UNIVERSITY OF AMSTERDAM

AMLAB Amsterdam Machine Learning Lab

# Group Equivariant Deep Learning

## Lecture 3 - Equivariant graph neural networks

### Lecture 3.4 - Group Theory | SO(3) irreps (Wigner-D matrices), Clebsch-Gordan TP

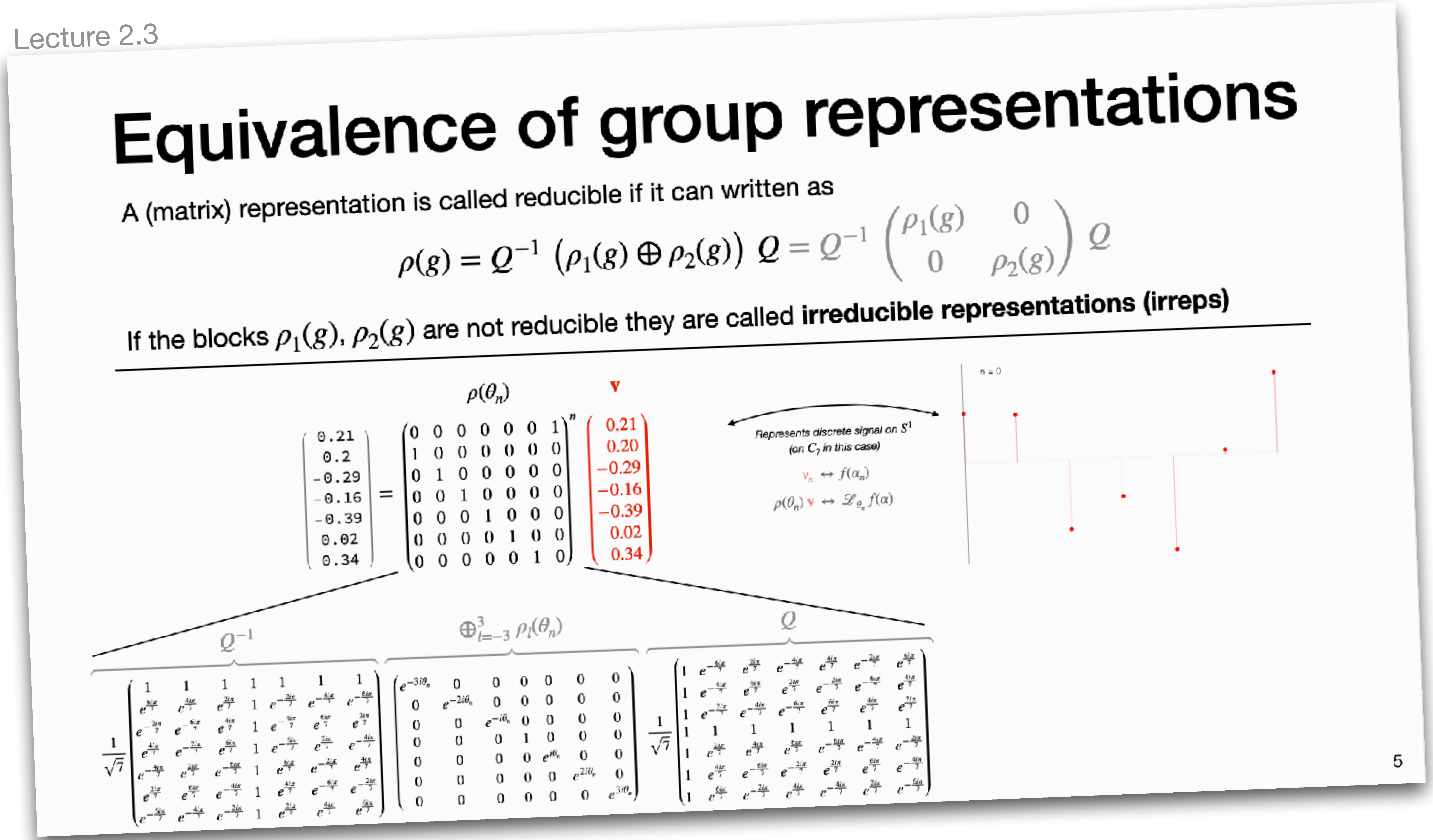
*Preliminaries for 3D steerable g-convs*

**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/

# Irreps of SO(3): Wigner-D matrices

Lecture 2.3



# Irreps of SO(3): Wigner-D matrices

*"frequency"*

Wigner-D matrices of type $l$ are the irreducible matrix representations of $SO(3)$.
We will denote these $(2l+1) \times (2l+1)$ dimensional matrices with $\mathbf{D}^{(l)}(\mathbf{R})$

$$\mathbf{D}^{(l)}(\mathbf{R}) \;=\; [\, D^{(l)}_{mn}(\mathbf{R}) \,]^{l}_{m,n=-l}$$

Wigner-D functions
form an orthogonal basis for $\mathbb{L}_2(SO(3))$!!!

The $(2l+1)$ dimensional vector space on which $\mathbf{D}^{(l)}(\mathbf{R})$ acts will be called a steerable vector space of type $l$ and denoted with $V_l = \mathbb{R}^{2l+1}$. A vector $\mathbf{v} \in V_l$ will be called a type-$l$ vector.

# Irreps of SO(3): Wigner-D matrices

Example (type 0): A type-0 vector $v \in V_0$ is just a scalar the trivially transforms by a $1 \times 1$ dimensional "matrix"

$$\mathbf{D}^{(0)}(\mathbf{R})\, v = 1\, v = v$$


Example (type 1): A type-1 vector $\mathbf{v} \in V_1$ is a 3D vector (e.g. velocity, force, displacement) that transforms directly via the rotation matrix $\mathbf{R} \in SO(3)$

$$\mathbf{D}^{(1)}(\mathbf{R})\, \mathbf{v} = \mathbf{R}\, \mathbf{v}$$


$D^{(l)}_{mn}(\mathbf{R}_{\alpha,\beta,\gamma})$

$$\begin{pmatrix} \cos(\alpha)\cos(\gamma) - \sin(\alpha)\cos(\beta)\sin(\gamma) & \sin(\beta)\sin(\gamma) & \cos(\alpha)\cos(\beta)\sin(\gamma) + \sin(\alpha)\cos(\gamma) \\ \sin(\alpha)\sin(\beta) & \cos(\beta) & -\cos(\alpha)\sin(\beta) \\ \sin(\alpha)(-\cos(\beta))\cos(\gamma) - \cos(\alpha)\sin(\gamma) & \sin(\beta)\cos(\gamma) & \cos(\alpha)\cos(\beta)\cos(\gamma) - \sin(\alpha)\sin(\gamma) \end{pmatrix}$$

# Wigner-D functions: complete orthogonal basis for functions on $SO(3)$

The Wigner-D functions $D^{(l)}_{mn} : SO(3) \rightarrow \mathbb{R}$ form a complete orthogonal basis for functions on $SO(3)$.
Thus any function can be represented in such an $SO(3)$ Fourier series:

$$f(\mathbf{R}) = \sum_{l} \sum_{m=-l}^{l} \sum_{n=-l}^{l} \hat{f}^{(l)}_{mn} D^{l}_{mn}(\mathbf{R})$$

$$= \sum_{l} \operatorname{tr}\left(\hat{f}^{(l)}\, \mathbf{D}^{(l)}(\mathbf{R}^{-1})\right)$$

General form Fourier trafo on $G$ (Peter-Weyl)

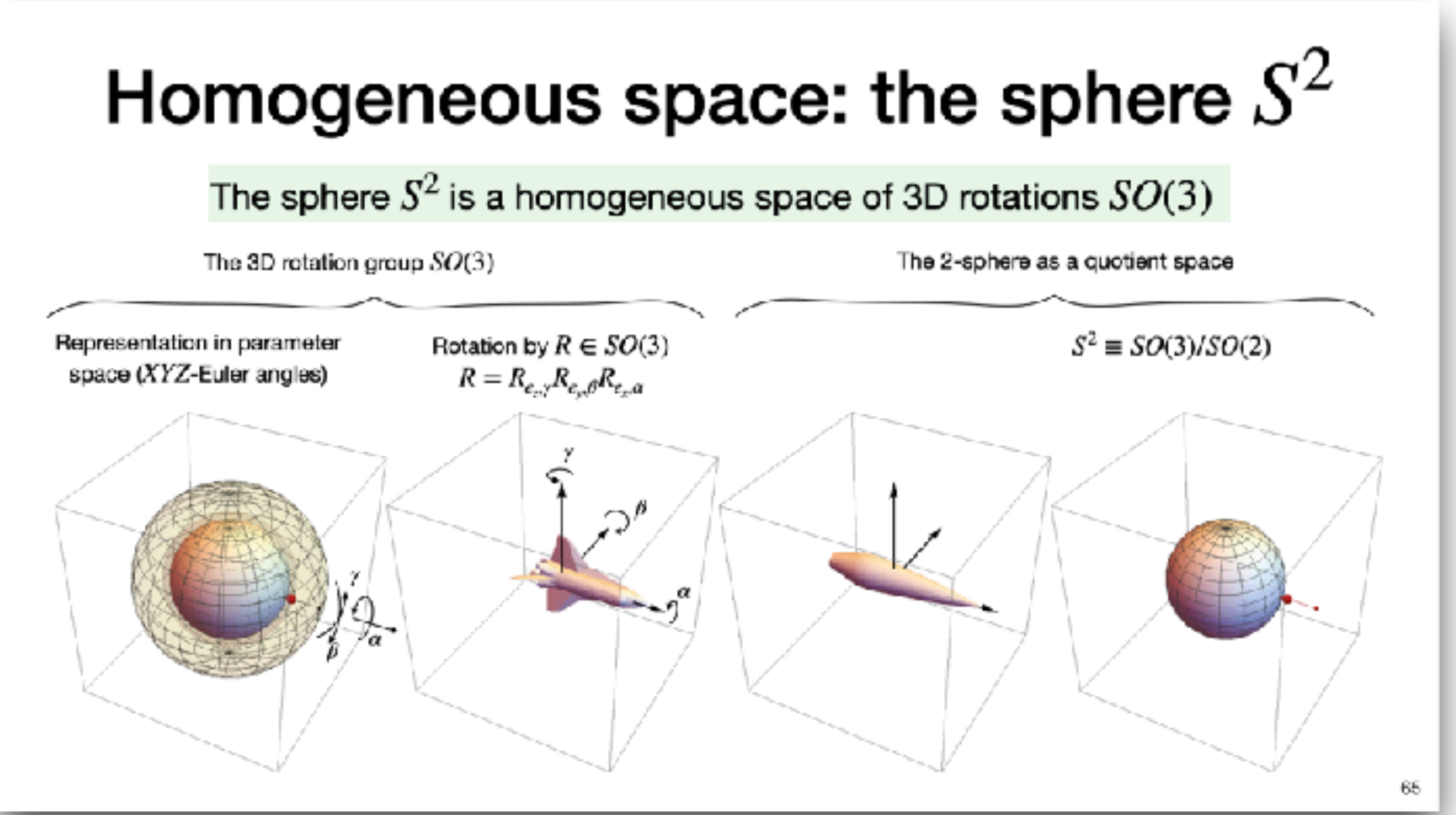
Forward $\quad [\mathcal{F}_G f]_l = \int_G f(g)\, \rho_l(g)\, \mathrm{d}g$

Inverse $\quad \mathcal{F}^{-1}[\hat{f}](g) = \sum_l d_{\rho_l} \operatorname{tr}\left[\hat{f}(\rho_l)\rho_l(g^{-1})\right]$

The central columns $D_{:0}$ is invariant to rotations $\mathbf{R}_\alpha$ around chosen reference axis (e.g. $\mathbf{e}_x$):

$$\forall_{\alpha \in [0, 2\pi)} : \quad D^{(l)}_{m0}(\mathbf{R}\,\mathbf{R}_\alpha) = D^{(l)}_{m0}(\mathbf{R})$$

These $SO(2)$-invariant functions coincide with functions on the sphere $S^2 \equiv SO(3)/SO(2)$: the spherical harmonics $Y : S^2 \rightarrow \mathbb{R}$!



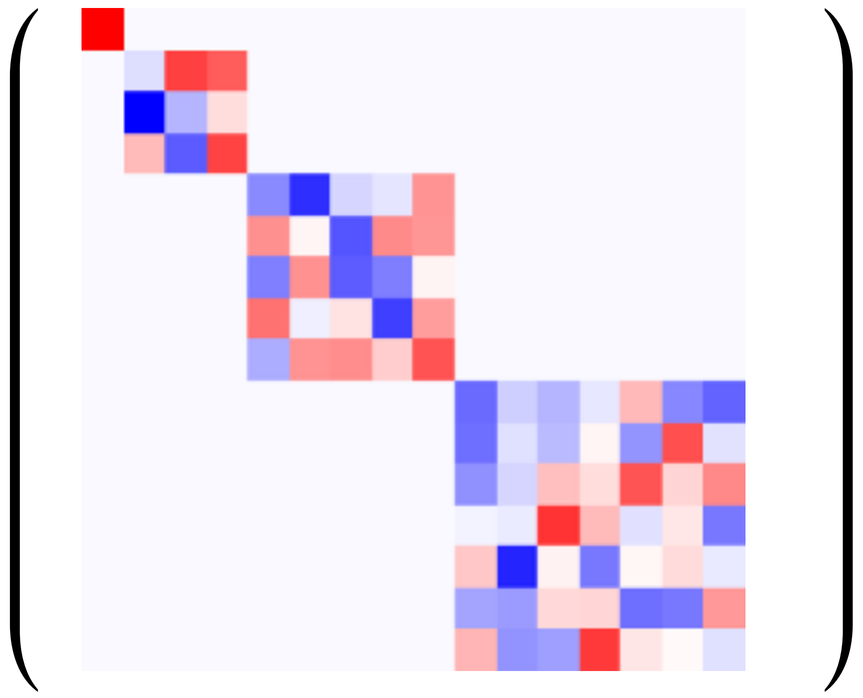
# Spherical harmonics: complete orthogonal basis for functions on $S^2$

- Solutions of Laplace equation (hence “harmonics”)

- A Fourier basis on the sphere $S^2$

- Fourier coefficients transform via block-diagonal representations

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



Spherical Harmonics

$$Y^{(l)}(\mathbf{n}_{\beta,\gamma}) \sim D^{(l)}_{m0}(\mathbf{R}_{\alpha,\beta,\gamma})$$

$$\mathbf{R}_{\alpha,\beta,\gamma} = \mathbf{R}_{e_z,\gamma}\mathbf{R}_{e_y,\beta}\mathbf{R}_{e_x,\alpha}$$
$$\mathbf{n}_{\beta,\gamma} = \mathbf{R}_{e_z,\gamma}\mathbf{R}_{e_y,\beta}\mathbf{e}_x$$

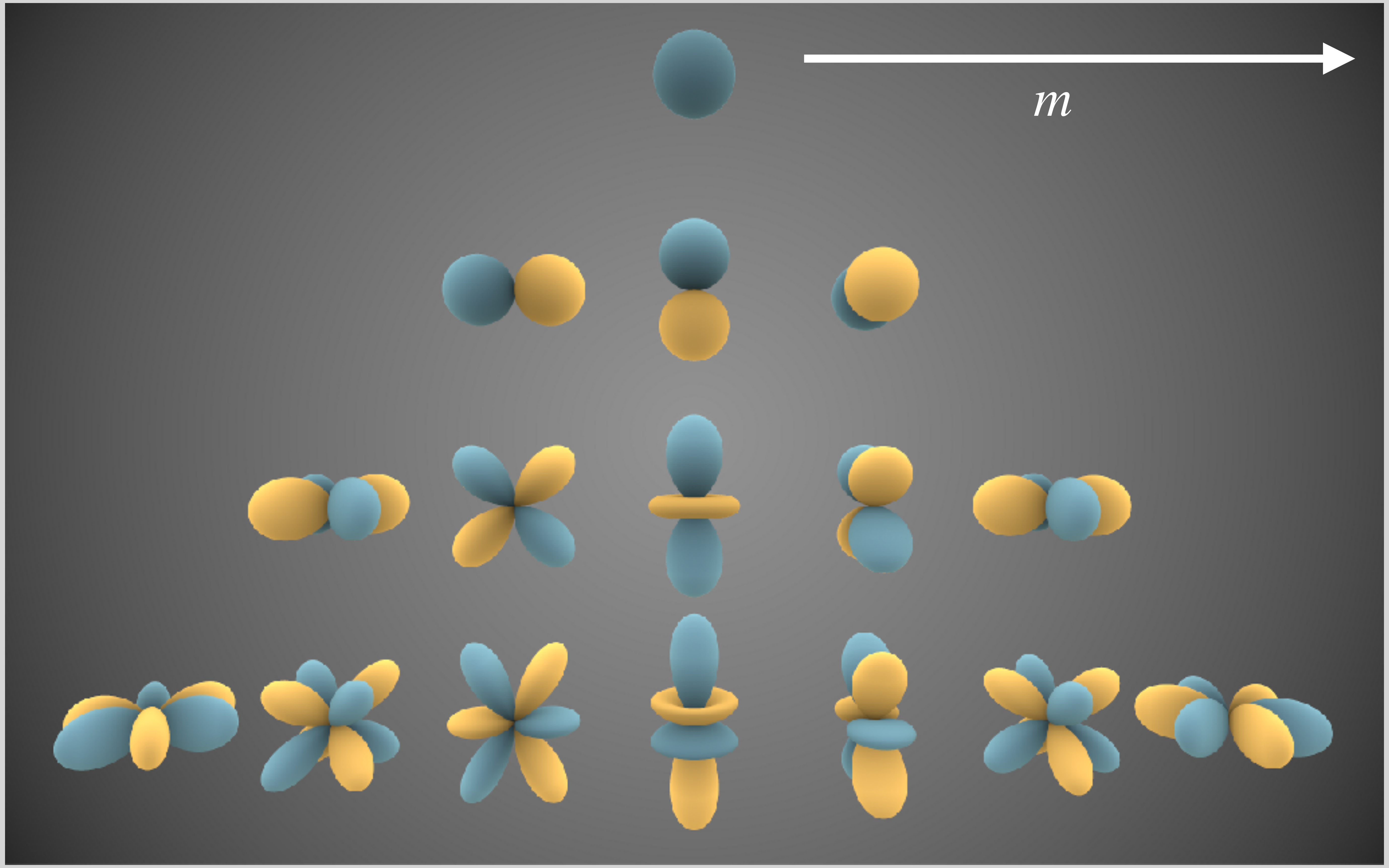
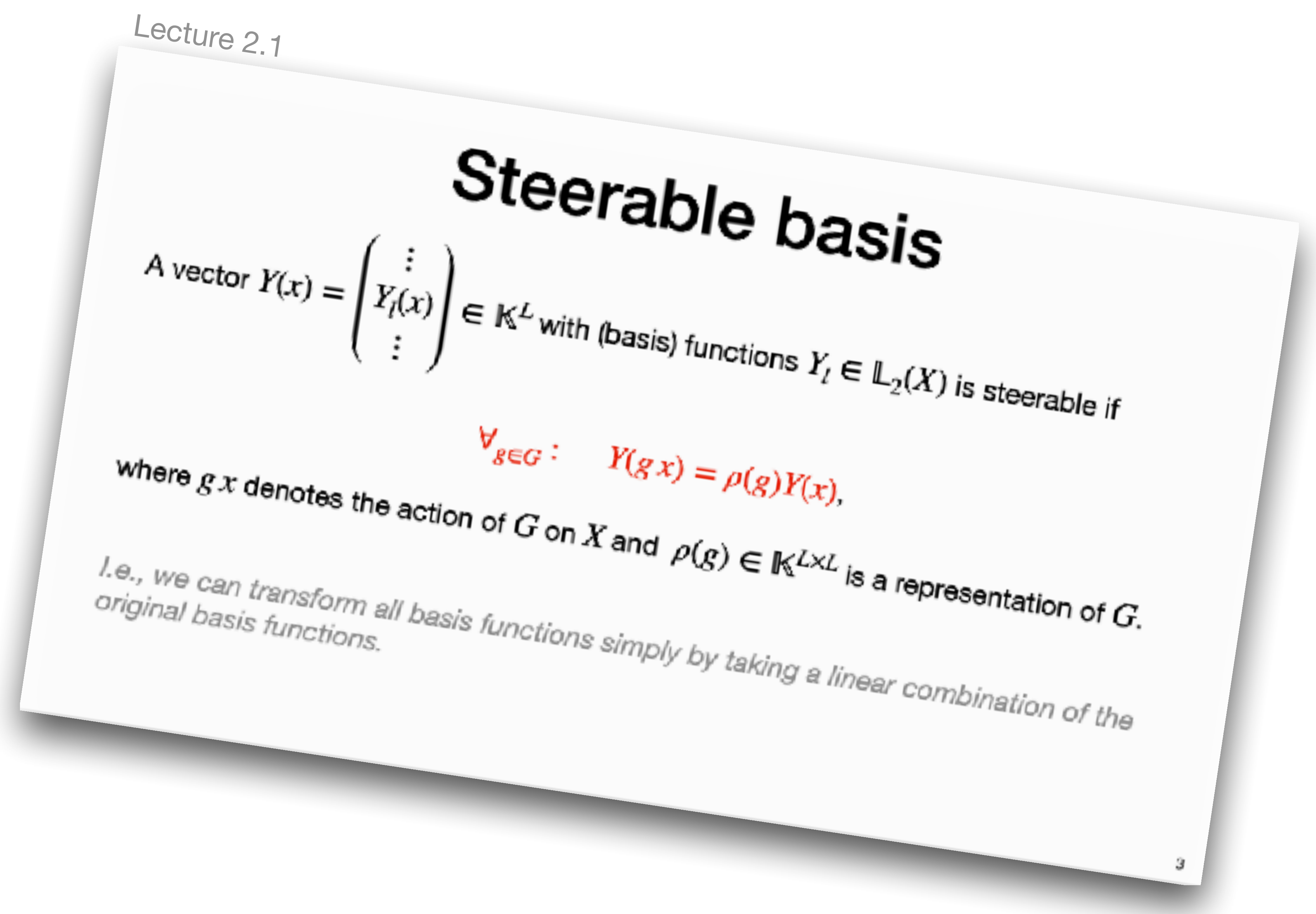



Figure from: https://en.wikipedia.org/wiki/Spherical_harmonics

# Spherical harmonics are $SO(3)$ steerable

Since $Y(\mathbf{n}_{\beta,\gamma}) \sim \mathbf{D}_{:0}(\mathbf{R}_{\alpha,\beta,\gamma})$ and $\mathbf{D}^{(l)}$ are (irreducible) representations ( $\mathbf{D}^{(l)}(\mathbf{R}\mathbf{R}') = \mathbf{D}^{(l)}(\mathbf{R})\mathbf{D}^{(l)}(\mathbf{R}')$ ) it follows

$$\forall_{\mathbf{R}\in SO(3),\mathbf{n}\in S^2}: \quad Y^{(l)}(\mathbf{R}\,\mathbf{n}) = \mathbf{D}^{(l)}(\mathbf{R})\,Y^{(l)}(\mathbf{n})$$


Lecture 2.1

## 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

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

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

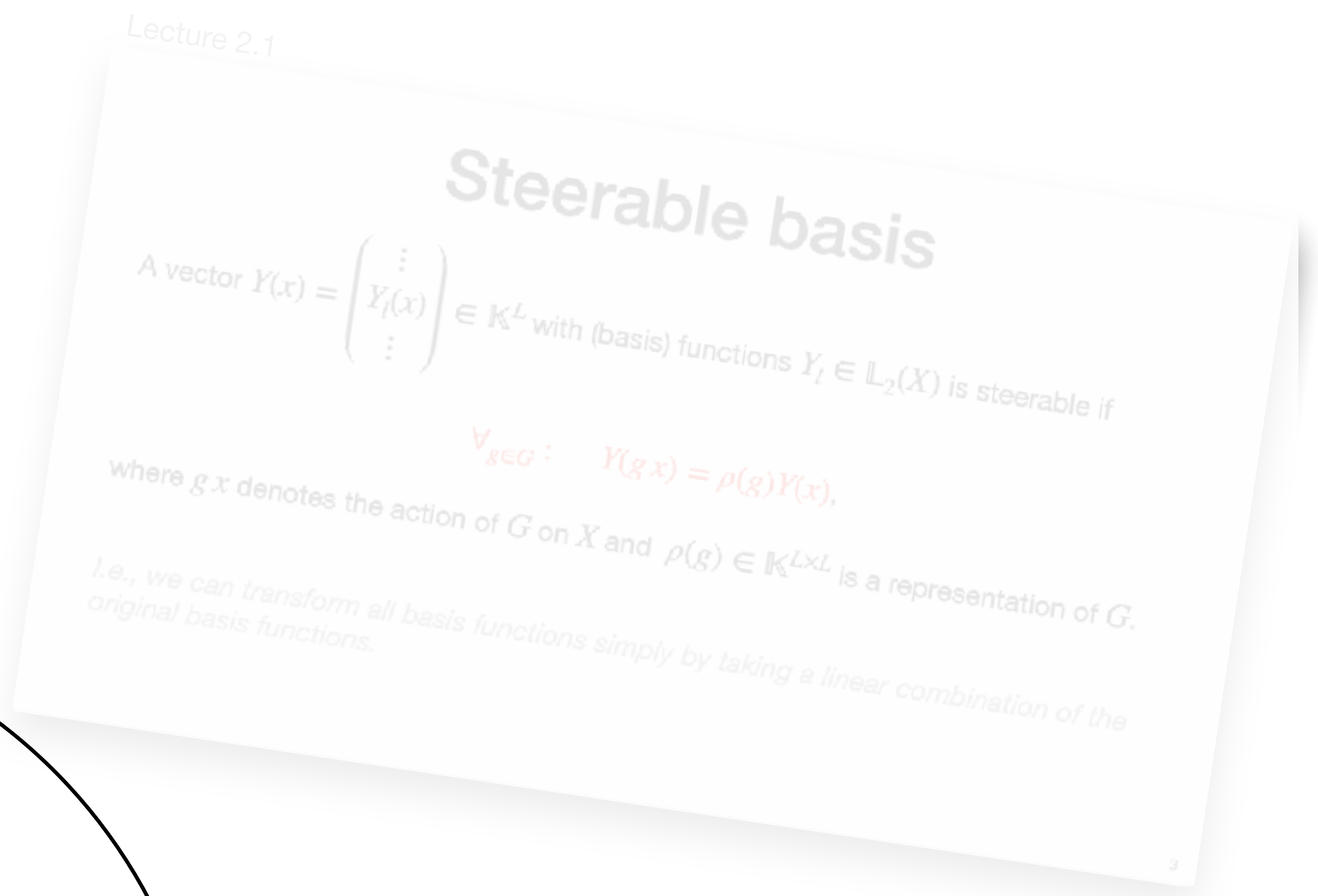
I.e., we can transform all basis functions simply by taking a linear combination of the original basis functions.

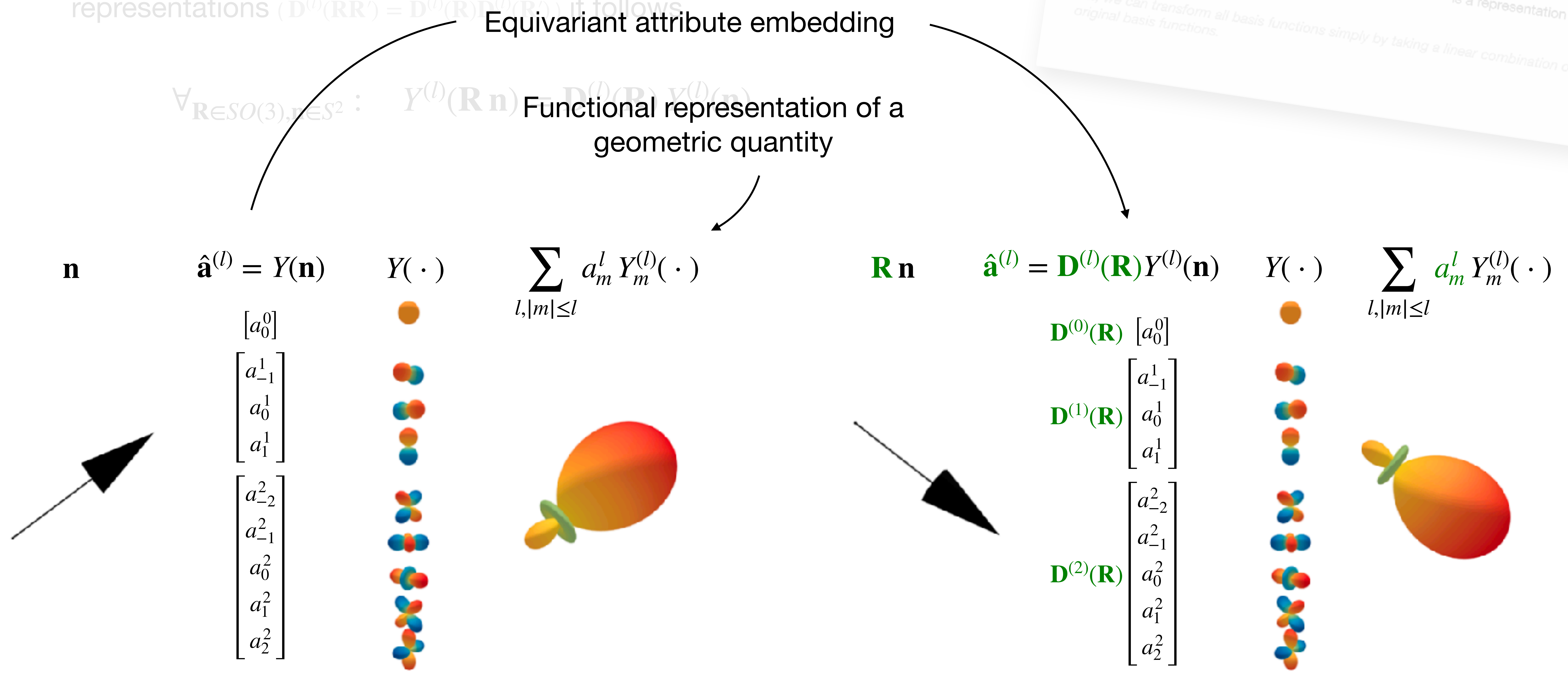
3

# Spherical harmonics are $SO(3)$ steerable

Since $Y(\mathbf{n}_{\beta,\gamma}) \sim \mathbf{D}_{:0}(\mathbf{R}_{\alpha,\beta,\gamma})$ and $\mathbf{D}^{(l)}$ are (irreducible) representations ($\mathbf{D}^{(l)}(\mathbf{R}\mathbf{R}') = \mathbf{D}^{(l)}(\mathbf{R})\mathbf{D}^{(l)}(\mathbf{R}')$) it follows

$$\forall_{\mathbf{R}\in SO(3),\mathbf{n}\in S^2}: \quad Y^{(l)}(\mathbf{R}\,\mathbf{n}) = \mathbf{D}^{(l)}(\mathbf{R})\,Y^{(l)}(\mathbf{n})$$



# Clebsch-Gordan Tensor Product

Consider the tensor product of two steerable vectors $\mathbf{a} \in V_{l_a}$ and $\mathbf{b} \in V_{l_b}$

$$\mathbf{a} \otimes \mathbf{b} = \mathbf{a}\mathbf{b}^T = \begin{pmatrix} a_1 b_1 & a_1 b_2 & \cdots \\ a_2 b_1 & a_2 b_2 & \\ \vdots & & \ddots \end{pmatrix}$$

The tensor product rotates via

( $\mathbf{a} \mapsto \mathbf{D}^{(l_a)}(\mathbf{R})\mathbf{a}$ , $\mathbf{b} \mapsto \mathbf{D}^{(l_b)}(\mathbf{R})\mathbf{b}$ )

$$\mathbf{a} \otimes \mathbf{b} \quad \mapsto \quad \mathbf{D}^{(l_a)}(\mathbf{R})\, \mathbf{a} \otimes \mathbf{b}\, \mathbf{D}^{(l_b)}(\mathbf{R})^T$$

Vectorized tensor products are steerable via:

( using identity $\mathrm{vec}(\mathbf{A}\mathbf{X}\mathbf{B}) = (\mathbf{B}^T \otimes \mathbf{A})\mathrm{vec}(\mathbf{X})$ )

$$\mathrm{vec}(\mathbf{a} \otimes \mathbf{b}) \quad \mapsto \quad (\, \mathbf{D}^{(l_b)}(\mathbf{R}^{-1}) \otimes \mathbf{D}^{(l_a)}(\mathbf{R})\, )\ \mathrm{vec}(\mathbf{a} \otimes \mathbf{b})$$

Its representation is block-diagonalizable:

$$\mathbf{Q}^{-1} \begin{pmatrix} \mathbf{D}^{l_1}(\mathbf{R}) & 0 & 0 & 0 \\ 0 & \mathbf{D}^{l_2}(\mathbf{R}) & 0 & 0 \\ 0 & 0 & \mathbf{D}^{l_3}(\mathbf{R}) & 0 \\ 0 & 0 & 0 & \ddots \end{pmatrix} \mathbf{Q} \quad \mathrm{vec}(\mathbf{a} \otimes \mathbf{b})$$

# Clebsch-Gordan Tensor Product

Consider the tensor product of two steerable vectors $\mathbf{a} \in V_{l_a}$ and $\mathbf{b} \in V_{l_b}$

$$\mathbf{a} \otimes \mathbf{b} = \mathbf{a}\mathbf{b}^T = \begin{pmatrix} a_1 b_1 & a_1 b_2 & \cdots \\ a_2 b_1 & a_2 b_2 & \\ \vdots & & \ddots \end{pmatrix}$$

The tensor product rotates via

( $\mathbf{a} \mapsto \mathbf{D}^{(l_a)}(\mathbf{R})\mathbf{a}$ , $\mathbf{b} \mapsto \mathbf{D}^{(l_b)}(\mathbf{R})\mathbf{b}$ )

$$\mathbf{a} \otimes \mathbf{b} \quad \mapsto \quad \mathbf{D}^{(l_a)}(\mathbf{R})\, \mathbf{a} \otimes \mathbf{b}\, \mathbf{D}^{(l_b)}(\mathbf{R})^T$$

Vectorized tensor products are steerable via:

( using identity $\mathrm{vec}(\mathbf{A}\mathbf{X}\mathbf{B}) = (\mathbf{B}^T \otimes \mathbf{A})\mathrm{vec}(\mathbf{X})$ )

$$\mathrm{vec}(\mathbf{a} \otimes \mathbf{b}) \quad \mapsto \quad (\, \mathbf{D}^{(l_b)}(\mathbf{R}^{-1}) \otimes \mathbf{D}^{(l_a)}(\mathbf{R})\, )\ \mathrm{vec}(\mathbf{a} \otimes \mathbf{b})$$

Its representation is block-diagonalizable:

$$\begin{pmatrix} \mathbf{D}^{l_1}(\mathbf{R}) & 0 & 0 & 0 \\ 0 & \mathbf{D}^{l_2}(\mathbf{R}) & 0 & 0 \\ 0 & 0 & \mathbf{D}^{l_3}(\mathbf{R}) & 0 \\ 0 & 0 & 0 & \ddots \end{pmatrix} \quad \mathrm{vec}(\mathbf{a} \otimes_{cg} \mathbf{b})$$

Clebsch-Gordan tensor product
includes change of basis!

# Clebsch-Gordan Tensor Product

Consider two steerable vectors $\mathbf{v}^{(l_1)} = \begin{pmatrix} \vdots \\ v^{(l_1)}_{m_1} \\ \vdots \end{pmatrix} \in V_{l_1}$ and $\mathbf{v}^{(l_2)} = \begin{pmatrix} \vdots \\ v^{(l_2)}_{m_2} \\ \vdots \end{pmatrix} \in V_{l_2}$ of type $l_1$ and $l_2$ respectively

The **Clebsch-Gordan tensor product** is defined as

$$\underbrace{(\mathbf{v}^{(l_1)} \otimes^w_{cg} \mathbf{v}^{(l_2)})^{(l)}_m}_{\text{A steerable output vector of type } l} = \sum_{m_1=-l_1}^{l_1} \sum_{m_2=-l_2}^{l_2} C^{(l,m)}_{(l_1,m_1)(l_2,m_2)} \, v^{(l_1)}_{m_1} \, v^{(l_2)}_{m_2}$$

- The Clebsch-Gordan tensor product is highly sparse (many $C^{(l,m)}_{(l_1,m_1)(l_2,m_2)} = 0$)
- In particular for all $l < |l_1 - l_2|$ and $l > l_1 + l_2$ the CG coefficients are zero.

**Familiar Examples**:

- Product of two scalars $(l_1 = 0,\, l_2 = 0,\, l = 0)$
- The scalar-vector product $(l_1 = 0,\, l_2 = 1,\, l = 1)$
- The dot product $(l_1 = 1,\, l_2 = 1,\, l = 0)$
- The cross product $(l_1 = 1,\, l_2 = 1,\, l = 1)$

# Clebsch-Gordan Tensor Product with the e3nn library

https://docs.e3nn.org/en/stable/

» Euclidean neural networks

Edit on GitHub

## Euclidean neural networks

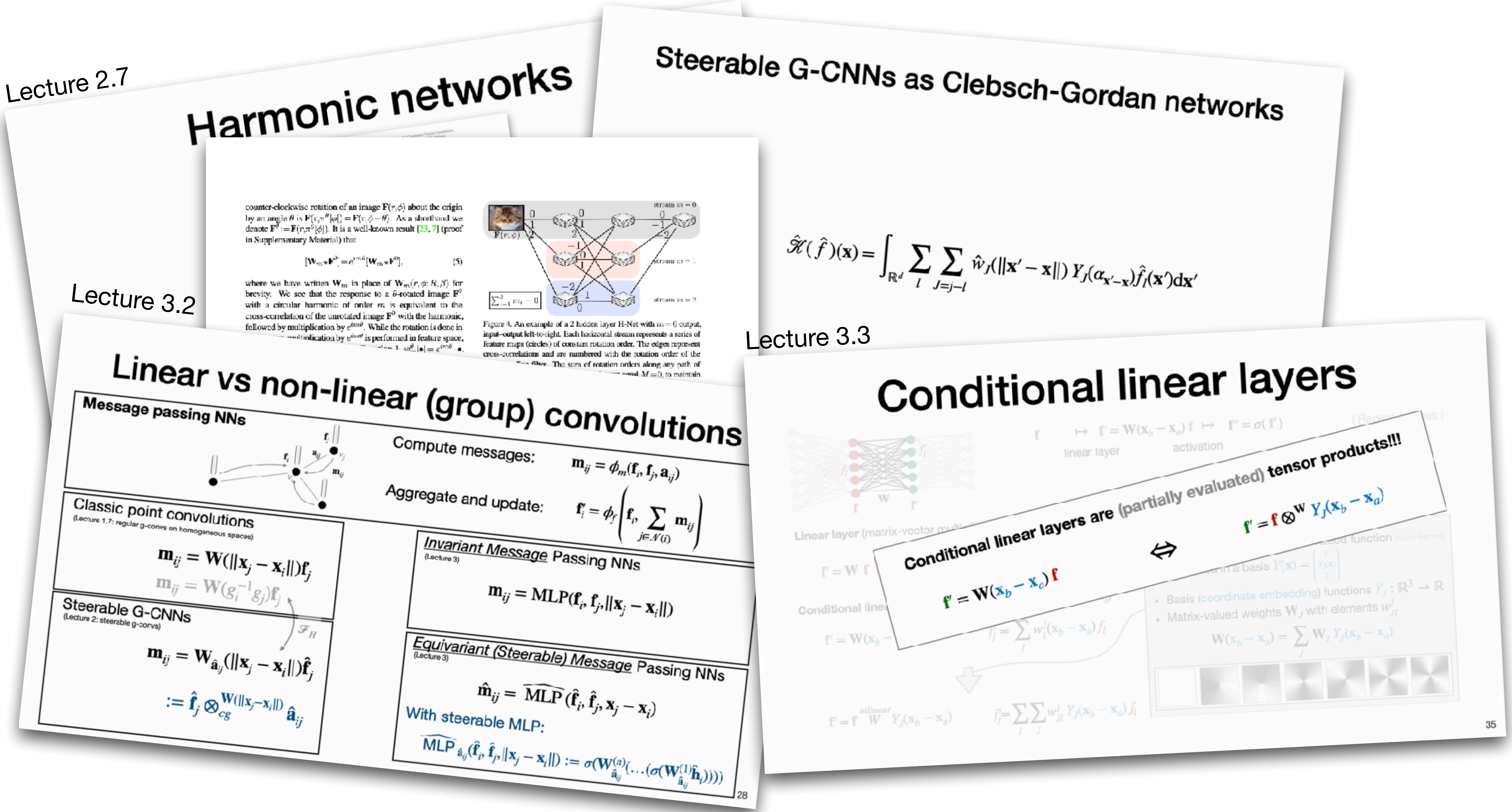
### What is `e3nn`?

`e3nn` is a python library based on pytorch to create equivariant neural networks for the group $O(3)$.

### Where to start?

- Guide to the `e3nn.o3.Irreps`: Irreducible representations
- Guide to implement a Convolution
- The simplest example to start with is Tetris Polynomial Example.
- Guide to implement a Transformer

- e3nn API
    - o3
    - nn
    - io
    - math
    - util
- User Guide
    - Install
    - Irreducible representations
    - Convolution
    - Normalization
    - Point inputs with periodic boundary conditions
    - Transformer
    - Equivariance Testing
    - TorchScript JIT Support
    - Change of Basis
- Examples


Lecture 2.7
Harmonic networks
Steerable G-CNNs as Clebsch-Gordan networks
$\hat{\mathcal{K}}(\hat{f})(\mathbf{x}) = \int_{\mathbb{R}^d} \sum_l \sum_{J=j-l} \hat{w}_J(\|\mathbf{x}' - \mathbf{x}\|)\, Y_J(\alpha_{\mathbf{x}'-\mathbf{x}}) \hat{f}_l(\mathbf{x}')d\mathbf{x}'$
Lecture 3.2
Lecture 3.3
Linear vs non-linear (group) convolutions
Message passing NNs
Compute messages:
$\mathbf{m}_{ij} = \phi_m(\mathbf{f}_i, \mathbf{f}_j, \mathbf{a}_{ij})$
Aggregate and update:
$\mathbf{f}'_i = \phi_f\left(\mathbf{f}_i, \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}\right)$
Classic point convolutions
(Lecture 1.7: regular g-convs on homogeneous spaces)
$\mathbf{m}_{ij} = \mathbf{W}(\|\mathbf{x}_j - \mathbf{x}_i\|)\mathbf{f}_j$
$\mathbf{m}_{ij} = \mathbf{W}(g_i^{-1} g_j)\mathbf{f}_j$
Steerable G-CNNs
(Lecture 2: steerable g-convs)
$\mathbf{m}_{ij} = \mathbf{W}_{\hat{\mathbf{a}}_{ij}}(\|\mathbf{x}_j - \mathbf{x}_i\|)\hat{\mathbf{f}}_j$
$:= \hat{\mathbf{f}}_j \otimes_{cg}^{\mathbf{W}(\|\mathbf{x}_j - \mathbf{x}_i\|)} \hat{\mathbf{a}}_{ij}$
$\mathcal{F}_H$
Invariant Message Passing NNs
(Lecture 3)
$\mathbf{m}_{ij} = \mathrm{MLP}(\mathbf{f}_i, \mathbf{f}_j, \|\mathbf{x}_j - \mathbf{x}_i\|)$
Equivariant (Steerable) Message Passing NNs
(Lecture 3)
$\hat{\mathbf{m}}_{ij} = \widehat{\mathrm{MLP}}(\hat{\mathbf{f}}_i, \hat{\mathbf{f}}_j, \mathbf{x}_j - \mathbf{x}_i)$
With steerable MLP:
$\widehat{\mathrm{MLP}}_{\hat{\mathbf{a}}_{ij}}(\hat{\mathbf{f}}_i, \hat{\mathbf{f}}_j, \|\mathbf{x}_j - \mathbf{x}_i\|) := \sigma(\mathbf{W}^{(n)}_{\hat{\mathbf{a}}_{ij}}(\ldots(\sigma(\mathbf{W}^{(1)}_{\hat{\mathbf{a}}_{ij}}\hat{\mathbf{h}}_i))))$
28
Conditional linear layers
linear layer
activation
Conditional linear layers are (partially evaluated) tensor products!!!
$\mathbf{f}' = \mathbf{W}(\mathbf{x}_b - \mathbf{x}_a)\,\mathbf{f}$
⇔
$\mathbf{f}' = \mathbf{f} \otimes^{\mathbf{W}} Y_J(\mathbf{x}_b - \mathbf{x}_a)$
35