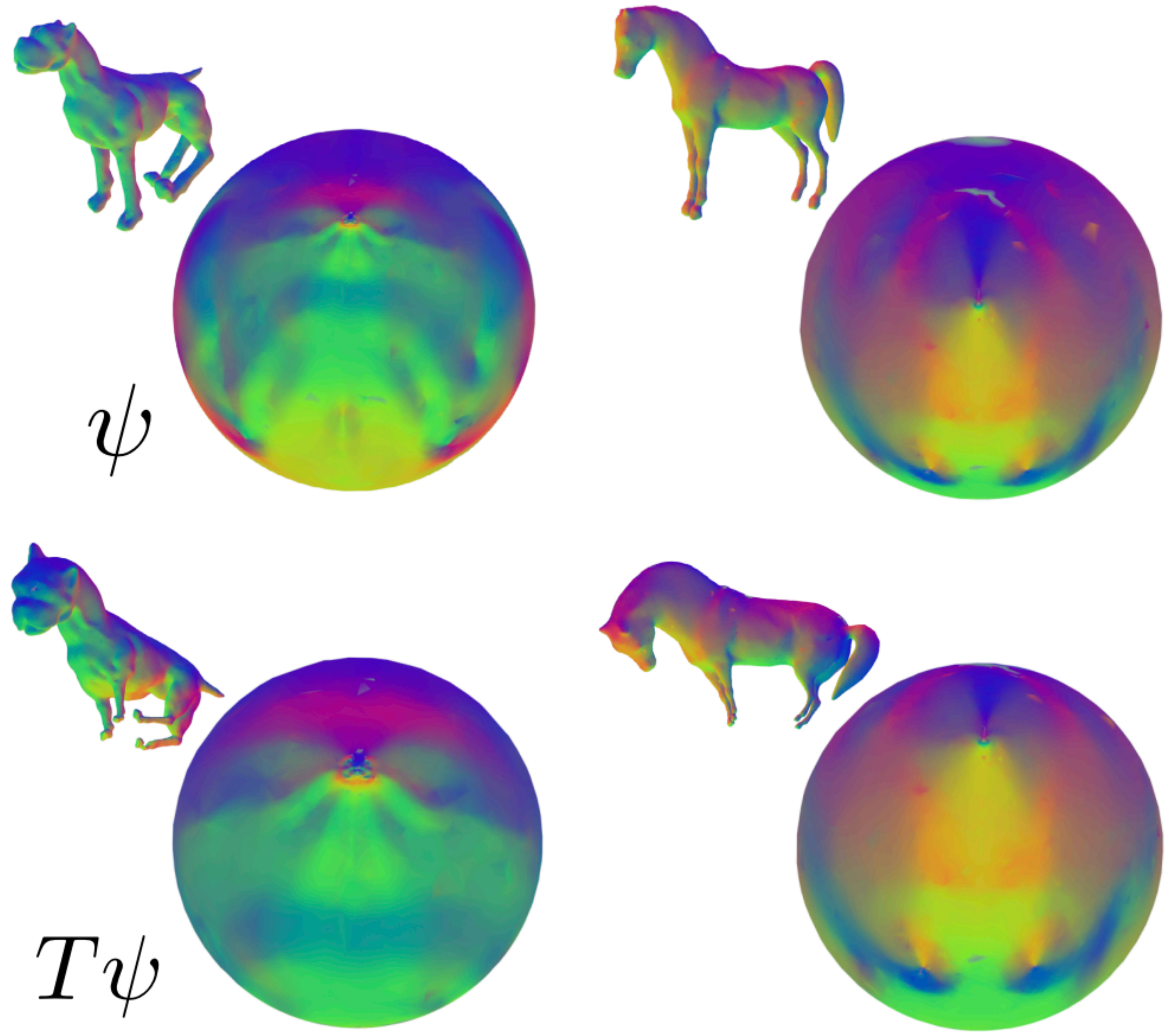
Homographic MNIST



	Acc.
NIso	92.52 (± 0.91)
w/ triplet	97.38 (± 0.23)
w/o $\mathcal{L}_E$	77.30 (± 2.56)
w/o $\mathcal{L}_M$	45.27 (± 1.20)
NFT [35]	41.93 (± 0.84)
w/ triplet	67.15 (± 1.10)
AE Baseline	46.64 (± 0.41)
homConv [6]	95.71 (± 0.09)
LieDecomp [43]	**98.30 (± 0.10)**

Table 1: **Hom. MNIST.**

Conformal Shape Classification



	Acc.
NIso	**90.26 (± 1.27)**
NFT [35]	83.24 (± 2.03)
AE Baseline	63.76 (± 2.47)
MC [7]	86.5

Table 2: **Conf. SHREC '11.**

Neural Isometries: Taming Transformations for Equivariant ML

Thomas W. Mitchel
PlayStation
tommy.mitchel@sony.com

Michael Taylor
PlayStation
mike.taylor@sony.com

Vincent Sitzmann
MIT
sitzmann@mit.edu

https://www.scenerepresentations.org/publications/niso/

Abstract

Real-world geometry and 3D vision tasks are replete with challenging symmetries that defy tractable analytical expression. In this paper, we introduce Neural Isometries, an autoencoder framework which learns to map the observation space to a general-purpose latent space wherein encodings are related by *isometries* whenever their corresponding observations are geometrically related in world space. Specifically, we regularize the latent space such that maps between encodings preserve a learned inner product and commute with a learned functional operator, in the same manner as rigid-body transformations commute with the Laplacian. This approach forms an effective backbone for self-supervised representation learning, and we demonstrate that a simple off-the-shelf equivariant network operating in the pre-trained latent space can achieve results on par with meticulously-engineered, handcrafted networks designed to handle complex, nonlinear symmetries. Furthermore, isometric maps capture information about the respective transformations in world space, and we show that this allows us to regress camera poses directly from the coefficients of the maps between encodings of adjacent views of a scene.

Tommy Mitchel
Collaborator and honorary group member

Summary

- The Fourier Transform does not appear out of nowhere, it is the simplest example for a change-of-basis that makes a group transformation “simple”.
- Other groups also have steerable bases, although the transformations aren’t usually as simple as for the Fourier Basis.
  - We talked about the spherical harmonics.
  - In your homework, you will re-implement “steerable filters”
- The matrices that effect the group transformations in the steerable basis are called the “irreducible representations”. They are always block-diagonal by virtue of “not mixing” the coefficients of eigenfunctions with different eigenvalues.

Acknowledegements

Erik Bekkers, Amsterdam Machine Learning Lab, University of Amsterdam
This mini-course serves as a module with the UvA Master AI course Deep Learning 2 https://uvadl2c.github.io/

Has some of the most amazing material on group theory for a computer science audience!

Hyunwoo Ryu and **Tommy Mitchel** have given me lots of feedback & contributed the steerable spherical harmonics animation!