

# GROUP EQUIVARIANT CNNs AND LOW COHERENCE MLPs



Advanced Deep Learning

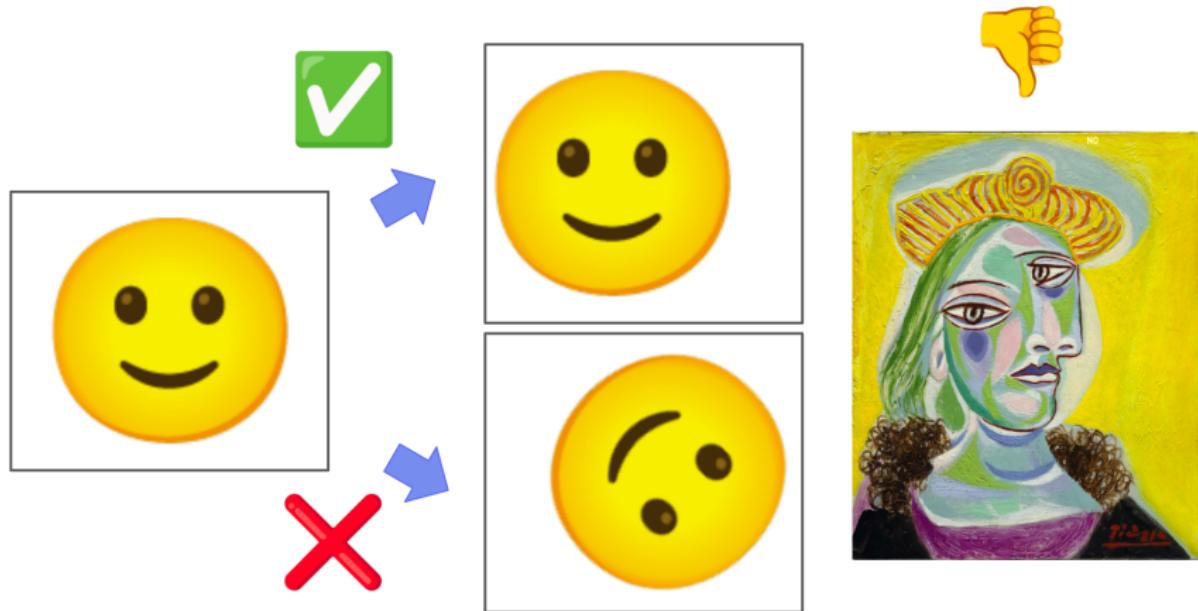
Luis Palacios

University of Trieste

July 2025

# Introduction

Convolutional Neural Networks are equivariant to translations but not to other transformations.



# Outline

1. Introduction
2. Outline
3. Part I: Group Equivariant Convolutional Neural Networks
  - 3.1 Regular Convolutional Neural Networks
  - 3.2 Group Equivariant Convolutional Neural Networks
  - 3.3 Implementation Results
4. Part II: Low Coherence MLPs
  - 4.1 Frames and Coherence
  - 4.2 Low Coherence Optimization
  - 4.3 Low Coherence MLPs
  - 4.4 Implementation Results
5. Annex
  - 5.1 Low Coherence MLPs: Fashion MNIST
  - 5.2 Group Theory 101 SpeedRun
  - 5.3 Group Equivariant Neural Networks
  - 5.4 Frame Theory Primer

# Part I: Group Equivariant Convolutional Neural Networks

# Convolutional Neural Networks (CNNs)

## Notation

Here we model images and signals in general as functions  $f : \mathbb{R}^d \rightarrow \mathbb{R}^K$ , with  $K$  the number of channels of an image or general signal. For simplicity we take  $K = 1$ , unless specified. We will focus on the space of square integrable functions,  $L^2(\mathbb{R}^d)$ .

## Convolution Operation

$$(k * f)(x) = \int_{\mathbb{R}^d} dx' k(x - x')f(x') \quad (1)$$

## Cross-Correlation Operation

$$(k \star f)(x) = \int_{\mathbb{R}^d} dx' k(x' - x)f(x') = (\mathcal{K}f)(x) \quad (2)$$

where

- $k \in L^1(\mathbb{R}^d)$  is a kernel function and,
- $\mathcal{K} : L^2(\mathbb{R}^d) \rightarrow L^2(\mathbb{R}^d)$  is a linear and bounded operator, which will be useful later.

**Note:** Here we refer to the convolution operation as the cross-correlation operation, which is more common in the literature. Furthermore, these two operations are intertwined in the forward and backward passes of a CNN.

## Convolution Operation as Template Matching

The convolution (correlation) operator can be seen as defined by the left representation of the translation group  $\mathcal{L}_x \in \text{Hom}(\mathbb{R}^d, \mathcal{U}(L^2(\mathbb{R}^d)))$ .

Furthermore, the convolution operation can be interpreted as a form of template matching, where we slide a kernel  $k$  over the input function  $f$  to produce a new function.

$$\begin{aligned}(k * f)(x) &= \int_{\mathbb{R}^d} dx' k(x' - x)f(x') \\&= \int_{\mathbb{R}^d} dx' k(x^{-1}x')f(x') \\&= \langle \mathcal{L}_x k | f \rangle_{L^2(\mathbb{R}^d)} \\&= (\mathcal{K}f)(x)\end{aligned}\tag{3}$$

where we leverage the fact that  $\mathbb{R}^d$  is a G-space of the  $(\mathbb{R}^d, +)$  group, and that  $k$  and  $f$  are functions defined on  $\mathbb{R}^d$  and thus it makes sense that the left regular representation  $\mathcal{L}_x$  acts on  $k$ .

**Note:** Here we use  $*$  because the (cross-)correlation operation is nothing more than the convolution operation with a reflected kernel, i.e.,  $k(x) = k(-x) = k^*(x)$ . This operation is called involution.

$$k * f = k^* \star f$$

## Equivariance of the convolution operation

The convolution operation is an equivariant map for the translation group

$$\begin{aligned}(k *_{\mathbb{R}^d} \mathcal{L}_t f)(x) &= \int_{\mathbb{R}^d} dx' k(x^{-1}x') \mathcal{L}_t f(x') \\&= \langle \mathcal{L}_x k | \mathcal{L}_t f \rangle_{L^2(\mathbb{R}^d)} \\&= \langle \mathcal{L}_{t^{-1}} \mathcal{L}_x k | f \rangle_{L^2(\mathbb{R}^d)} = \langle \mathcal{L}_{t^{-1}} x k | f \rangle_{L^2(\mathbb{R}^d)} \\&= \int_{\mathbb{R}^d} dx' k(x^{-1}t x') f(x') \\&= (k *_{\mathbb{R}^d} f)(t^{-1}x) \\&= [\mathcal{L}_t(k *_{\mathbb{R}^d} f)](x)\end{aligned}$$

**Remark:**  $dx' = d(tx')$  because  $dx'$  is the left Haar measure, which is an invariant measure on the group.

**Note:** The convolution operation is not an equivariant map for the group of rotations because a representation  $\mathcal{L}_t \in \text{Hom}(SO(d), \mathcal{U}(L^2(SO(d))))$  whereas the kernel function  $k$  and the input function  $f$  are defined on  $\mathbb{R}^d$ . Thus, the convolution operation does not preserve the equivariance property for rotations.

# Group Equivariant Neural Networks

## Group Equivariant Convolutional Neural Networks (G-CNNs)

$$\begin{aligned}(k *_G f)(g) &= \int_G dg' k(g^{-1}g')f(g') \\ &= \langle \mathcal{L}_g k | f \rangle_{L^2(G)} = (\mathcal{K}f)(g)\end{aligned}\tag{4}$$

with  $\mathcal{K} = L^2(G) \rightarrow L^2(G)$  the group correlation operator.

## Lifting-Correlation (The link)

$$\begin{aligned}(k *_G f)(g) &= \int_{\mathbb{R}^d} dx' k(g^{-1}x')f(x') \\ &= \int_{\mathbb{R}^d} dx' k(h^{-1}x^{-1}x')f(x') \\ &= \int_{\mathbb{R}^d} dx' \mathcal{L}_h k(x^{-1}x')f(x') \\ &= \langle \mathcal{L}_x \mathcal{L}_h k | f \rangle_{L^2(\mathbb{R}^d)} = (\mathcal{K}f)(g)\end{aligned}\tag{5}$$

with  $g = (x, h)$  and  $\mathcal{K} = L^2(\mathbb{R}^d) \rightarrow L^2(G)$  the lifting correlation operator.

# Summary of Equivariant Layers

## Theorem

A linear layer between feature maps is equivariant if and only if it is a group convolutions.

Reference: [1]

## Summary

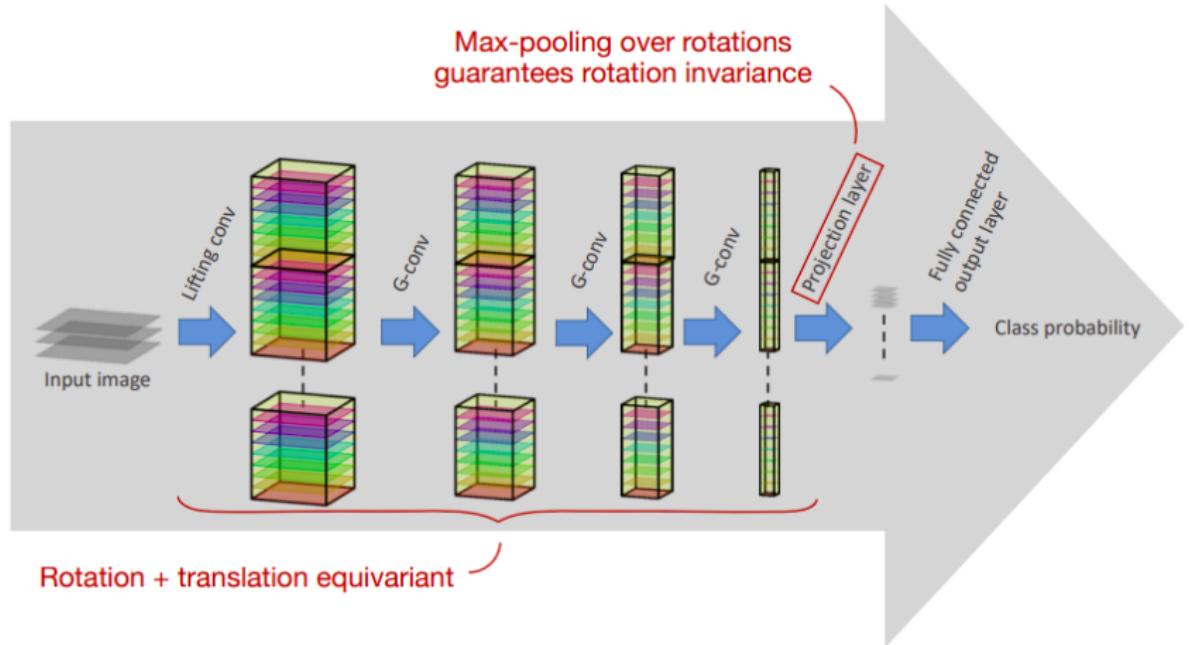
In a nutshell, we can differentiate the types of equivariant layers by:

- convolution operator  $\mathcal{K} : L^2(X) \rightarrow L^2(Y)$
- the linear part  $H$  of the group  $G$ , which adds restrictions to the kernel function  $k$

## Limitations of G-CNNs

- The group  $G$  must be finite and the convolutions need to be discrete.

# Workflow for working with GCNNs



Reference: [1]

# Practical Convolution Filter Implementation

## Planar Convolution

For the translation group in  $G = (\mathbb{Z}^2, +)$ , we have:

- input feature maps shape at layer  $l$ :  $K^\ell \times H \times W$
- output feature maps shape at layer  $l+1$ :  $K^{\ell+1} \times H' \times W'$
- layer filter weights shape at layer  $l$ :  $K^{\ell+1} \times K^\ell \times \tilde{H} \times \tilde{W}$

## Group Convolution

For an affine group  $G = \mathbb{R}^d \rtimes H$ , we have:

- input feature maps shape at layer  $l$ :  $K^\ell \times S^\ell \times H \times W$
- output feature maps shape at layer  $l+1$ :  $K^{\ell+1} \times S^{\ell+1} \times H' \times W'$
- layer filter weights shape at layer  $l$ :  $K^{\ell+1} \times S^\ell \times K^\ell \times S^\ell \times \tilde{H} \times \tilde{W}$

where:

- $K^\ell$  is the number of feature maps at layer  $\ell$ ,
- $H \times W$  is the spatial dimensions of the input feature maps,
- $S^\ell$  is dimension of the group linear part  $H$  of the feature maps at layer  $\ell$ ,
- $S^\ell \times H \times W$  is the group space dimensions of the feature maps at layer  $\ell$ . **Group convolutions just add the dimension of the group linear part  $H$  to the feature maps.**

# Implementation Results

## Architectures

- MLP: 1 hidden layer, 128 hidden units, dropout 0.2, batch norm, ReLU activation
- CNN: 4 hidden layers, 32 filters, 5x5 kernel, 1x1 stride, 0x0 padding
- GCNN: 4 hidden layers, 16 filters, 5x5 kernel, 1x1 stride, 0x0 padding,  $C_4$  group

| Model | MNIST              | Fashion-MNIST      | CIFAR-10           |
|-------|--------------------|--------------------|--------------------|
| MLP   | 0.34, 0.98         | 0.19, 0.89         | 0.24, 0.52         |
| CNN   | 0.46, <b>0.99</b>  | 0.22, <b>0.91</b>  | 0.29, <b>0.68</b>  |
| GCNN  | <b>0.92</b> , 0.98 | <b>0.46</b> , 0.86 | <b>0.35</b> , 0.49 |

Table: Test Accuracy (T.A.) Results for Different Models and Datasets. Entries: (T.A. with data augmentation, T.A. without data augmentation)

**Note:** The diehidral group  $D_4$  gave issues in the `InterpolativeGroupKernel` implementation for the `GroupConvolution` layer.  $D_4$  is a 2-dimensional group,  $C_4$  is a 1-dimensional group.

Reference: [2]

# Part II: Low Coherence MLPs

## Frames and Coherence

Definition: Frame

A **frame** is a set of vectors  $\{x_i\}_{i=1}^n$  in a Hilbert space  $H$  such that there exist constants  $A, B > 0$  (called frame bounds) satisfying:

$$A\|x\|^2 \leq \sum_{i=1}^N |\langle x_i, x \rangle|^2 \leq B\|x\|^2 \quad \forall x \in H$$

A frame generalizes the notion of a basis but allows redundancy.

Let  $H = \mathbb{R}^m$  and  $X \in \mathbb{R}^{m \times n}$  be the matrix whose  $i$ -th column is  $x_i$ , i.e.,  $X = [x_1 \ x_2 \ \dots \ x_n]$ . Then the frame condition can be written as:

$$A\|x\|^2 \leq x^T X X^T x \leq B\|x\|^2.$$

### Special Frame Types

- **Unit-norm frame:** Each frame vector has unit norm:  $\|x_i\| = 1$
- **Equiangular frame:** All pairwise inner products are equal in absolute value:  $|\langle x_i, x_j \rangle| = c$  for  $i \neq j$
- **Tight frame:**  $A = B$  and  $XX^T = \frac{n}{m}I_m$

References: [3], [4]

# Coherence and Equivariance

Definition: Frame Coherence

The **coherence** of a frame is a measure of how closely packed the frame vectors are. For a matrix  $X$  with frame vectors as columns, it is defined as:

$$\mu(X) = \max_{i \neq j} |\langle x_i, x_j \rangle|$$

Lower coherence means better **spread** of vectors and **less redundancy**.

## Welch Bound

For an  $m$ -dimensional unit-norm frame with  $n$  vectors, the **Welch bound** gives a theoretical lower bound on coherence:

$$\mu(X) \geq \sqrt{\frac{n-m}{m(n-1)}} = \mu_{\text{Welch}}$$

Equality is achieved iff the frame is **tight** and **equiangular**.

## Connection to Equivariance and Stability

- **Low coherence frames** are more equivariant with respect to group actions, as the frame vectors are more uniformly distributed in the Hilbert space.

# Low Coherence Optimization Problem

## Upper Bound for Structured Frames

For unit-norm tight frames with  $\kappa$  distinct inner product values (each appearing equally often), the coherence  $\mu$  satisfies:

$$\mu \leq \sqrt{\kappa} \mu_{\text{Welch}} = \mu_{\text{tight}}$$

## Optimization Objective

We seek  $n$  unit-norm vectors in  $\mathbb{R}^m$  (columns of matrix  $X$ ) whose worst-case inner product is as small as possible:

$$L_{\text{coh}}(X) = \min_{X \in \mathbb{R}^{m \times n}} \max_{1 \leq i < j \leq n} \langle x_i, x_j \rangle \quad \text{subject to} \quad \|x_i\| = 1 \quad \forall i$$



## Rationale

- Minimizing the maximum inner product in value (not magnitude) drives vectors to spread out maximally because lower inner products imply vectors are more widely separated.

## Log-Sum-Exp Approximation

### Smooth Surrogate Function

Since the maximum function is non-smooth (i.e., not differentiable) and we want to minimize it using gradient descent, we use the log-sum-exp surrogate:

$$L_{\text{coh}}(X) = \frac{1}{\lambda} \log \left( \sum_{i \neq j} \exp(\lambda \langle x_i, x_j \rangle) \right)$$

where  $\lambda > 0$  is a parameter controlling approximation tightness.

Reference: [5]

### Approximation Bounds

For  $a_{ij} = \langle x_i, x_j \rangle$ , the log-sum-exp satisfies:

$$\max_{i \neq j} a_{ij} \leq L_{\text{coh}}(X) \leq \max_{i \neq j} a_{ij} + \frac{\log(n)}{\lambda}$$

As  $\lambda \rightarrow \infty$ ,  $L_{\text{coh}}(X)$  converges to the true maximum.

# Regularized Optimization

## Regularized Loss Function

We combine the coherence loss with regularization terms to encourage both equiangularity and tightness:

$$L_{\text{total-coh}}(X) = L_{\text{coh}}(X) + \alpha \cdot L_{\text{equi}}(X) + \beta \cdot L_{\text{tight}}(X)$$

where:

- $L_{\text{equi}}(X) = \text{Var}(\{|\langle x_i, x_j \rangle| : i \neq j, i < j\})$  encourages equiangularity
- $L_{\text{tight}}(X) = \|XX^T - \frac{n}{m}I_m\|_F^2$  encourages tightness
- $\alpha, \beta > 0$  are regularization hyperparameters

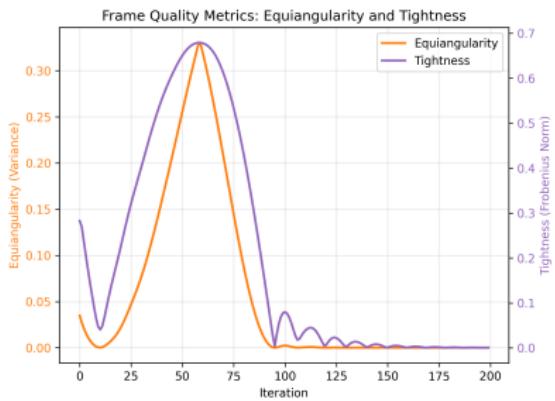
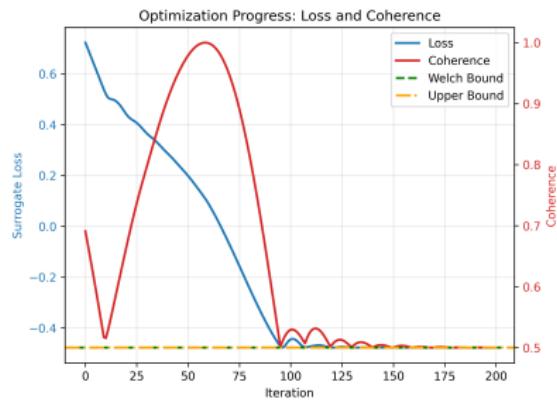
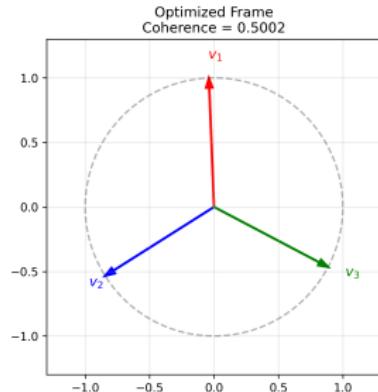
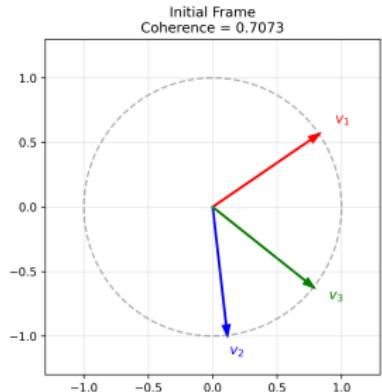
## Gradient Descent Update

The optimization is performed via gradient descent:

$$x_i^{(k+1)} = x_i^{(k)} - \eta \frac{\partial L_{\text{total-coh}}(X^{(k)})}{\partial x_i}$$

where  $\eta$  is the learning rate and the unit-norm constraint is enforced after each step, i.e.  $x_i^{(k+1)} = \frac{x_i^{(k)}}{\|x_i^{(k)}\|}$ .

# Optimization Example: Low Coherent Frames in 2D



# Low Coherence MLPs Formulation

## MLP Weight Matrices as Frames

For each linear layer in an MLP with weight matrix  $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ :

- When  $d_{\text{in}} > d_{\text{out}}$ : columns of  $W$  form frame vectors in  $\mathbb{R}^{d_{\text{out}}}$
- Frame matrix:  $X = W$  where  $X \in \mathbb{R}^{m \times n}$  with  $m = d_{\text{out}}$ ,  $n = d_{\text{in}}$
- Goal: minimize coherence  $\mu(W) = \max_{i \neq j} |\langle w_i, w_j \rangle|$  among weight columns

## Combined Loss Function

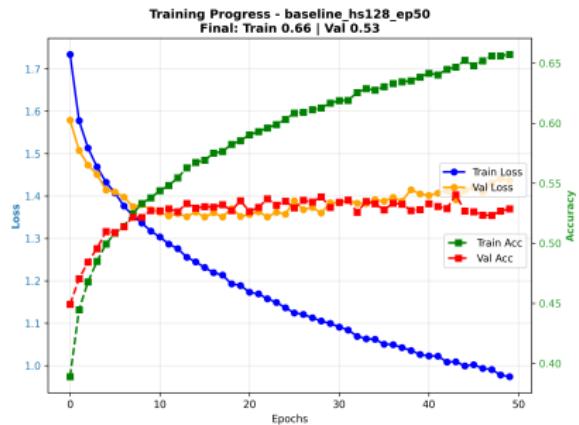
The total loss combines classification and coherence objectives:

$$L_{\text{total}}(y, \hat{y}, W) = L_{\text{original}}(y, \hat{y}) + \gamma_{\text{coh}} \sum_l L_{\text{total-coh}}(W^{(l)}, \lambda, \alpha, \beta)$$

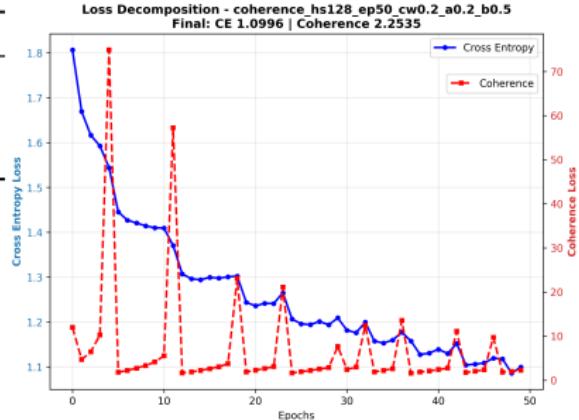
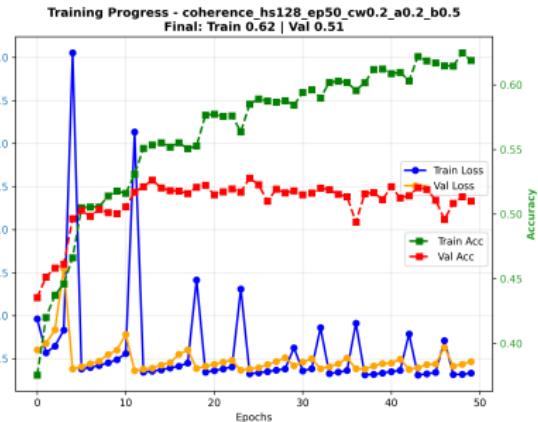
where:

- $L_{\text{original}}$  is the original loss function for the task at hand, e.g. cross-entropy
- $L_{\text{total-coh}}$  is the coherence loss function, see previous slides, and  $\lambda, \alpha, \beta$  are shared hyperparameters for all layers
- $\gamma_{\text{coh}} > 0$  is the coherence regularization weight
- Sum is over all linear layers  $l$  with weight matrices  $W^{(l)}$

# Implementation Results: CIFAR10



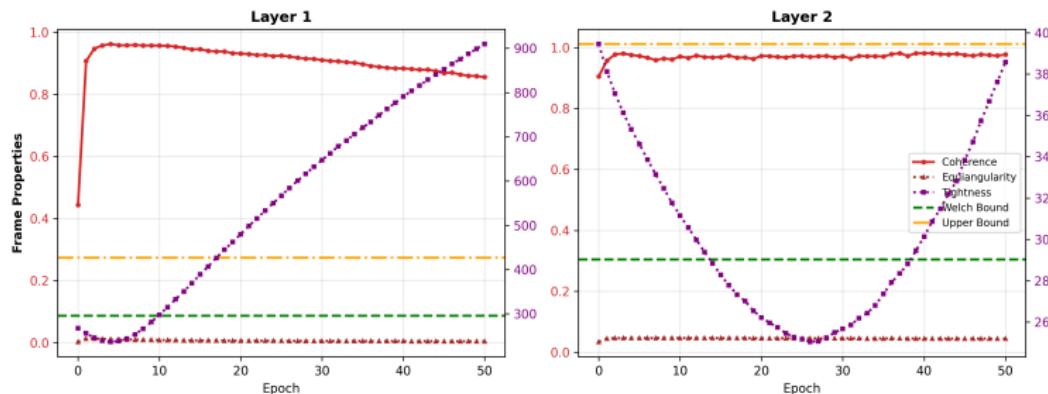
|                     | Baseline | Coherence   |
|---------------------|----------|-------------|
| Final Test Loss     | 1.44     | 2.31        |
| Final Test Accuracy | 0.52     | <b>0.51</b> |
| Best Epoch          | 43       | <b>24</b>   |
| Best Val Accuracy   | 0.54     | 0.53        |



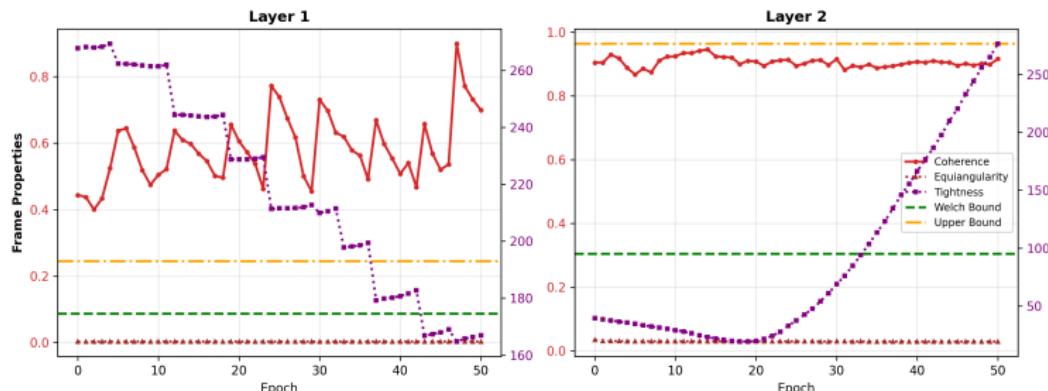
- A similar result is observed for MNIST, with better accuracy, but not for Fashion MNIST (see Annex).

# Implementation Results: CIFAR10

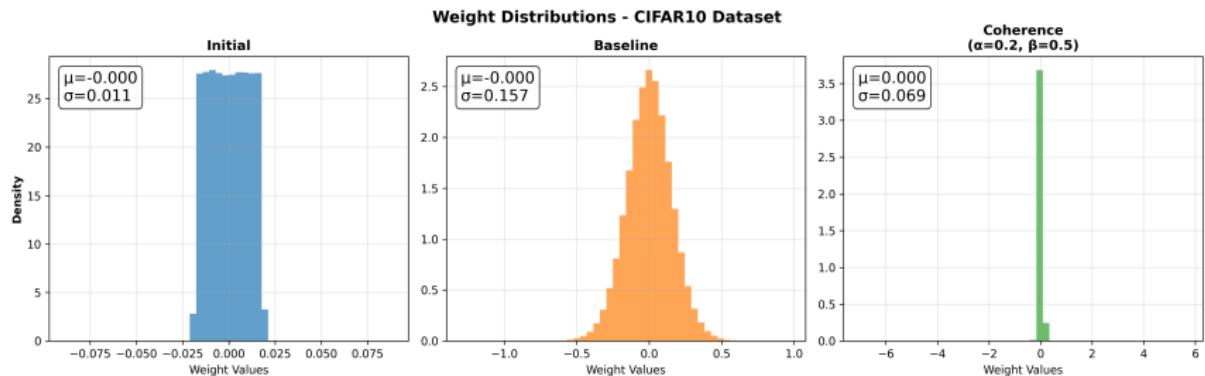
Per-Layer Frame Properties - baseline\_hs128\_ep50



Per-Layer Frame Properties - coherence\_hs128\_ep50\_cw0.2\_a0.2\_b0.5



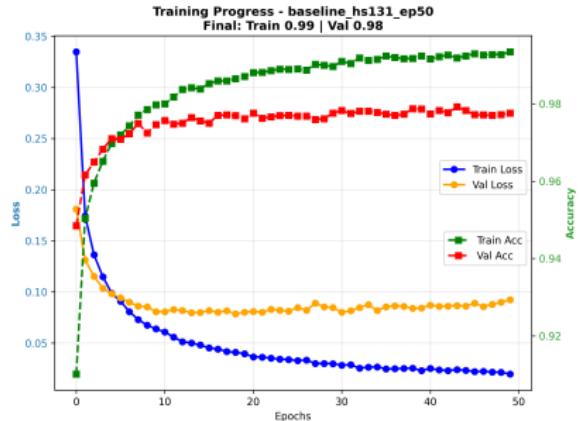
# Implementation Results: CIFAR10



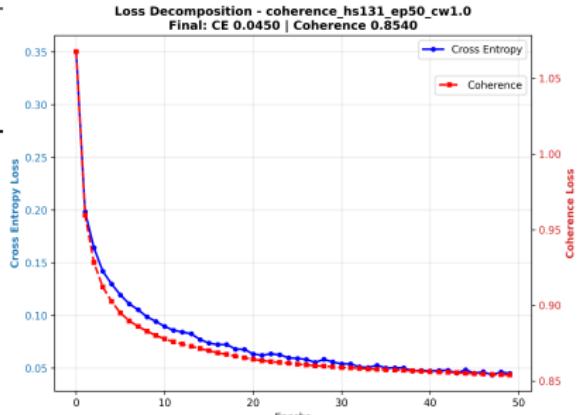
**Remark:** Regularization = Coherence + Equiangularity + Tightness.

**Obs:** The minimum coherence optimization slows down training.

# Implementation Results: MNIST

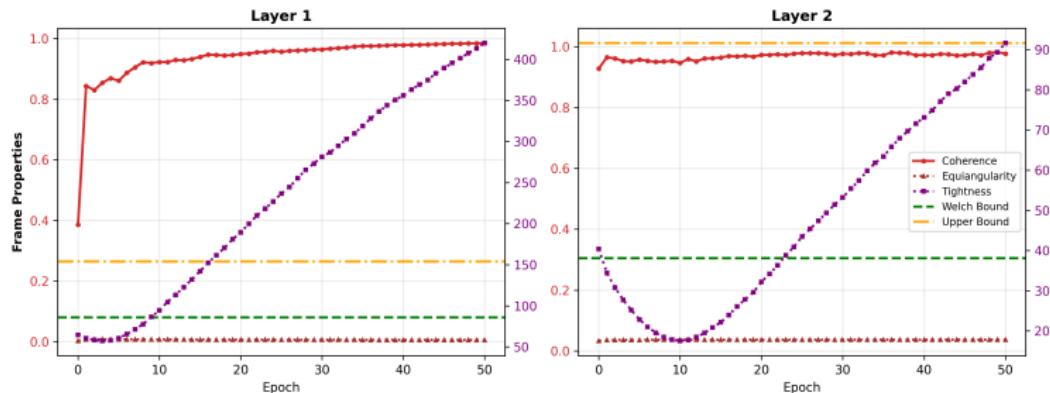


|                     | Baseline | Coherence |
|---------------------|----------|-----------|
| Final Test Loss     | 0.08     | 0.94      |
| Final Test Accuracy | 0.98     | 0.97      |
| Best Epoch          | 43       | 38        |
| Best Val Accuracy   | 0.98     | 0.97      |

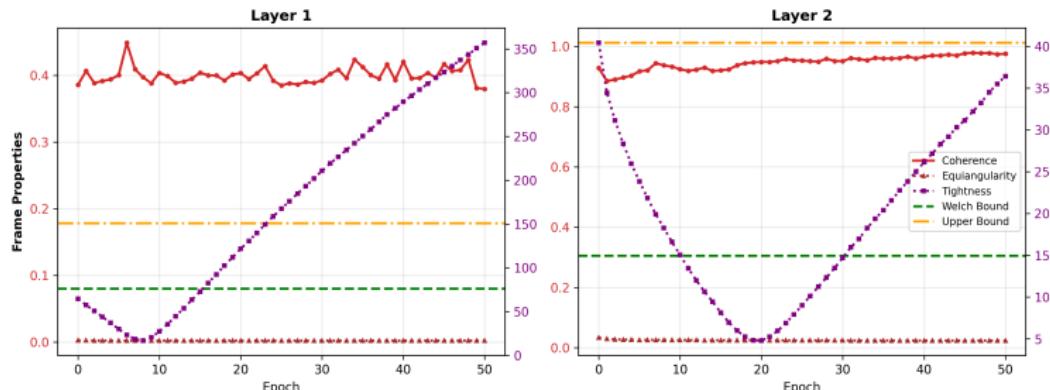


# Implementation Results: MNIST

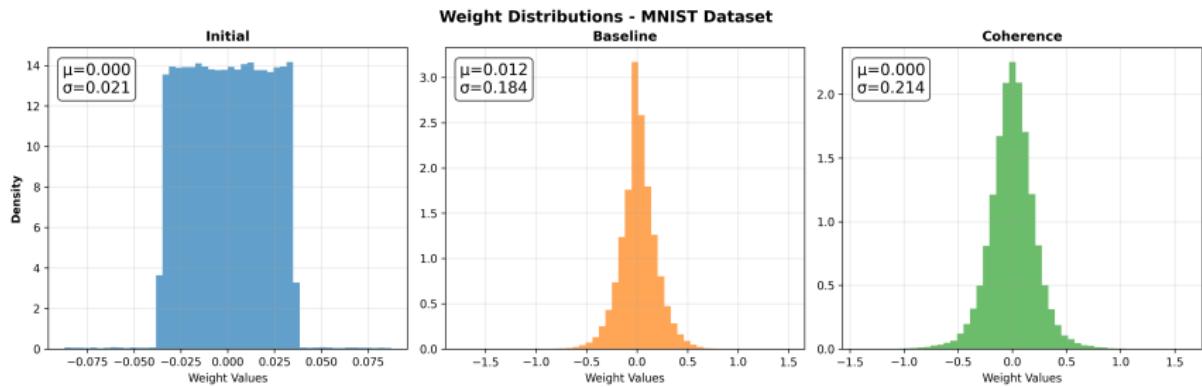
Per-Layer Frame Properties - baseline\_hs131\_ep50



Per-Layer Frame Properties - coherence\_hs131\_ep50\_cw1.0



# Implementation Results: MNIST



**Remark:** Regularization = Coherence.

Thank you for your

$$\text{softmax} \left( \frac{QK^\top}{\sqrt{d_k}} \odot M \right) V$$

## References I

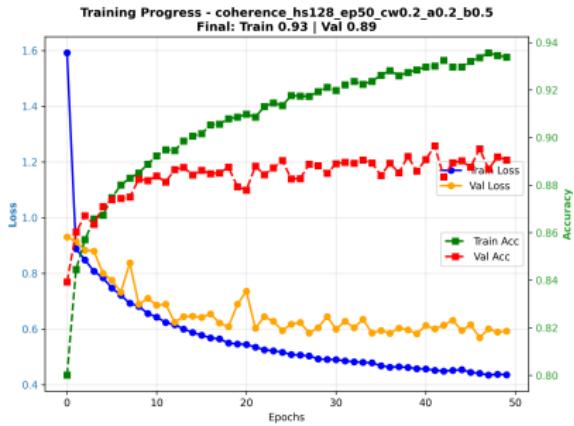
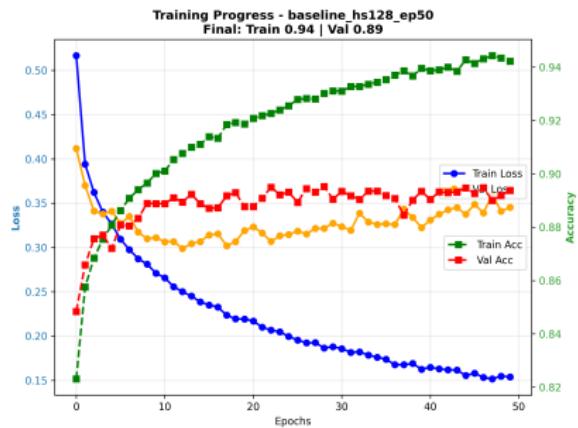
- [1] E. Bekkers, “UvA - an introduction to group equivariant deep learning,” (2025), [Online]. Available: <https://uvagedl.github.io/>.
- [2] D. Knigge and P. Lippe, “Regular group convolutions,” University of Amsterdam Deep Learning Tutorials. (2024), [Online]. Available: [https://uvadlc-notebooks.readthedocs.io/en/latest/tutorial\\_notebooks/DL2/Geometric\\_deep\\_learning/tutorial1\\_regular\\_group\\_convolutions.html](https://uvadlc-notebooks.readthedocs.io/en/latest/tutorial_notebooks/DL2/Geometric_deep_learning/tutorial1_regular_group_convolutions.html) (visited on 01/27/2025).
- [3] M. Thill and B. Hassibi, *Low-coherence frames from group fourier matrices*, Sep. 18, 2015. doi: [10.48550/arXiv.1509.05739](https://doi.org/10.48550/arXiv.1509.05739). arXiv: [1509.05739\[cs\]](https://arxiv.org/abs/1509.05739). [Online]. Available: <http://arxiv.org/abs/1509.05739> (visited on 07/06/2025).
- [4] M. Thill and B. Hassibi, “Group frames with few distinct inner products and low coherence,” *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5222–5237, Oct. 2015, ISSN: 1053-587X, 1941-0476. doi: [10.1109/TSP.2015.2450195](https://doi.org/10.1109/TSP.2015.2450195). arXiv: [1509.05087\[cs\]](https://arxiv.org/abs/1509.05087). [Online]. Available: <http://arxiv.org/abs/1509.05087> (visited on 07/06/2025).
- [5] Wikipedia contributors, “LogSumExp,” Wikipedia, The Free Encyclopedia. (Jun. 23, 2024), [Online]. Available: <https://en.wikipedia.org/wiki/LogSumExp> (visited on 01/27/2025).

## References II

- [6] L. Veefkind, “A probabilistic approach to learning the degree of equivariance in steerable CNNs,”, 2024.
- [7] E. J. Bekkers, “An introduction to equivariant convolutional neural networks for continuous groups,”, 2021. [Online]. Available: <https://uvagedl.github.io/GroupConvLectureNotes.pdf>.

# Annex

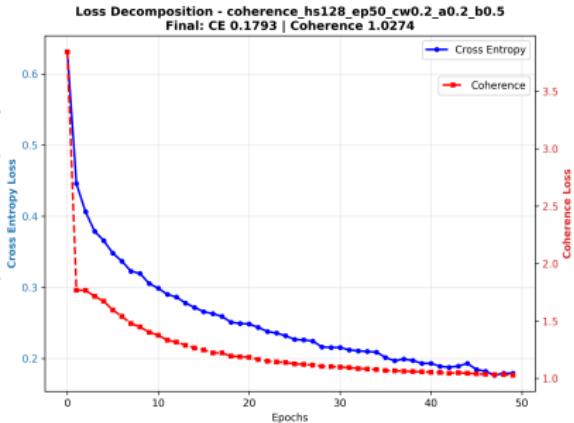
# Implementation Results: Fashion MNIST



---

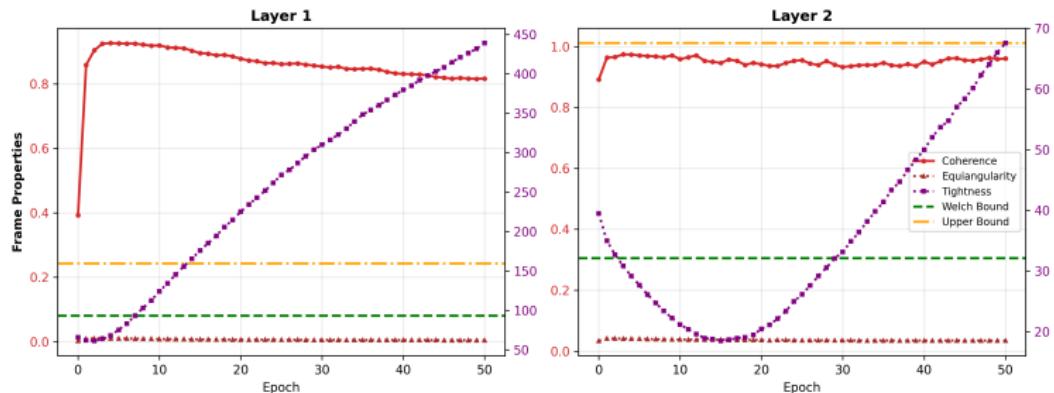
|                     | Baseline | Coherence   |
|---------------------|----------|-------------|
| Final Test Loss     | 0.37     | 0.61        |
| Final Test Accuracy | 0.89     | <b>0.89</b> |
| Best Epoch          | 28       | <b>41</b>   |
| Best Val Accuracy   | 0.90     | 0.90        |

---

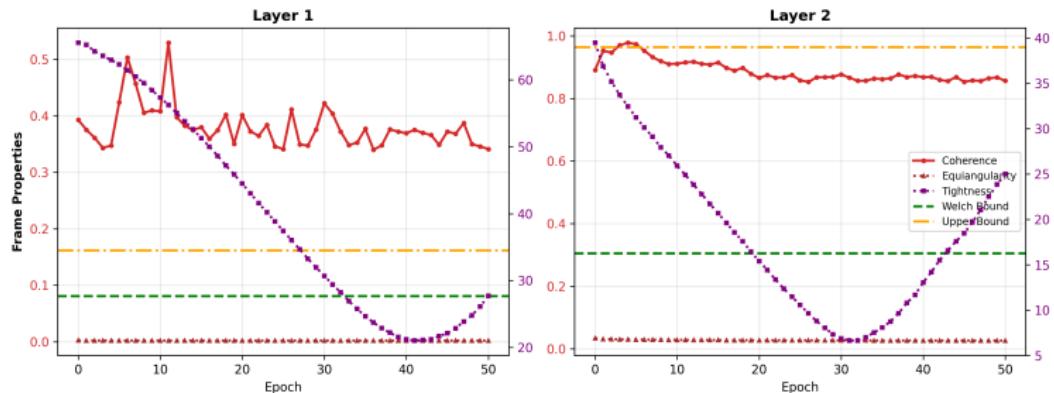


# Implementation Results: Fashion MNIST

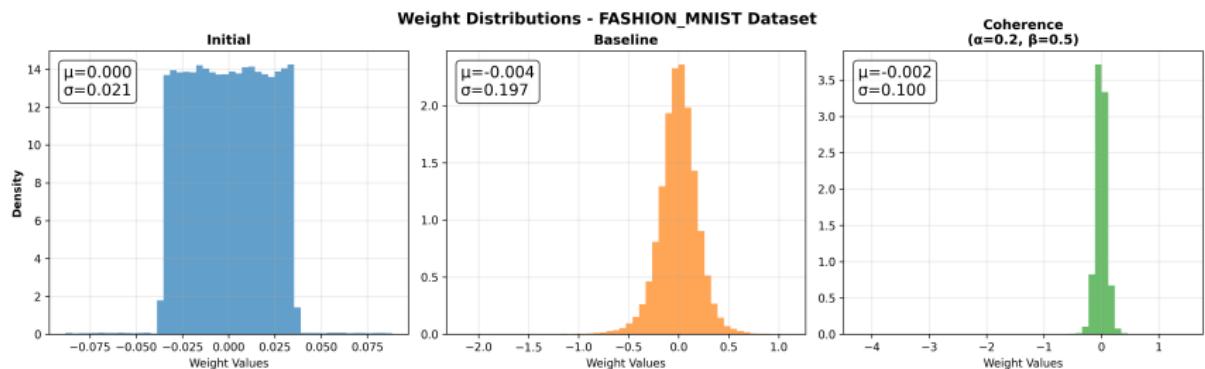
Per-Layer Frame Properties - baseline\_hs128\_ep50



Per-Layer Frame Properties - coherence\_hs128\_ep50\_cw0.2\_a0.2\_b0.5



# Implementation Results: Fashion MNIST



**Remark:** Regularization = Coherence + Equiangularity + Tightness.

# Group Theory

## Group

A group is a **set**  $G$  with a binary operation  $\cdot : G \times G \rightarrow G$  that satisfies:

- **Closure:**  $\forall g, h \in G, g \cdot h \in G$ .
- **Associativity:**  $\forall g, h, \xi \in G, (g \cdot h) \cdot \xi = g \cdot (h \cdot \xi)$ .
- **Identity Element:**  $\exists! e \in G$  st.  $\forall g \in G, e \cdot g = g \cdot e = g$ .
- **Inverse Element:**  $\forall g \in G, \exists! g^{-1} \in G$  st.  $g \cdot g^{-1} = g^{-1} \cdot g = e$ .

## (Abuse of) Notation

$$G = (G, \cdot) \quad , \quad hg \equiv h \cdot g, \quad \forall h, g \in G$$

## Group Homomorphism

A map  $\phi : G \rightarrow H$  between groups  $G = (G, \cdot)$  and  $H = (H, *)$  is a group **homomorphism** if:

$$\phi(g_1 \cdot g_2) = \phi(g_1) * \phi(g_2) \quad \forall g_1, g_2 \in G$$

i.e., the group operation is preserved under the map  $\phi$ . The “multiplication” of elements in  $G$  is mapped to the “composition” of elements in  $H$ .

# Group Theory

## Group Action and G-space

Given a group  $G$ , a (left)  $G$ -space  $X$  is a set equipped with a group action  $G \times X \rightarrow X$ ,  $(g, x) \mapsto g \cdot x$ , i.e. a map satisfying the following axioms:

- Identity:  $\forall x \in X: e \cdot x = x$
- Compatibility:  $\forall a, b \in G, \forall x \in X: a \cdot (b \cdot x) = (a \cdot b) \cdot x$

If these axioms hold,  $G$  acts on the  $G$ -space  $X$ .

## (Abuse of) Notation

$$gx \equiv g \odot x, \quad \forall g \in G, \forall x \in X$$

## Orbit of a Group

Given a group  $G$  acting on a set  $X$ , the **orbit of an element**  $x \in X$  under the group action is the set of all elements that can be reached from  $x$  by applying elements of  $G$ :

$$\mathcal{O}_x = \{g \cdot x : g \in G\} \subseteq X$$

An orbit is then the “path” traced by  $x$  under the action of  $G$ .

Reference: [6]

# Group Theory

## Direct Product Groups

Given two groups  $(G, *)$  and  $(H, +)$ , the direct product group  $(G \times H, \cdot)$  is defined as the Cartesian product  $G \times H$  of the two sets, along with the group law:

$$(g_1, h_1) \cdot (g_2, h_2) = (g_1 * g_2, h_1 + h_2).$$

$K \times H$  is often used as the shorthand notation for the direct product between  $H$  and  $K$ .

## Semi-direct Product Groups

Considering two groups  $(N, *)$  and  $(H, +)$  and a group action  $\phi : H \times N \rightarrow N$  of  $H$  on  $N$ , the semi-direct product group  $N \rtimes_{\phi} H$  is defined as the Cartesian product  $N \times H$  with the following binary operation:

$$(n_1, h_1) \cdot (n_2, h_2) = (n_1 * \phi(h_1, n_2), h_1 + h_2)$$

and the inverse element:

$$(n, h)^{-1} = (\phi(h^{-1}, n^{-1}), h^{-1})$$

# Group Theory

## Homogeneous Space

A homogeneous space is a  $G$ -space with a transitive action on  $G$ , meaning that for every pair of points  $x, y \in X$ , there exists an element  $g \in G$  such that the action of  $g$  on  $x$  moves  $x$  to  $y$ . Formally, this is expressed as:

$$\forall x, y \in X, \exists g \in G \text{ s.t. } g \cdot x = y,$$

**Remark:** All points are connected by a group action, and the whole space is just a single orbit. In other words, the space looks the same from any point.

## Stabilizer Group

Given a group  $G$  acting on a set  $X$ , the **stabilizer** of an element  $x \in X$  under the group action is the set of all group elements that fix  $x$ :

$$\text{Stab}_G(x) = \{g \in G : g \cdot x = x\}$$

**Lemma:** Let  $X$  be a homogeneous space of a group  $G$ . Then,  $X$  can be identified with  $G/H$ , where  $H = \text{Stab}_G(x_0)$  for any  $x_0 \in X$ .

# Group Theory

## Linear Group Representation

A Linear Group Representation  $\rho$  of a group  $G$  on a vector space  $V$  is a group homomorphism from  $G$  to the general linear group  $GL(V)$ :

$$\rho : G \rightarrow GL(V) \quad \text{s.t.} \quad \rho(g_1 \cdot_G g_2)v = \rho(g_1) \cdot_{GL(V)} \rho(g_2)v \quad \forall g_1, g_2 \in G, \forall v \in V$$

$\rho(g)$  is an invertible linear transformation that preserves the structure of  $G, V$ .

## Matrix Representation

A matrix representation is a linear group representation, with a choice of basis, that acts on a finite dimensional vector space  $\mathbb{R}^d$  such that all group elements are represented as concrete matrices  $\mathbf{D}(g) \in GL(\mathbb{R}^d)$ . **Used in practice.**

## Left Regular Representation

For a group  $G$  and function  $f : G \rightarrow Y$ , with  $Y$  a field, the left-regular representation  $\mathcal{L}_g$  **acts** on  $f$  by “shifting” its domain. For a group element  $g \in G$ , the action  $\mathcal{L}_g$  on  $f$  results in a new function  $\mathcal{L}_g f : G \rightarrow Y$ ,  $\mathcal{L}_g f \in L^2(G)$ , defined by:

$$(\mathcal{L}_g f)(h) = f(g^{-1}h) \quad \forall h, g \in G$$

# Group Theory

## Affine Groups

Affine groups  $G = \mathbb{R}^d \rtimes H$  are a class of groups that are the semidirect product of a group  $H \subseteq GL(\mathbb{R}^d)$  acting on  $\mathbb{R}^d$ , with  $GL(\mathbb{R}^d)$  the group of general linear transformations acting on  $\mathbb{R}^d$ .

## Example: Roto-translation Group

The **roto-translation group**  $SE(2)$  is the Special Euclidean Group in 2D, consisting of all combinations of translations and rotations in the plane. It can be defined as:

$$SE(2) = \mathbb{R}^2 \rtimes SO(2)$$

Each element is represented as  $(x, \theta)$  where  $x \in \mathbb{R}^2$  is a translation vector and  $\theta \in SO(2)$  is a rotation angle. The group operation is:

$$(x_1, \theta_1) \cdot (x_2, \theta_2) = (\mathbf{R}_{\theta_1} x_2 + x_1, \theta_1 + \theta_2)$$

where  $\mathbf{R}_\theta$  is the 2D rotation matrix for angle  $\theta$ .

# Group Theory

## Equivariance

Given a group  $G$  and two sets  $X$  and  $Y$  that are acted on by  $G$ , a map (e.g., a layer in a neural network)  $f : X \rightarrow Y$  is equivariant iff

$$\forall x \in X, \forall g \in G : f(g \cdot x) = g \cdot f(x)$$

- Equivariance guarantees that input transformations result in predictable output transformations.
- Input transformations shift information rather than losing it, so no information is lost.
- Enables weight sharing across group transformations - features detected at one location/pose are equally detected at others.

## Invariance

Given a group  $G$  and two sets  $X$  and  $Y$  that are acted on by  $G$ , a map  $f : X \rightarrow Y$  is invariant iff

$$\forall x \in X, \forall g \in G : f(g \cdot x) = f(x)$$

- Output remains constant for all inputs related by group actions.

## Equivariant Neural Networks

In the basic case, a neural network is a function that maps an input  $f$  to an output function  $g$  via a series of linear maps  $\mathcal{N}_i$  followed by non-linear activation functions, which can be expressed as:

$$g = \mathcal{N}(f) = \mathcal{N}_L \circ \mathcal{N}_{L-1} \circ \dots \circ \mathcal{N}_1(f) \quad (6)$$

Theorem: Dunford-Pettis: Linear bounded maps are integral transforms

Let  $K : L^2(X) \rightarrow L^2(Y)$  be a linear and bounded operator that maps between spaces of feature maps  $L^2(X)$  and  $L^2(Y)$ . Then there exists a kernel  $k \in L^1(Y \times X)$  such that  $K$  is an integral transform via:

$$(Kf)(y) = \int_X k(y, x)f(x) d\mu_X(x)$$

with  $f \in L^2(X)$ , and  $d\mu_X$  a “Radon measure” on  $X$ .

**Vectors:**

$$f^{l+1} = Wx \quad \leftrightarrow \quad f_j^{l+1} = \sum_{i=1}^{N_l} w_{ji} x_i^l \quad (7)$$

**Signals:**

$$f^{l+1} = Kf^l \quad \leftrightarrow \quad f^l(y) = \int_X k(y, x)f^l(x)d\mu_X(x) \quad (8)$$

# Types of Equivariant Neural Networks

Theorem: Equivariant linear layers on homogeneous spaces

Let  $\mathcal{K} : L^2(X) \rightarrow L^2(Y)$  be a linear and bounded operator. Let  $X, Y$  be homogeneous spaces on which a (Lie) group  $G$  acts transitively, and let  $d\mu_X$  be a “Radon measure” on  $X$ . Suppose  $\mathcal{K}$  is equivariant to the group  $G$  via  $L_g^Y \circ \mathcal{K} = \mathcal{K} \circ L_g^X$ , with  $L_g^X$  and  $L_g^Y$  are the left-regular representations of  $G$  on  $L^2(X)$  and  $L^2(Y)$ , respectively. **Then:**

- ①  $\mathcal{K}$  is a group convolution given by:

$$(\mathcal{K}f)(y) = \int_X d\mu_X(x) \frac{d\mu_X(g_y^{-1}x)}{d\mu_X(x)} k(g_y^{-1}x) f(x)$$

for any  $g_y \in G$  such that  $y = g_y y_0$  for some fixed origin  $y_0 \in Y$ .

- ② The kernel must satisfy the symmetry constraint:

$$k(x) = \frac{d\mu_X(g_y^{-1}x)}{d\mu_X(x)} k(h^{-1}x)$$

for any  $h \in H = \text{Stab}_G(y_0)$ , and any  $x \in X$ .

Reference: [7]

## Types of Equivariant Neural Networks

### Isotropic $\mathbb{R}^d$ Convolutions ( $X = Y = \mathbb{R}^d$ )

An isotropic  $\mathbb{R}^d$  convolution layer maps between planar signals  $L_2(\mathbb{R}^d)$  with  $\mathcal{K}$  a planar correlation given by

$$(\mathcal{K}f)(y) = \int_{\mathbb{R}^d} \frac{1}{|\det h|} k(\mathbf{x} - \mathbf{y}) f(\mathbf{x}) d\mathbf{x} \quad (9)$$

and in which  $k$  satisfies

$$\forall h \in H : \quad k(\mathbf{x}) = \frac{1}{|\det h|} k(h^{-1}\mathbf{x})$$

### Lifting Layer ( $X = \mathbb{R}^d$ , $Y = G$ )

A lifting layer maps from planar signals  $L_2(\mathbb{R}^d)$  to signals  $L_2(G)$  on the group  $G$ , with  $\mathcal{K}$  a lifting correlation given by

$$(\mathcal{K}f)(g) = \int_{\mathbb{R}^d} \frac{1}{|\det h|} k(g^{-1}\tilde{\mathbf{x}}) f(\tilde{\mathbf{x}}) d\tilde{\mathbf{x}} \quad (10)$$

### Group Convolution Layer ( $X = Y = G$ )

A group convolution layer maps between  $G$ -feature maps in  $L_2(G)$ , with  $\mathcal{K}$  a group correlation given by

$$(\mathcal{K}f)(g) = \int_G k(g^{-1}\tilde{g}) f(\tilde{g}) d\tilde{g} \quad (11)$$

with  $d\tilde{g}$  the left “Haar” measure on  $G$ .

## Example: Roto-translation lifting convolutional layer

Let  $G = SE(2) = \mathbb{R}^2 \rtimes SO(2)$  be the roto-translation group in 2D. The lifting correlation of a convolutional layer represented by the linear operator  $\mathcal{K}^l : \mathbb{R}^2 \rightarrow SE(2)$  on a function  $f \in L^2(\mathbb{R}^2)$  is:

$$\begin{aligned} f^{l+1}(g) &= (\mathcal{K}^l f)(g) \\ &= (k *_{SE(2)} f)(g) \\ &= \left\langle \mathcal{L}_x \mathcal{L}_\theta k \middle| f^l \right\rangle_{L^2(\mathbb{R}^2)} \\ &= \left\langle \mathcal{L}_x k_\theta \middle| f^l \right\rangle_{L^2(\mathbb{R}^2)} \\ &= (k_\theta *_{\mathbb{R}^2} f)(x) \end{aligned}$$

where  $f^{l+1} \in L^2(SE(2))$ ,  $g = (x, \theta)$ ,  $x \in \mathbb{R}^2$  and  $\theta \in SO(2)$ .

**Note:** the lifting convolution and G-convolutions can be obtained by first precomputing a filter bank of transformed kernels and then apply them to the input signal via the usual planar correlation (GPU optimized) on  $\mathbb{R}^2$ , or  $\mathbb{Z}^2$  to be precise in the discrete practical case. Here, the transformed kernel is obtained via  $k_\theta(x) = k(\mathbf{R}_\theta^{-1}x)$ . Computationally, this says that to get the transformed kernel at the point  $x$ , we need to do a lookup in the original kernel  $k$  at the point  $\mathbf{R}_\theta^{-1}x$ , which is the unique point that gets mapped to  $x$  by  $\theta$ .

## Frame Theory

### Matrix Definition of a Frame

The frame condition can be written as:

$$A\|x\|^2 \leq x^T X X^T x = \|X^T x\|^2 \leq B\|x\|^2.$$

This follows because:

$$\sum_{i=1}^n |\langle x, f_i \rangle|^2 = \sum_{i=1}^n (x^T f_i)^2 = x^T \left( \sum_{i=1}^n f_i f_i^T \right) x = x^T X X^T x = \|X^T x\|^2.$$

### Frame Operator

Given a frame  $\{f_i\}_{i=1}^n$ , the **frame operator**  $S : H \rightarrow H$  is defined by

$$Sx = \sum_{i=1}^n \langle x, f_i \rangle f_i.$$

Using Dirac notation, we can write:

$$S = \sum_{i=1}^n |f_i\rangle \langle f_i| \implies S|x\rangle = \sum_{i=1}^n \langle x|f_i\rangle |f_i\rangle$$

## Frame Theory

### Outer Product Decomposition of the Frame Operator

The frame operator can be written as

$$S = XX^T = \sum_{i=1}^n f_i f_i^T$$

**Proof:** Recall that the standard basis vector  $e_i \in \mathbb{R}^n$  satisfies  $(e_i)_j = \delta_{ij}$  such that  $X = \sum_{i=1}^n f_i e_i^T$ , because  $f_i e_i^T$  produces a matrix with  $f_i$  in the  $i$ -th column and zeros elsewhere.

Now compute the frame operator:

$$\begin{aligned} XX^T &= \left( \sum_{i=1}^n f_i e_i^T \right) \left( \sum_{j=1}^n e_j f_j^T \right)^T \\ &= \left( \sum_{i=1}^n f_i e_i^T \right) \left( \sum_{j=1}^n e_j f_j^T \right) \\ &= \sum_{i=1}^n \sum_{j=1}^n f_i (e_i^T e_j) f_j^T = \sum_{i=1}^n f_i f_i^T \quad \square \end{aligned}$$

## Tight Frame

A frame  $\{f_i\}_{i=1}^n$  is called a **tight frame** if the frame bounds satisfy  $A = B$ . In this case, the frame inequality becomes:

$$\sum_{i=1}^n |\langle x, f_i \rangle|^2 = A \|x\|^2 \quad \forall x \in \mathcal{H}$$

## Key Properties of Tight Frames

- **Optimal reconstruction:** Equal frame bounds provide stability
- **Energy preservation:** Perfect energy balance across all directions
- **Simplified analysis:** Frame operator becomes a scalar multiple of identity
- **Connection to equiangular frames:** Combined with equiangularity, achieves Welch bound

## Frame Operator of Tight Frames

Let  $\{f_i\}_{i=1}^n$  be a tight frame in  $\mathbb{R}^m$ , and let  $X \in \mathbb{R}^{m \times n}$  be the matrix of frame vectors. Then:

$$XX^T = AI_m$$

## Proof

Let  $S = XX^T$  be the frame operator. Since the frame is tight with bound  $A$ :

$$\langle Sx, x \rangle = \sum_{i=1}^n |\langle x, f_i \rangle|^2 = A\|x\|^2 = \langle Ax, x \rangle$$

Thus, for all  $x \in \mathbb{R}^m$ :  $\langle (S - AI)x, x \rangle = 0$ .

This implies  $S = AI_m$  by the polarization identity and positive-definiteness of  $S$ . □

## Frame Theory

### Tight Frame Bound for Unit Vectors

If the frame vectors are unit norm and the frame is tight, then:

$$A = \frac{n}{m}$$

### Proof

Taking the trace of both sides of  $XX^T = AI_m$ :

$$\text{Tr}(XX^T) = \sum_{i=1}^n \text{Tr}(f_i f_i^T) = \sum_{i=1}^n \|f_i\|^2 = n$$
$$\text{Tr}(AI_m) = A \cdot m$$

Equating:  $Am = n \Rightarrow A = \frac{n}{m}$ . □

### Geometric Interpretation

For unit-norm tight frames, each coordinate direction receives exactly  $\frac{n}{m}$  units of total energy from all frame vectors, providing perfect energy balance.