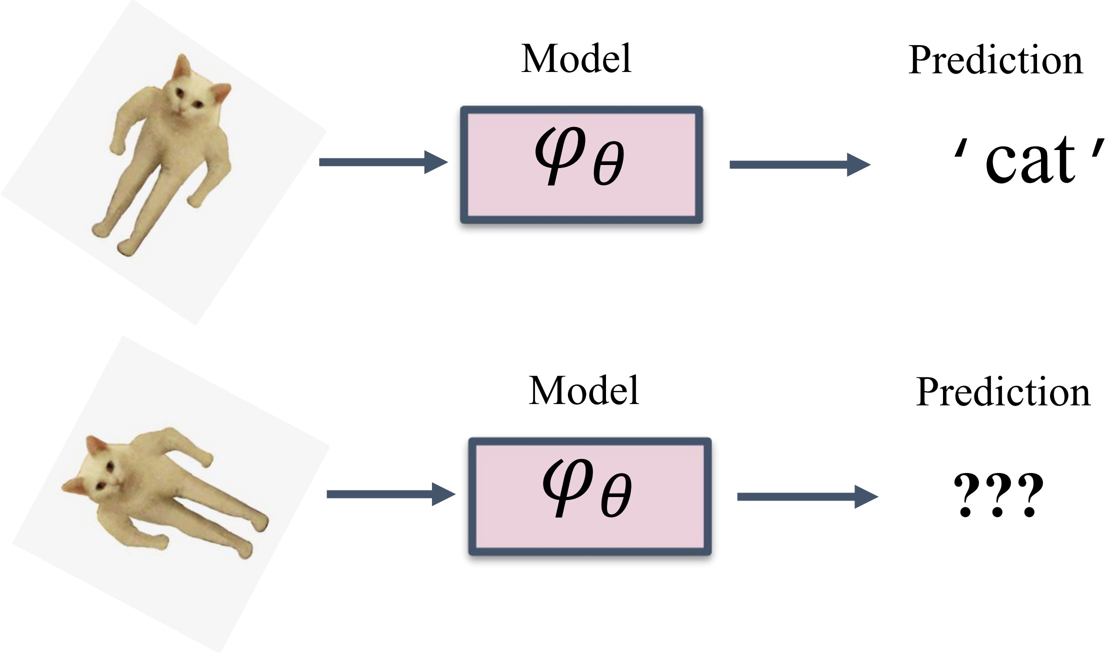
Symmetries in ML models and Group Equivariance II

Wenhan Gao

AMS, Stony Brook University

Recap: Group Equivariance in ML

Group equivariance in ML models is about **enforcing symmetries** in our architectures. In a nutshell, group equivariance means that **if the input is transformed, the output will be changed in a predictable way.**



Model

φ_θ

Prediction

'cat'

Model

φ_θ

Prediction

???

Issues in Data Augmentation:

- ✗ **No guarantee** of having symmetries in the model
- ✗ **Wasting valuable net capacity** on learning symmetries from data
- ✗ **Redundancy** in learned feature representation

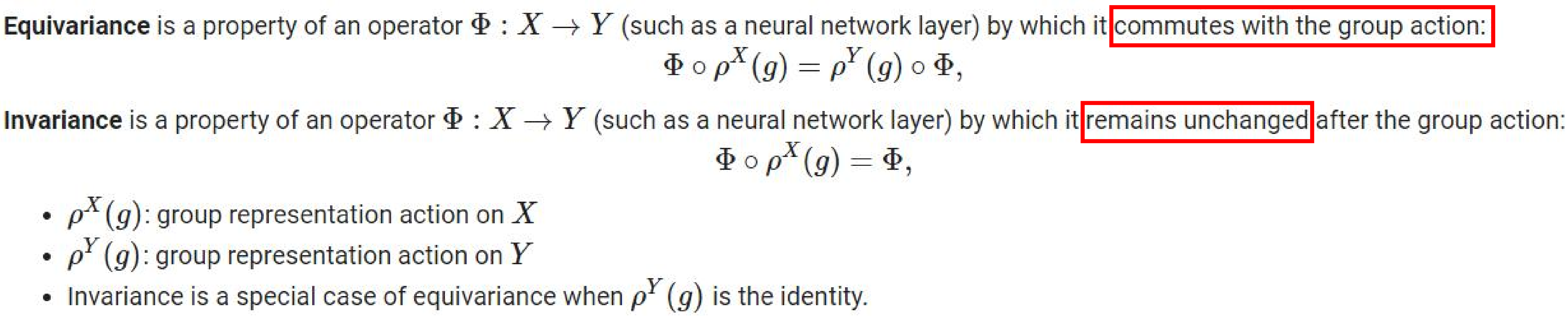
Solution:

- ✓ Building symmetries into the model by design!

Building symmetry into ML has led to major breakthroughs in deep learning:

- ○ Imposing **translational symmetry** and parameter sharing allowed CNNs to essentially solve computer vision.
- ○ Group Equivariance **conceptualizes CNN success (symmetry exploitation) and generalizes it to tasks with other symmetries.**

Recap: Equivariance and Invariance



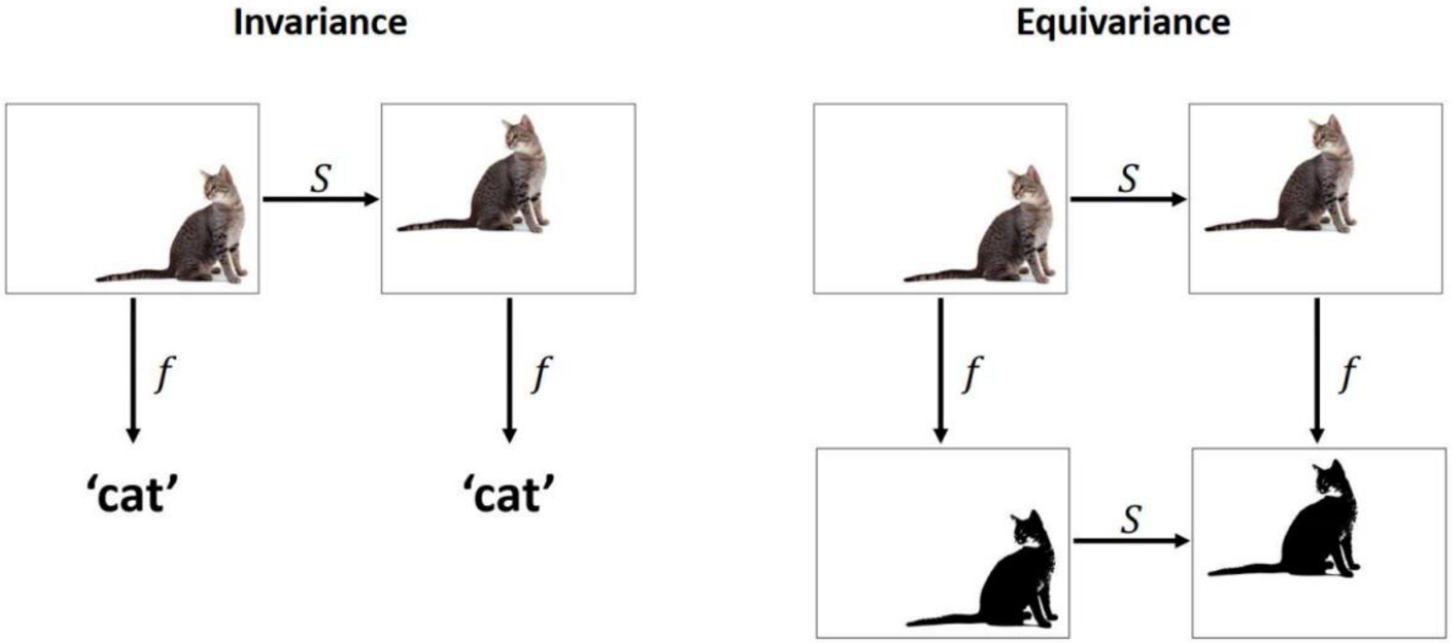
Equivariance is a property of an operator $\Phi : X \to Y$ (such as a neural network layer) by which it commutes with the group action:

$$\Phi \circ \rho^X(g) = \rho^Y(g) \circ \Phi,$$

Invariance is a property of an operator $\Phi : X \to Y$ (such as a neural network layer) by which it remains unchanged after the group action:

$$\Phi \circ \rho^X(g) = \Phi,$$

- $\rho^X(g)$: group representation action on X
- $\rho^Y(g)$: group representation action on Y
- Invariance is a special case of equivariance when $\rho^Y(g)$ is the identity.



Invariance

Equivariance

Recap: Regular Group CNN

High-level ideas:

- In CNNs, we scan through many translated versions of the image while creating the feature map. If we need to compute the cross-correlation for a transformed image, we can simply go and look up the relevant outputs because we have already computed them.

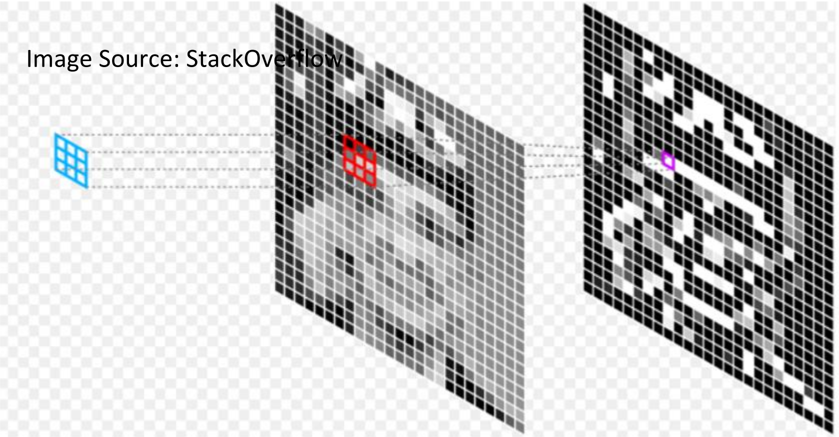
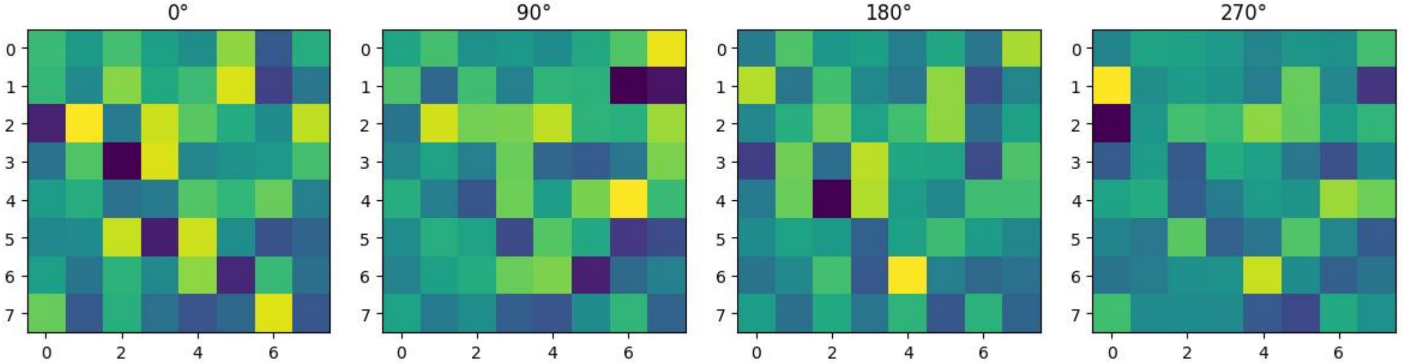


Image Source: StackOverflow

CNN: One Feature Map



0°

90°

180°

270°

Group CNN: Multiple Feature Maps

- In Group-CNNs, we follow the same idea: we lift the image to the group space and scan through all the transformed versions of the image.

$$\text{translational weight sharing} \Longleftrightarrow \text{translation group equivariance}$$
$$\text{affine weight sharing} \Longleftrightarrow \text{affine group equivariance}$$

Left-over: Steerable Group CNN

There is an issue in regular group CNNs: if the group size is large or even infinite, it is intractable to compute the summation/integral. For example, if we wish to achieve equivariance to continuous rotations, regular group CNNs cannot generate feature maps for every degree of rotation.

Solution: Use steerable features. More practically, we aim to approximate continuous feature maps with discrete representations. The rotation group, which is defined on a ring, is steered by the Fourier basis.

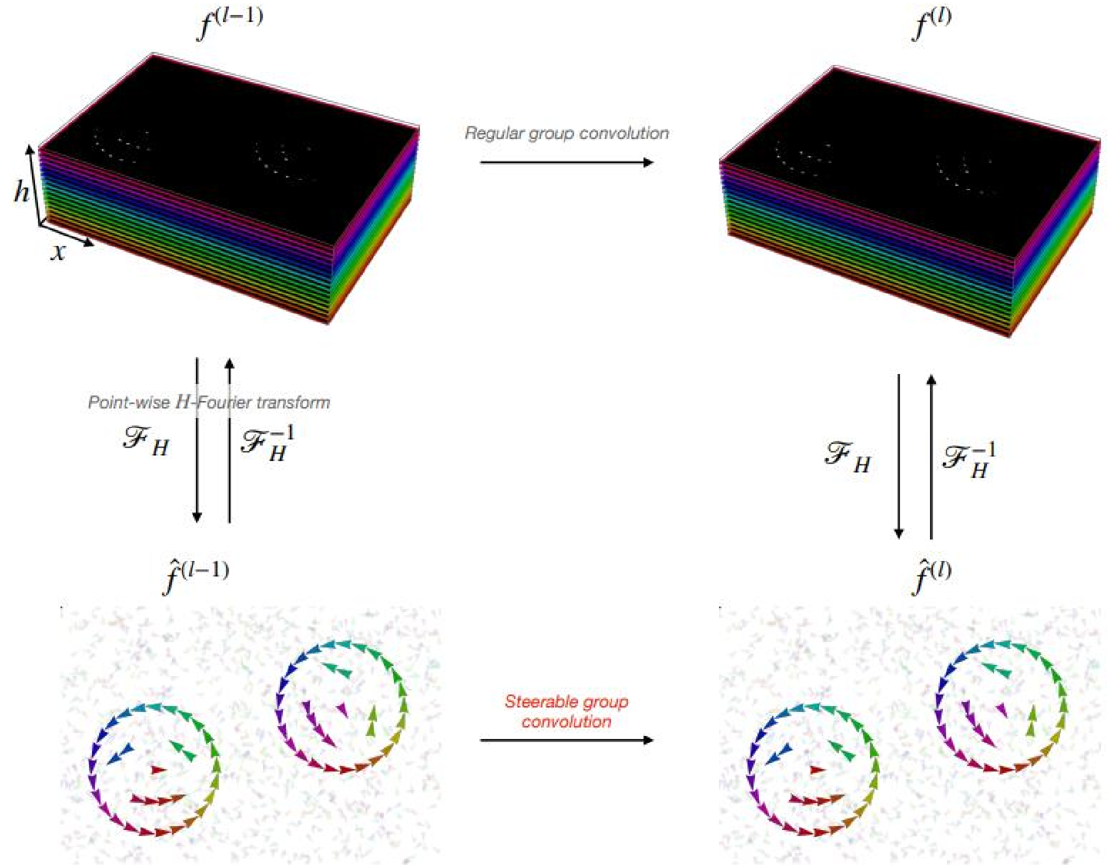


Image From:UvA - An Introduction to Group Equivariant Deep Learning

Mathematical Preliminary: Fourier Basis and Series

The set of functions

$$\{1, \cos(n\theta), \sin(n\theta)\}_{n=1}^{\infty}$$

forms a complete orthogonal basis for functions defined on the interval $[0, 2\pi]$ under suitable conditions.

WLOG, we can write the complete orthonormal basis as:

$$\{\frac{1}{\sqrt{2\pi}}, \frac{\cos(\theta)}{\sqrt{\pi}}, \frac{\sin(\theta)}{\sqrt{\pi}}, \frac{\cos(2\theta)}{\sqrt{\pi}}, \frac{\sin(2\theta)}{\sqrt{\pi}}, \frac{\cos(3\theta)}{\sqrt{\pi}}, \frac{\sin(3\theta)}{\sqrt{\pi}}, \ldots\}$$

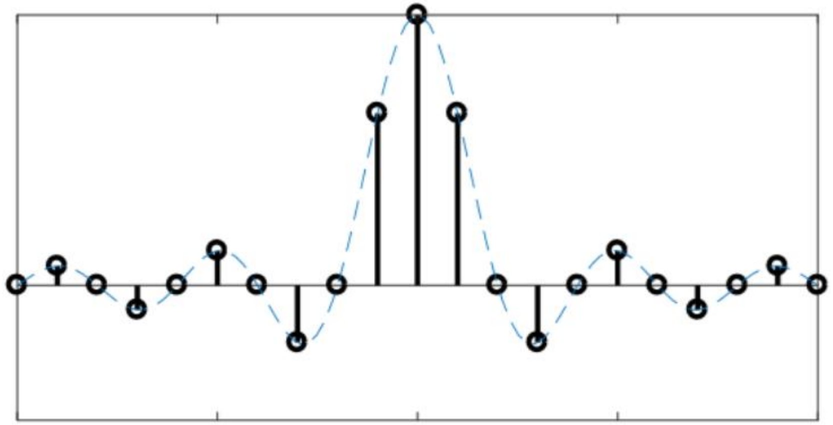
The fourier series for functions defined on the interval

$$\begin{aligned} f(\theta) &= \frac{a_0}{2} + \sum_{n=1}^{\infty}(a_n \cos n\theta + b_n \sin n\theta) \\ &= \frac{a_0}{2} + \sum_{n=1}^{\infty}\left(a_n \frac{e^{in\theta}+e^{-in\theta}}{2} + b_n \frac{e^{in\theta}-e^{-in\theta}}{2i}\right) \\ &= \frac{a_0}{2} + \sum_{n=1}^{\infty}\frac{a_n - ib_n}{2}e^{in\theta} + \sum_{n=1}^{\infty}\frac{a_n+ib_n}{2}e^{-in\theta} \\ &= \sum_{n=-\infty}^{\infty} c_n e^{in\theta}, \end{aligned}$$

where $c_0 = \frac{a_0}{2}, \;\; c_n = \frac{a_n - ib_n}{2}, \;\; c_{-n} = \frac{a_n+ib_n}{2}$.

The coefficients can be obtained by

$$c_n = \frac{1}{2\pi}\int_0^{2\pi} f(\theta)\, e^{-in\theta} d\theta, \;\; n = 0, \pm 1, \pm 2, \ldots$$

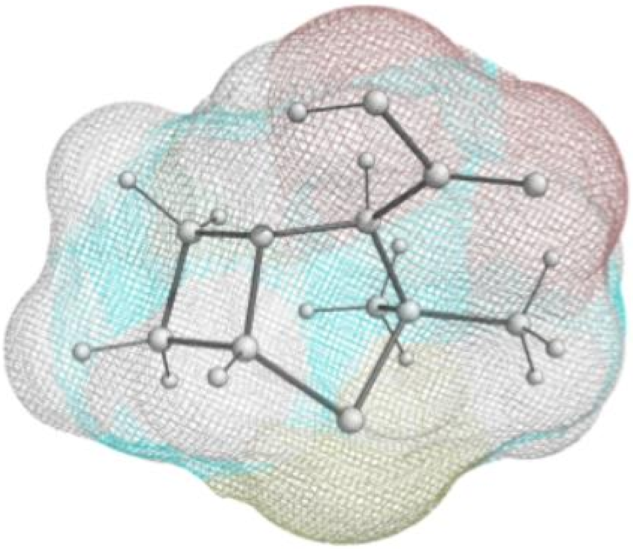


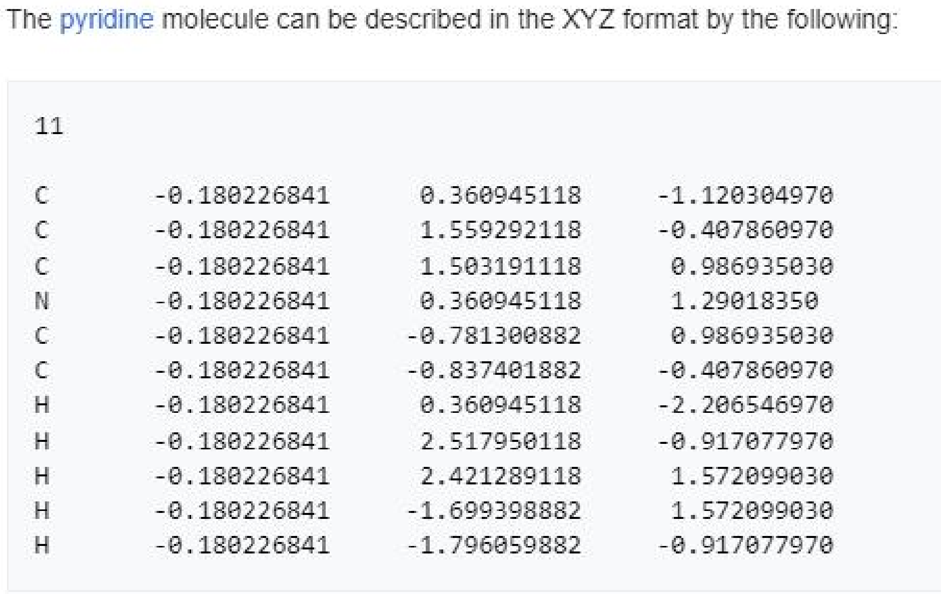
Any function defined on $\mathbb{L}_2(S_1)$ can be written as a fourier series.

Introduction: Group Averaging

Group convolution is a popular framework for equivariant neural networks in fields such as computer vision, where the data can be treated as ***functions***, and transformations act on the domain of these functions. In today's talk, we will delve into another framework, group averaging, for designing equivariant networks. Group averaging is currently mainly applied to "***geometric data***", e.g., point clouds or molecular structures that can be viewed as a set of "points/nodes" with coordinate information.

A Particular application is in geometric graph networks for computational physics, computational chemistry, 3D computer vision, etc.. It is desirable to have E(3) equivariant graph NNs in these tasks. Maintaining coordinate information offers ***greater expressivity*** compared to invariant methods that rely on relative distances [1].





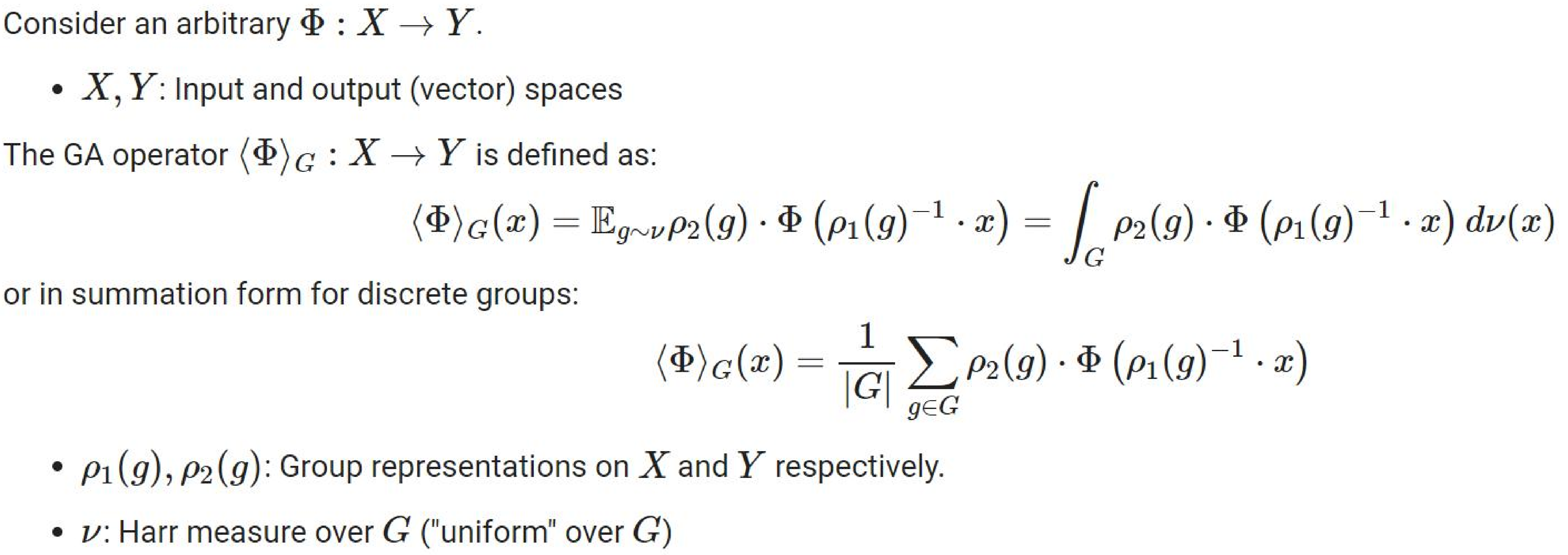
The pyridine molecule can be described in the XYZ format by the following:

```
11

C       -0.180226841      0.360945118     -1.120304970
C       -0.180226841      1.559292118     -0.407860970
C       -0.180226841      1.503191118      0.986935030
N       -0.180226841      0.360945118      1.29018350
C       -0.180226841     -0.781300882      0.986935030
C       -0.180226841     -0.837401882     -0.407860970
H       -0.180226841      0.360945118     -2.206546970
H       -0.180226841      2.517950118     -0.917077970
H       -0.180226841      2.421289118      1.572099030
H       -0.180226841     -1.699398882      1.572099030
H       -0.180226841     -1.796059882     -0.917077970
```

[1]On the Expressive Power of Geometric Graph Neural Networks

Definition: Group Averaging



Consider an arbitrary $\Phi : X \to Y$.

- X, Y: Input and output (vector) spaces

The GA operator $\langle \Phi \rangle_G : X \to Y$ is defined as:

$$\langle \Phi \rangle_G(x) = \mathbb{E}_{g \sim \nu} \rho_2(g) \cdot \Phi\left(\rho_1(g)^{-1} \cdot x\right) = \int_G \rho_2(g) \cdot \Phi\left(\rho_1(g)^{-1} \cdot x\right) d\nu(x)$$

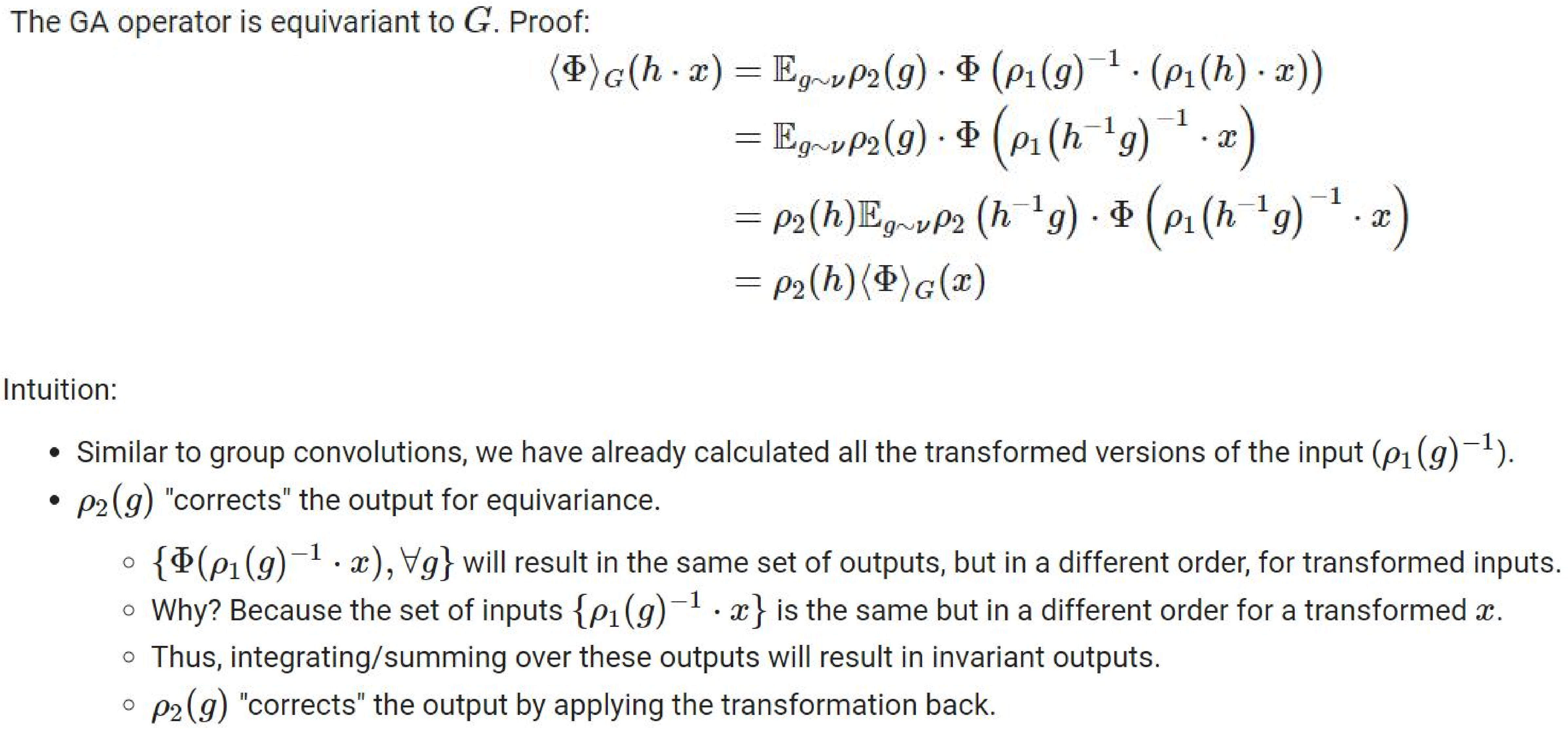
or in summation form for discrete groups:

$$\langle \Phi \rangle_G(x) = \frac{1}{|G|} \sum_{g \in G} \rho_2(g) \cdot \Phi\left(\rho_1(g)^{-1} \cdot x\right)$$

- $\rho_1(g), \rho_2(g)$: Group representations on X and Y respectively.
- ν: Harr measure over G ("uniform" over G)

Issue: When $|G|$ is large (e.g., combinatorial groups such as permutations) or infinite (e.g., continuous groups such as rotations), then exact averaging is intractable.

Intuition: Group Averaging



The GA operator is equivariant to G. Proof:

$$\begin{aligned}
\langle\Phi\rangle_G(h\cdot x) &= \mathbb{E}_{g\sim\nu}\rho_2(g)\cdot\Phi\left(\rho_1(g)^{-1}\cdot(\rho_1(h)\cdot x)\right)\\
&= \mathbb{E}_{g\sim\nu}\rho_2(g)\cdot\Phi\left(\rho_1\left(h^{-1}g\right)^{-1}\cdot x\right)\\
&= \rho_2(h)\mathbb{E}_{g\sim\nu}\rho_2\left(h^{-1}g\right)\cdot\Phi\left(\rho_1\left(h^{-1}g\right)^{-1}\cdot x\right)\\
&= \rho_2(h)\langle\Phi\rangle_G(x)
\end{aligned}$$

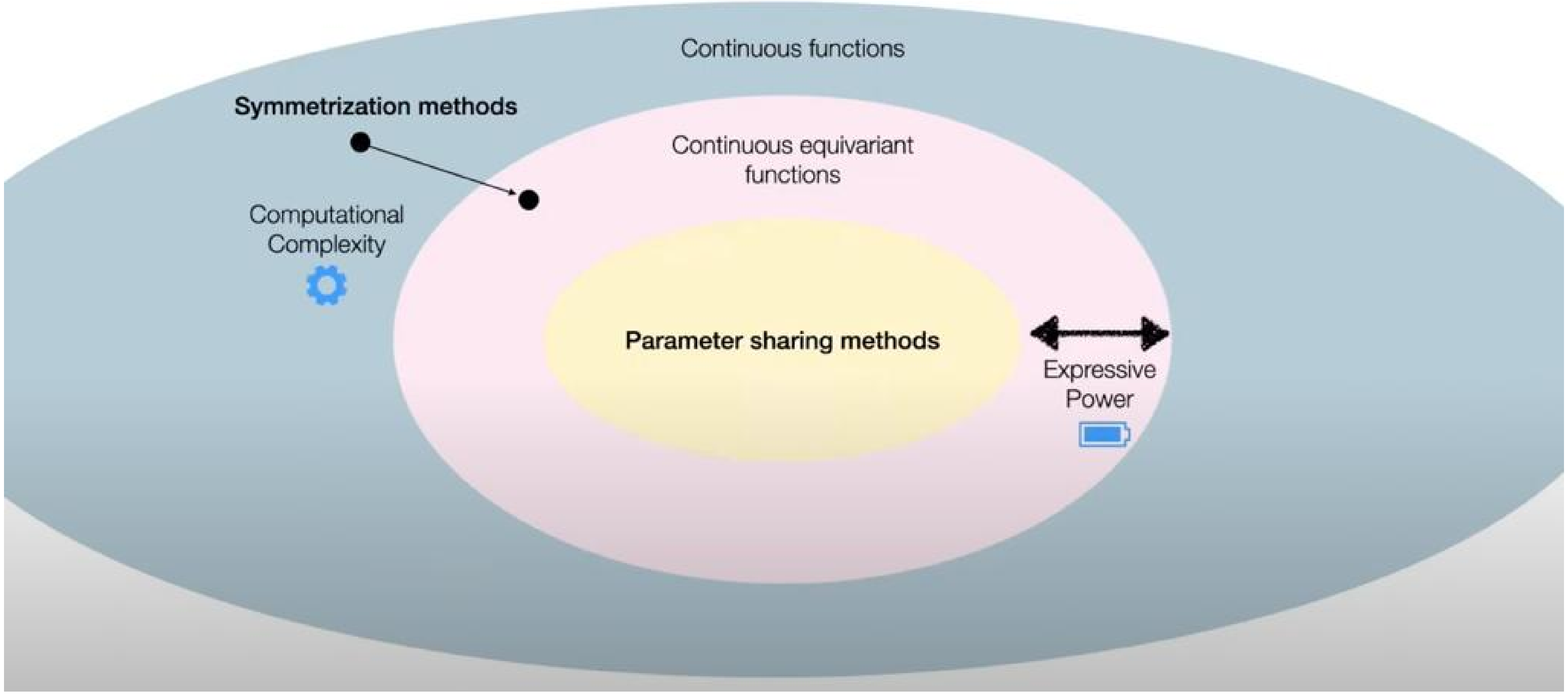
Intuition:

- Similar to group convolutions, we have already calculated all the transformed versions of the input ($\rho_1(g)^{-1}$).
- $\rho_2(g)$ "corrects" the output for equivariance.
    - $\{\Phi(\rho_1(g)^{-1}\cdot x),\forall g\}$ will result in the same set of outputs, but in a different order, for transformed inputs.
    - Why? Because the set of inputs $\{\rho_1(g)^{-1}\cdot x\}$ is the same but in a different order for a transformed x.
    - Thus, integrating/summing over these outputs will result in invariant outputs.
    - $\rho_2(g)$ "corrects" the output by applying the transformation back.

Expressive Power: Group Averaging

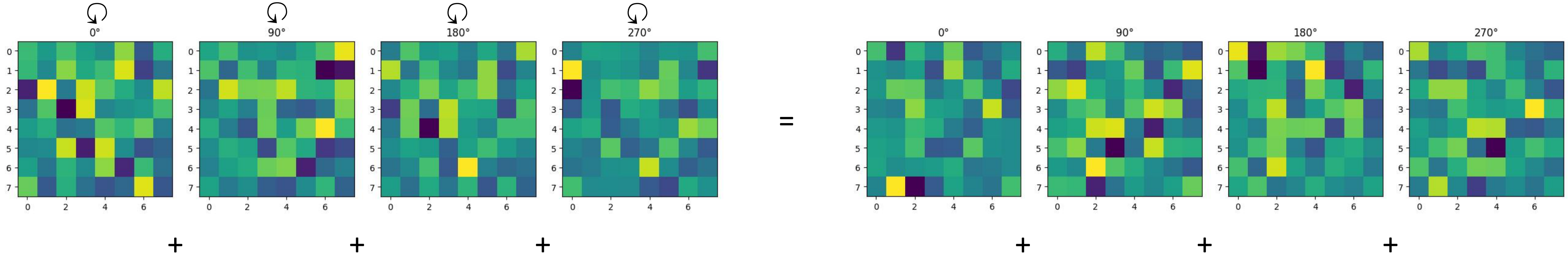
GA is as expressive as its backbone Φ when Φ is equivariant to G. To see this:

$$\begin{aligned}\langle \Phi \rangle_G(x) &= \mathbb{E}_{g\sim\nu} g \cdot \Phi\left(g^{-1} \cdot x\right) \\ &= \mathbb{E}_{g\sim\nu} g \cdot g^{-1} \cdot \Phi(x) \\ &= \Phi(x)\end{aligned}$$



Connections between Group Averaging and Group Convolution

Recall that in Group Conv, a rotation to the input will result in planar rotation + periodic shifting for the output feature map. **If we average over the feature maps to get a single feature map, the result is still equivariant to rotations, i.e., a rotation in the input will result in a rotation in the output feature map.**



Recall the definition of lifting layer:

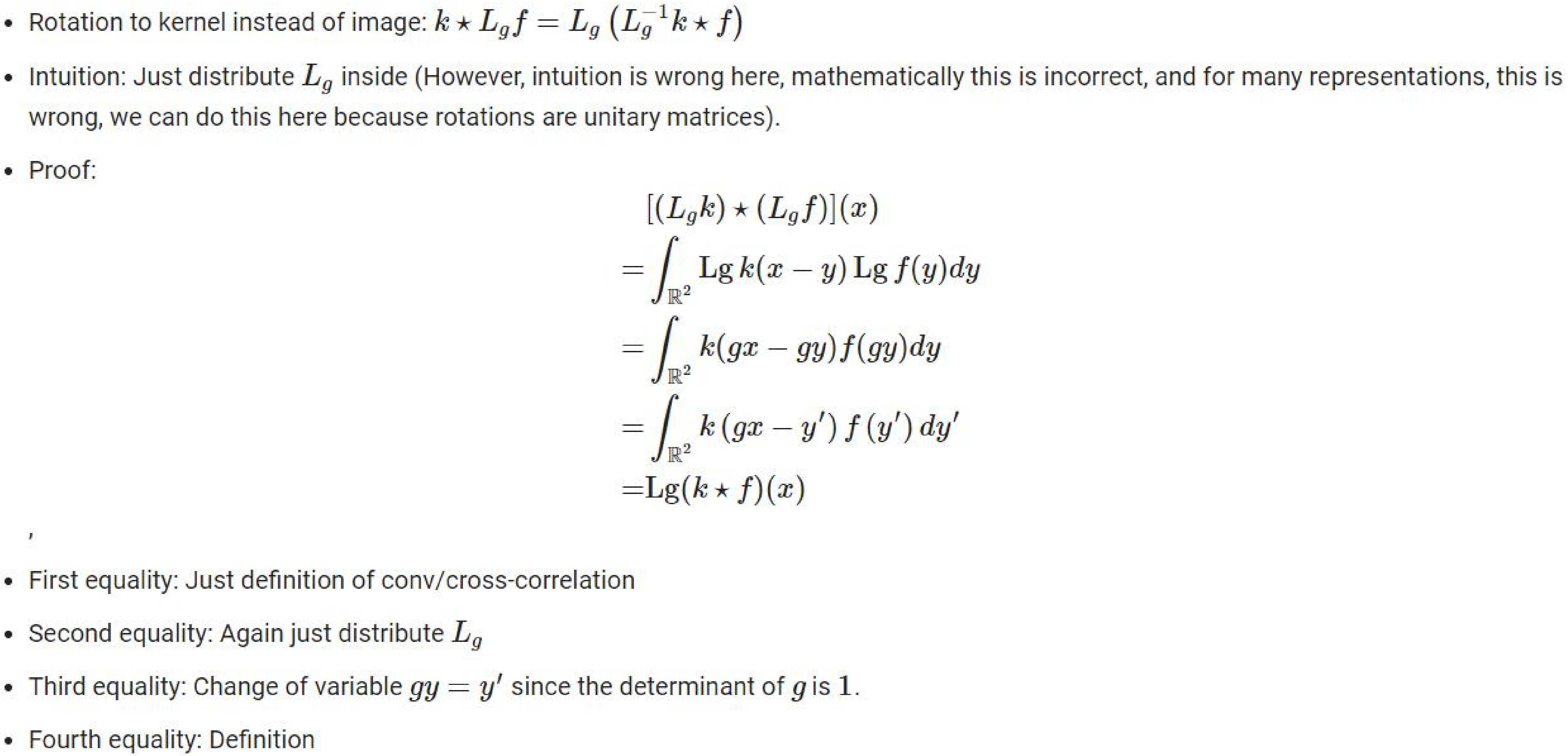
$$[f \star k](g) = \int_{\mathbb{R}^d} f(y)k\left(g^{-1}y\right)dy = (L_g \cdot k) \star f.$$

Make cross-correlation to be the backbone for group averaging:

$$\langle \Phi \rangle_G(f) = \mathbb{E}_{g\sim\nu} L_g \cdot \left[k \star \left(L_g^{-1} \cdot f\right)\right] = \frac{1}{|G|} \underbrace{\sum_{g\in G} L_g \cdot \left[k \star \left(L_g^{-1} \cdot f\right)\right]} = \overbrace{\frac{1}{|G|} \sum_{g\in G} \underbrace{(L_g \cdot k) \star f}_{\text{A feature map after lifting}}}^{\text{Sum over all feature maps}}.$$

Here, we have the “correction term”, but in group convolutions, we do not have the “correction term”. This is actually the result of moving the rotation from the image to the kernel. We'll demonstrate this in the next slide

Connections between Group Averaging and Group Convolution



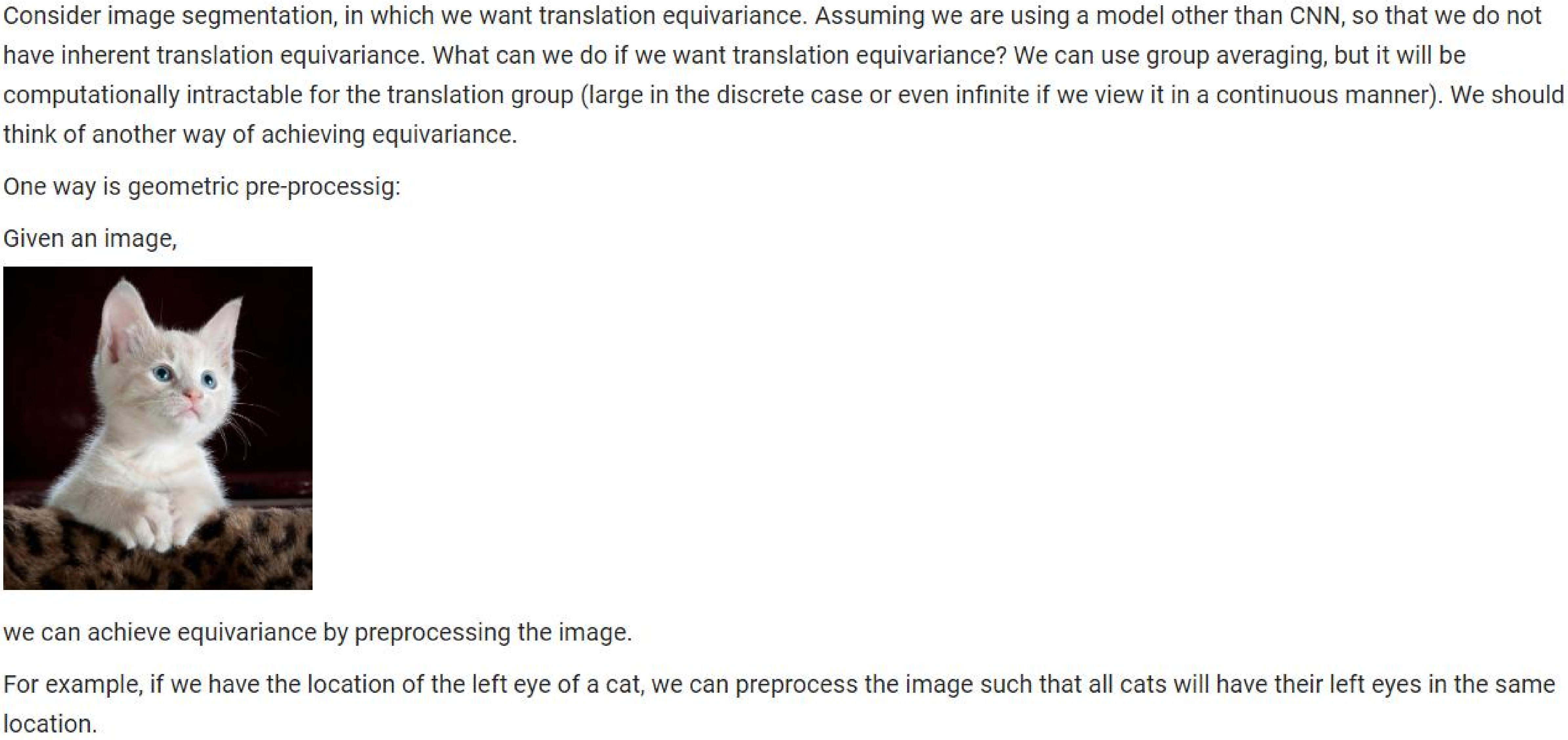
- Rotation to kernel instead of image: $k \star L_g f = L_g \left(L_g^{-1} k \star f\right)$
- Intuition: Just distribute L_g inside (However, intuition is wrong here, mathematically this is incorrect, and for many representations, this is wrong, we can do this here because rotations are unitary matrices).
- Proof:

$$\begin{aligned} & [(L_g k) \star (L_g f)](x) \\ = & \int_{\mathbb{R}^2} \mathrm{Lg}\, k(x-y)\, \mathrm{Lg}\, f(y) dy \\ = & \int_{\mathbb{R}^2} k(gx-gy) f(gy) dy \\ = & \int_{\mathbb{R}^2} k\left(gx-y'\right) f\left(y'\right) dy' \\ = & \mathrm{Lg}(k \star f)(x) \end{aligned}$$

,

- First equality: Just definition of conv/cross-correlation
- Second equality: Again just distribute L_g
- Third equality: Change of variable $gy = y'$ since the determinant of g is 1.
- Fourth equality: Definition

Motivating Example: Pre-processing



Consider image segmentation, in which we want translation equivariance. Assuming we are using a model other than CNN, so that we do not have inherent translation equivariance. What can we do if we want translation equivariance? We can use group averaging, but it will be computationally intractable for the translation group (large in the discrete case or even infinite if we view it in a continuous manner). We should think of another way of achieving equivariance.

One way is geometric pre-processig:

Given an image,

we can achieve equivariance by preprocessing the image.

For example, if we have the location of the left eye of a cat, we can preprocess the image such that all cats will have their left eyes in the same location.

Definition: Frame Averaging

A frame is defined as a set valued function $\mathscr{F} : X \to 2^G$. (Taking an input $x \in X$ and mapping to a subset of G)

A frame is equivariant to G if $\mathscr{F}(g \cdot x) = g\mathscr{F}(x), \forall g \in G$. (equality of sets)

Frame Averaging operator is defined as:

$$\langle \Phi \rangle_{\mathcal{F}}(x) = \frac{1}{|\mathcal{F}(x)|} \sum_{g \in \mathcal{F}(x)} \rho_2(g) \Phi\left(\rho_1(g)^{-1} x\right)$$

Similar to group averaging, we can prove that frame averaging operator is equivariant to G if $\mathcal{F}$ is equivariant to G, and it is as expressive as its backbone if its backbone is equivariant.

Example:

Consider $X = \mathbb{R}^n, Y = \mathbb{R}^{\mathrm{n}}$, and $G = \mathbb{R}$ with addition as the group action. We choose the group actions in this case to be $\rho_1(t)\boldsymbol{x} = \boldsymbol{x} + t\mathbf{1}$, and $\rho_2(a)y = y + t$, where $t \in G, \boldsymbol{x} \in X, y \in Y$, and $\mathbf{1} \in \mathbb{R}^n$ is the vector of all ones.

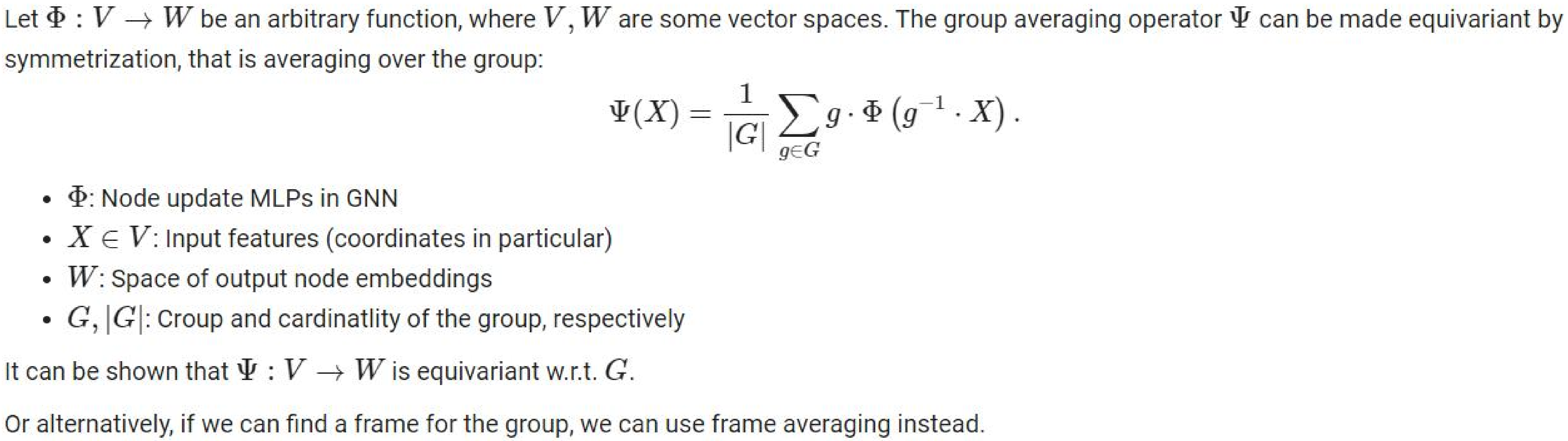
We can define the frame in this case using the averaging operator $\mathcal{F}(\boldsymbol{x}) = \left\{\frac{1}{n}\mathbf{1}^T\boldsymbol{x}\right\} \subset G = \mathbb{R}$.

Note that in this case the frame contains only one element from the group, in other cases finding such a small frame is hard or even impossible.

One can check that this frame is equivariant. The FA: $\langle \Phi \rangle_{\mathcal{F}}(\boldsymbol{x}) = \Phi\left(\boldsymbol{x} - \frac{1}{n}\left(\mathbf{1}^T x\right)\mathbf{1}\right) + \frac{1}{n}\mathbf{1}^T x$ in the equivariant case.

- Intuition: Geometric pre-processing, we subtract the average and then add the average back to obtain equivariance.

Frame Averaging: Practical Usage in Geometric GNNs



Let $\Phi : V \to W$ be an arbitrary function, where V, W are some vector spaces. The group averaging operator Ψ can be made equivariant by symmetrization, that is averaging over the group:

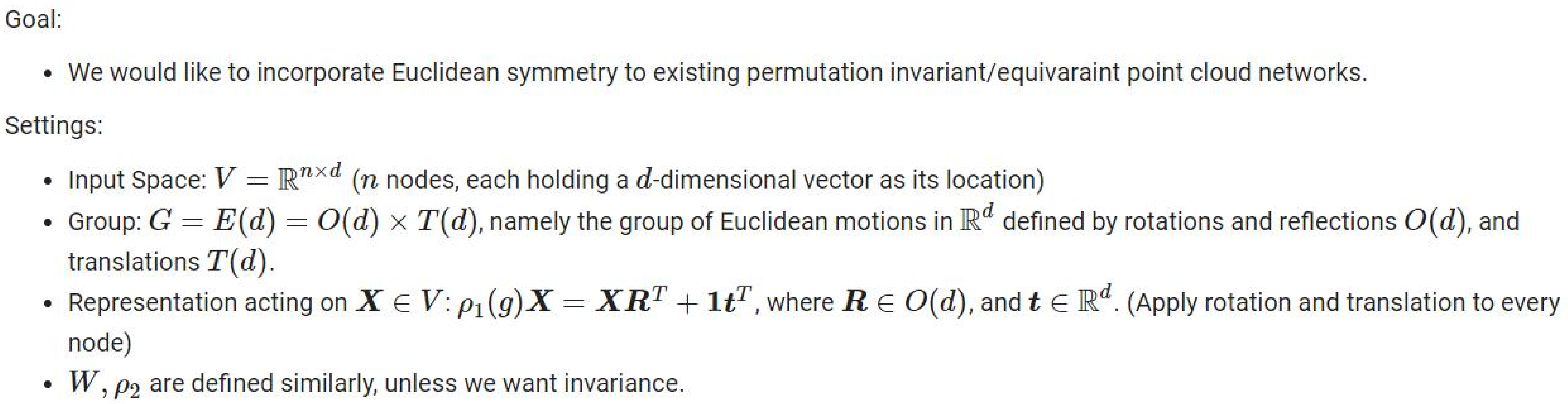
$$\Psi(X) = \frac{1}{|G|} \sum_{g \in G} g \cdot \Phi\left(g^{-1} \cdot X\right).$$

- Φ: Node update MLPs in GNN
- $X \in V$: Input features (coordinates in particular)
- W: Space of output node embeddings
- $G, |G|$: Croup and cardinatlity of the group, respectively

It can be shown that $\Psi : V \to W$ is equivariant w.r.t. G.

Or alternatively, if we can find a frame for the group, we can use frame averaging instead.

Frame Averaging: Practical Instantiation for E(3)



Goal:

- We would like to incorporate Euclidean symmetry to existing permutation invariant/equivaraint point cloud networks.

Settings:

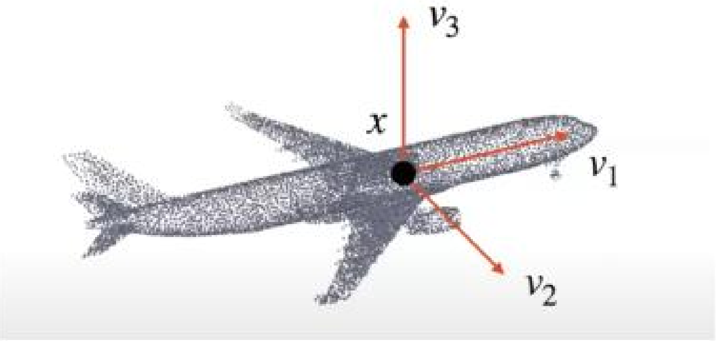
- Input Space: $V = \mathbb{R}^{n \times d}$ (n nodes, each holding a d-dimensional vector as its location)
- Group: $G = E(d) = O(d) \times T(d)$, namely the group of Euclidean motions in $\mathbb{R}^d$ defined by rotations and reflections $O(d)$, and translations $T(d)$.
- Representation acting on $\boldsymbol{X} \in V$: $\rho_1(g)\boldsymbol{X} = \boldsymbol{X}\boldsymbol{R}^T + \mathbf{1}\boldsymbol{t}^T$, where $\boldsymbol{R} \in O(d)$, and $\boldsymbol{t} \in \mathbb{R}^d$. (Apply rotation and translation to every node)
- W, ρ_2 are defined similarly, unless we want invariance.

Frame Averaging: Practical Instantiation for E(3)

Frame $\mathcal{F}(\boldsymbol{X})$ is defined based on Principle Component Analysis (PCA), as follows:

- Let $\boldsymbol{t} = \frac{1}{n}\boldsymbol{X}^T\mathbf{1} \in \mathbb{R}^d$ be the centroid of $\boldsymbol{X}$
- $\boldsymbol{C} = \left(\boldsymbol{X} - \mathbf{1}\boldsymbol{t}^T\right)^T \left(\boldsymbol{X} - \mathbf{1}\boldsymbol{t}^T\right) \in \mathbb{R}^{d\times d}$ the covariance matrix computed after removing the centroid from $\boldsymbol{X}$. In the generic case the eigenvalues of $\boldsymbol{C}$ satisfy $\lambda_1 < \lambda_2 < \cdots < \lambda_d$.
- Let $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_d$ be the unit length corresponding eigenvectors.
- Then we define $\mathcal{F}(\boldsymbol{X}) = \{([\alpha_1\boldsymbol{v}_1, \ldots, \alpha_d\boldsymbol{v}_d], t) \mid \alpha_i \in \{-1, 1\}\} \subset E(d)$.
- $[\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d]$ is a set of orthonormal vectors in $\mathbb{R}^d$, i.e., a basis of $\mathbb{R}^d$. Moreover, these vectors will "rotate" in the same way as the input.
- $\mathcal{F}(\boldsymbol{X})$ based on the covariance and centroid are $E(d)$ equivariant.

Example in 3D:



Intuition: Frame Averaging

Overall, frame averaging achieves equivariance by:

1. Create a "coordinate system" (or a set of orthonormal basis in the general case) that will change accroding the group actions acting on the input.
   - Meaning that if one input is a transformed version of another, this coordinate system will be transformed in a predictable way!
2. Representing the input in terms of this "coordinate system".
   - Inputs will be the same after this step (if one is a transformed version of another), the group action is reflected on the coordinate system now!
3. Reconstruction by the "coordinate system".
   - Now the group action is reflected on the output!

$$\langle \Phi \rangle_{\mathcal{F}}(x) = \frac{1}{|\mathcal{F}(x)|} \sum_{g \in \mathcal{F}(x)} \rho_2(g) \Phi\left(\rho_1(g)^{-1} x\right)$$

Mapping the input to a set of “coordinate systems” that respects transformations on the input

Just synchronize over all possible such “coordinate systems”

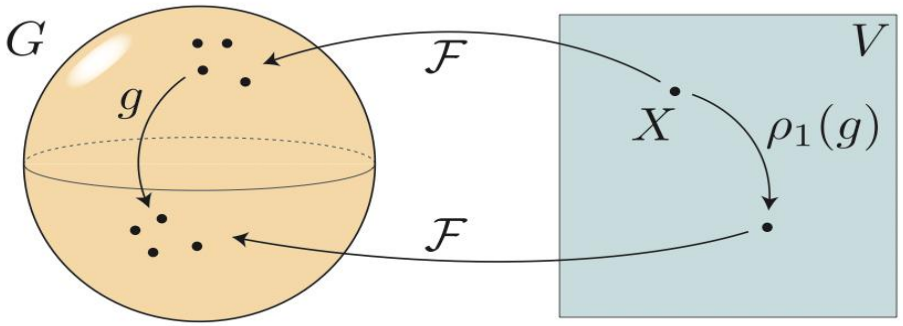
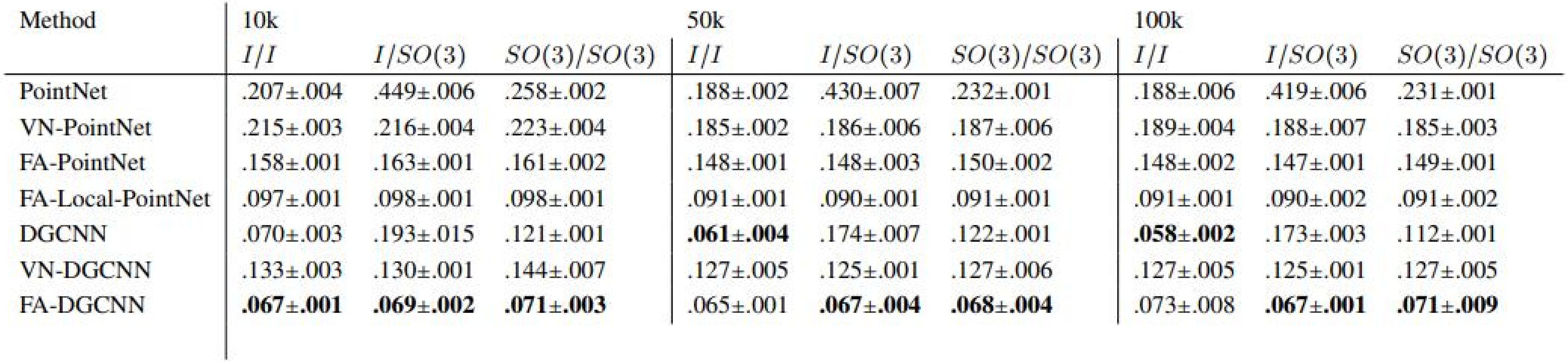


Image source: FRAME AVERAGING FOR INVARIANT AND EQUIVARIANT NETWORK DESIGN https://openreview.net/pdf?id=zIUyj55nXR

Results

Experimental results in [1] suggest that Frame Averaging (FA) significantly improves the model performance in point cloud tasks:



Method	10k			50k			100k		
	I/I	$I/SO(3)$	$SO(3)/SO(3)$	I/I	$I/SO(3)$	$SO(3)/SO(3)$	I/I	$I/SO(3)$	$SO(3)/SO(3)$
PointNet	.207±.004	.449±.006	.258±.002	.188±.002	.430±.007	.232±.001	.188±.006	.419±.006	.231±.001
VN-PointNet	.215±.003	.216±.004	.223±.004	.185±.002	.186±.006	.187±.006	.189±.004	.188±.007	.185±.003
FA-PointNet	.158±.001	.163±.001	.161±.002	.148±.001	.148±.003	.150±.002	.148±.002	.147±.001	.149±.001
FA-Local-PointNet	.097±.001	.098±.001	.098±.001	.091±.001	.090±.001	.091±.001	.091±.001	.090±.002	.091±.002
DGCNN	.070±.003	.193±.015	.121±.001	**.061±.004**	.174±.007	.122±.001	**.058±.002**	.173±.003	.112±.001
VN-DGCNN	.133±.003	.130±.001	.144±.007	.127±.005	.125±.001	.127±.006	.127±.005	.125±.001	.127±.005
FA-DGCNN	**.067±.001**	**.069±.002**	**.071±.003**	.065±.001	**.067±.004**	**.068±.004**	.073±.008	**.067±.001**	**.071±.009**

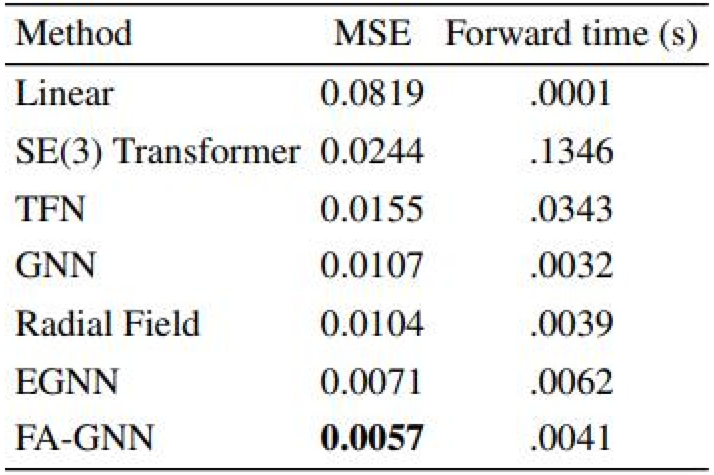
Table 1: Normal estimation, ABC dataset (Koch et al., 2019) benchmark.

Task: Normal Estimation in Geometry Processing
Data Type: Point Clouds
VN: Vector Neurons: A General Framework for SO(3)-Equivariant Networks

[1]: FRAME AVERAGING FOR INVARIANT AND EQUIVARIANT NETWORK DESIGN https://openreview.net/pdf?id=zIUyj55nXR

Results

Experimental results in [1] suggest that Frame Averaging (FA) significantly improves the model performance:

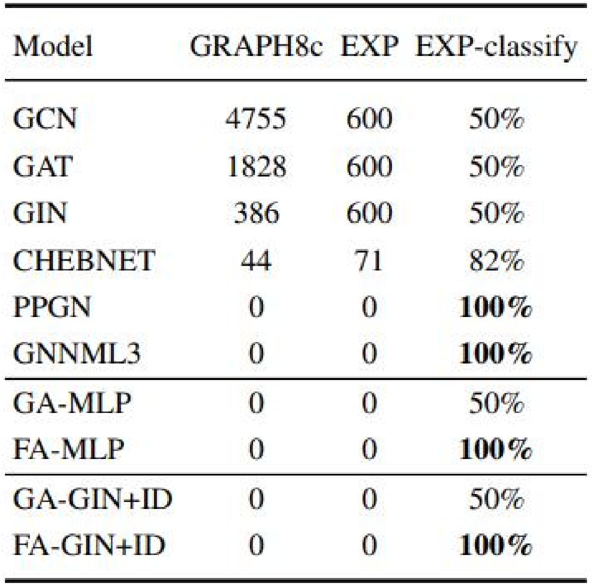


Method	MSE	Forward time (s)
Linear	0.0819	.0001
SE(3) Transformer	0.0244	.1346
TFN	0.0155	.0343
GNN	0.0107	.0032
Radial Field	0.0104	.0039
EGNN	0.0071	.0062
FA-GNN	**0.0057**	.0041

Table 3: n-body experiment (Satorras et al., 2021).

Task: N-body Problem (E(3) Equivariant)

Data Type: N = 5, 5 particles equipped with its initial position, velocity, and other physical information



Model	GRAPH8c	EXP	EXP-classify
GCN	4755	600	50%
GAT	1828	600	50%
GIN	386	600	50%
CHEBNET	44	71	82%
PPGN	0	0	**100%**
GNNML3	0	0	**100%**
GA-MLP	0	0	50%
FA-MLP	0	0	**100%**
GA-GIN+ID	0	0	50%
FA-GIN+ID	0	0	**100%**

Table 2: Graph separation (Balcilar et al., 2021; Abboud et al., 2021).

Task: Graph separation (S(n) invariant task)

Data Type: GRAPH8c that consists of all non-isomorphic, connected 8 node graphs

and EXP that consists of 3-WL distinguishable graphs that are not 2-WL distinguishable

[1]: FRAME AVERAGING FOR INVARIANT AND EQUIVARIANT NETWORK DESIGN https://openreview.net/pdf?id=zIUyj55nXR

Questions for Averaging-based Methods

- ❑ Most applications of frame averaging are applied to geometric data (point clouds). Can we adapt this idea for other data types, e.g., image data, functional data?
- ❑ Can we have an averaging-based design that also respects local symmetries?
- ❑ Under what conditions can we find a frame, and under what conditions can we find an optimal frame?
- ❑ ...
- ❑ Questions from the audience