

EECE 576L

Advanced Topics in Deep Learning

Lecture 4: Group Equivariant Deep Learning

Part 1: Groups and actions

Renjie Liao

University of British Columbia

2026/27 Winter Session, Term 1 (Fall 2026)

Outline

- 1. Groups and actions**
2. Group convolutions
3. Steerable kernels
4. Typed features and 3D networks
5. Implementation and verification (optional)

Outline

1. Groups and actions

- **Choose the CNN symmetry**
- Motion groups and actions
- How CNN features transform
- Symmetrization by averaging
- Image domains and homogeneous spaces
- Normality and quotient groups

2. Group convolutions

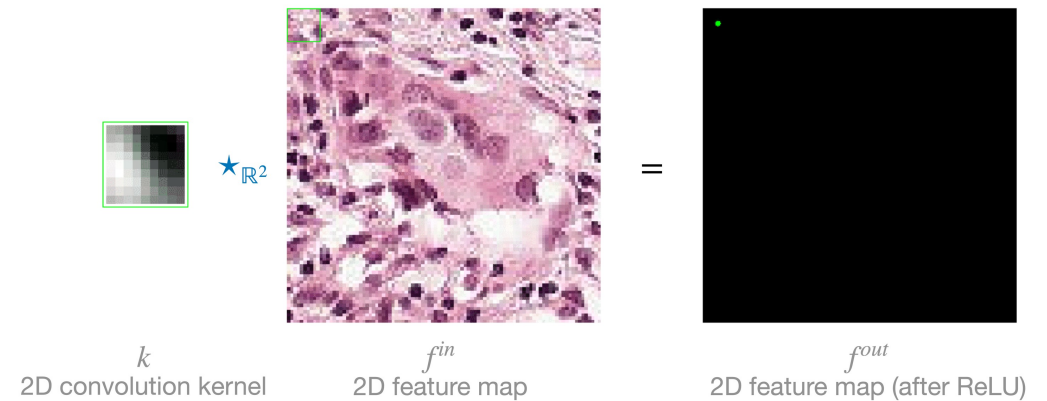
3. Steerable kernels

4. Typed features and 3D networks

5. Implementation and verification (optional)

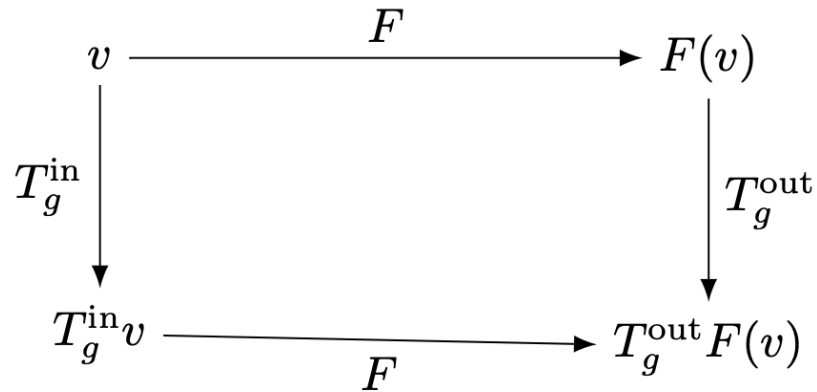
Rotation-aware CNNs

- Classification: preserve the image class.
- Segmentation: move and rotate the mask.
- Intermediate features: retain location and orientation.



Equivariance: two routes, one answer

$$F(T_g^{\text{in}}v) = T_g^{\text{out}}F(v)$$



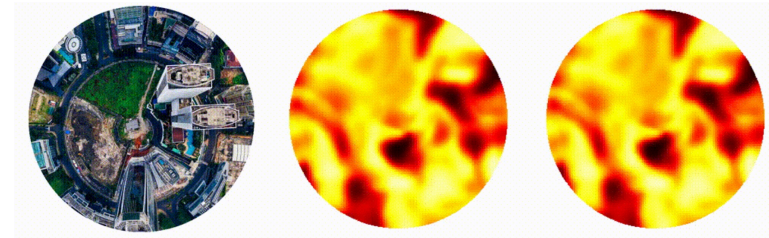
Invariance is the special case with an unchanged output.

Augmentation and architectural constraints

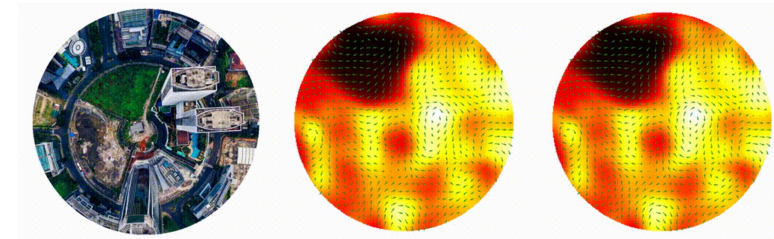
[Play animation at 7:46](#) 

Augmentation exposes the learner to transformed examples.

CNN



Group-equivariant CNN



Equivariant layers enforce the transformation rule for every weight setting.

Benefit depends on whether the task symmetry is appropriate.

Outline

1. Groups and actions

- Choose the CNN symmetry
- **Motion groups and actions**
- How CNN features transform
- Symmetrization by averaging
- Image domains and homogeneous spaces
- Normality and quotient groups

2. Group convolutions

3. Steerable kernels

4. Typed features and 3D networks

5. Implementation and verification (optional)

Group: formal definition

A **group** is a set with a **group product** satisfying four rules.

$$G \times G \rightarrow G, \quad (g, h) \mapsto gh$$

1. Closure

$$g, h \in G \implies gh \in G$$

2. Associativity

$$\forall g, h, k \in G : \quad (gh)k = g(hk)$$

3. Identity

$$\exists e \in G, \forall g \in G : \quad eg = ge = g$$

4. Inverses

$$\forall g \in G, \exists g^{-1} \in G : \quad gg^{-1} = g^{-1}g = e$$

Group actions

An **action** specifies how a transformation moves a point within the same domain.

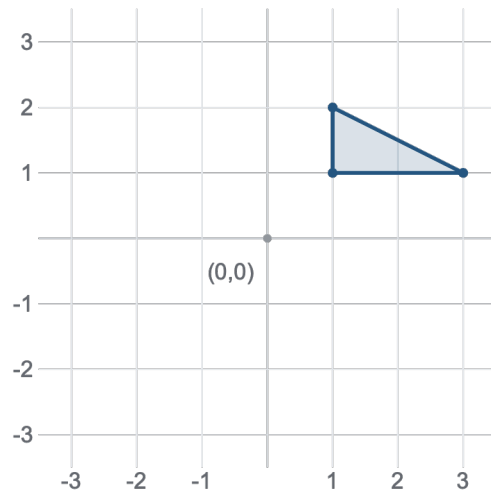
$$\alpha : G \times X \rightarrow X, \quad \alpha(g, x) = g \cdot x$$

$$e \cdot x = x, \quad g \cdot (h \cdot x) = (gh) \cdot x$$

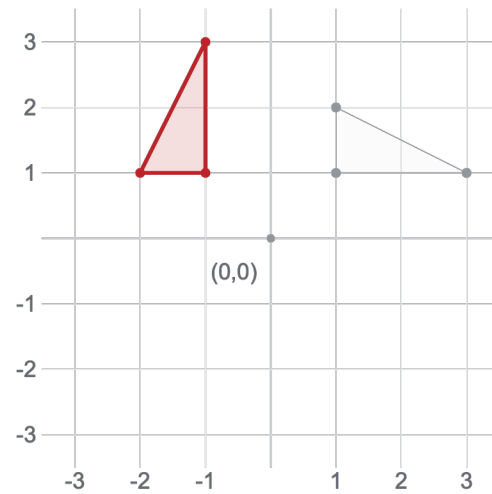
The **codomain** requires every transformed point to remain in the domain.

Four grid orientations

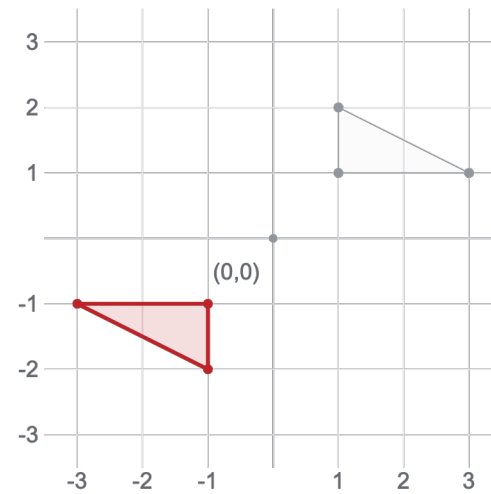
$$C_4 = \{R_0, R_1, R_2, R_3\}, \quad R_r = R(r\pi/2)$$



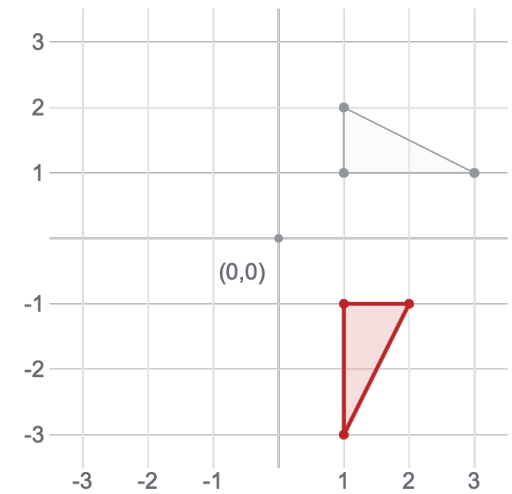
R_0



R_1



R_2

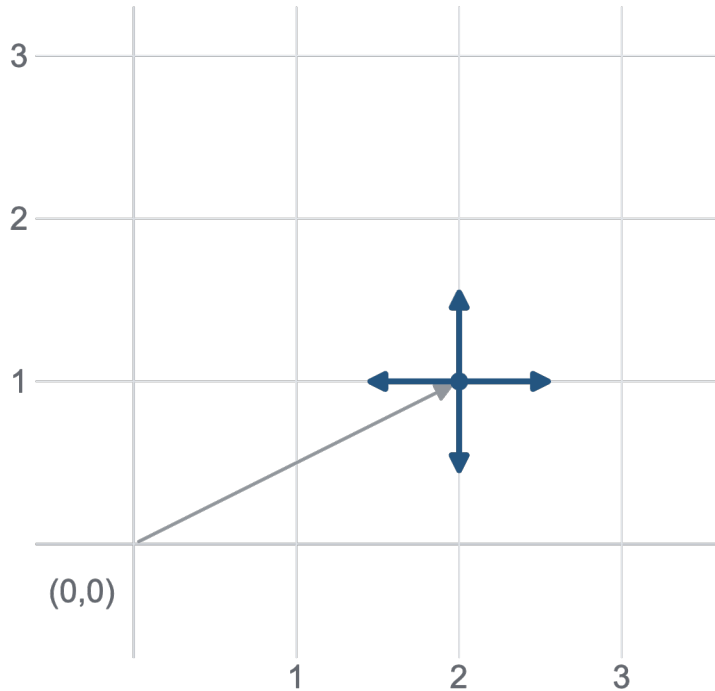


R_3

$$R_{(r+s) \bmod 4} = R_r R_s, \quad R_r^{-1} = R_{(-r) \bmod 4}$$

Grid motions: position and orientation

A pose stores a translation vector and a rotation matrix.



$$p4 = \mathbb{Z}^2 \rtimes C_4$$

$$g = (t, R), \quad g \cdot x = t + Rx$$

$$(t, R)(u, S) = (t + Ru, RS)$$

$$(t, R)^{-1} = (-R^{-1}t, R^{-1})$$

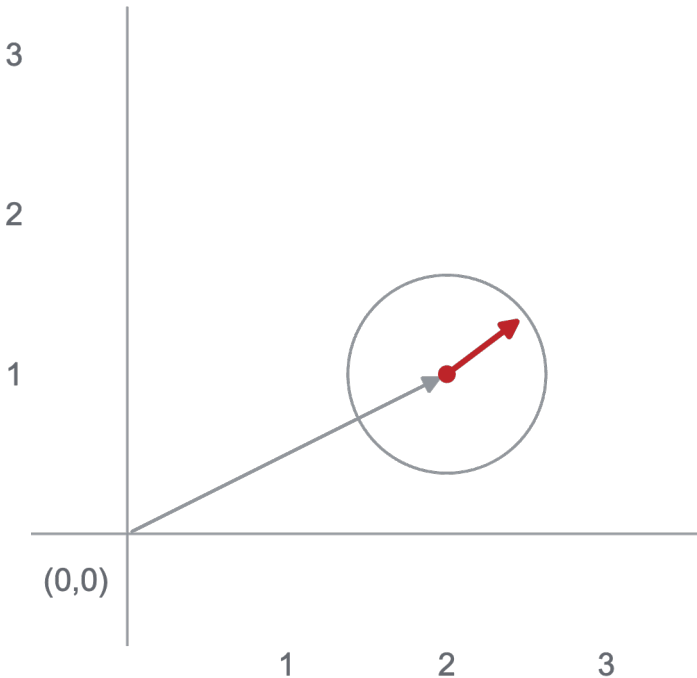
Continuous planar motions

The same pose structure permits real translations and every planar angle.

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

$$R(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$$

$$g = (t, R(\theta)), \quad g \cdot x = t + R(\theta)x$$

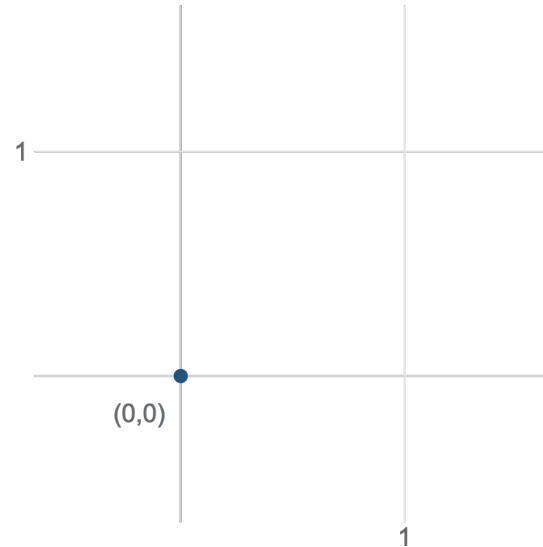


Sampled pixels require interpolation for general angles.

Composition order

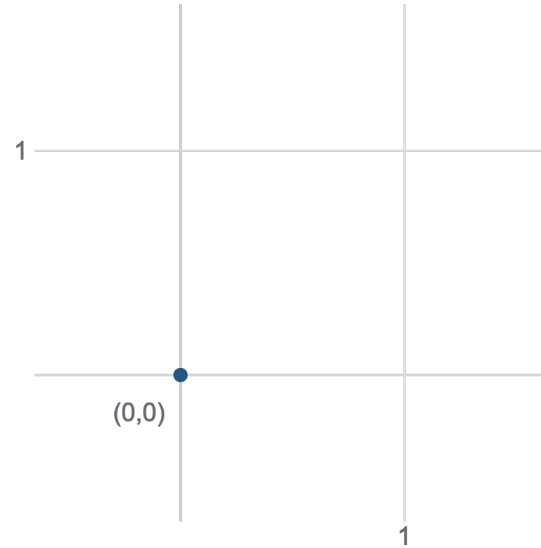
$$g = ((0, 0), R_1), \quad h = ((1, 0), I)$$

Translate, then rotate



$$x = (0, 0)$$

Rotate, then translate

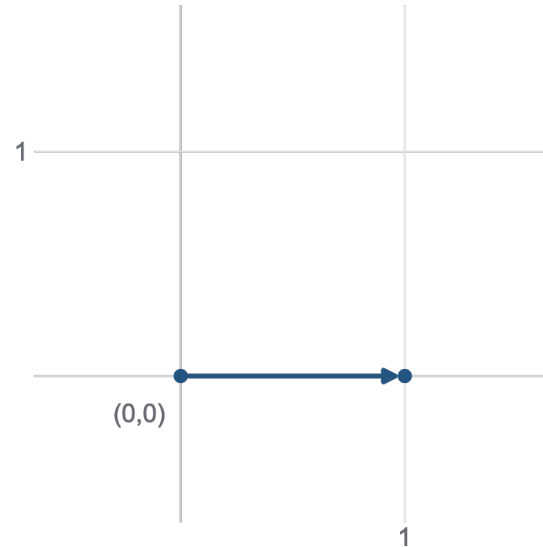


$$x = (0, 0)$$

Composition order

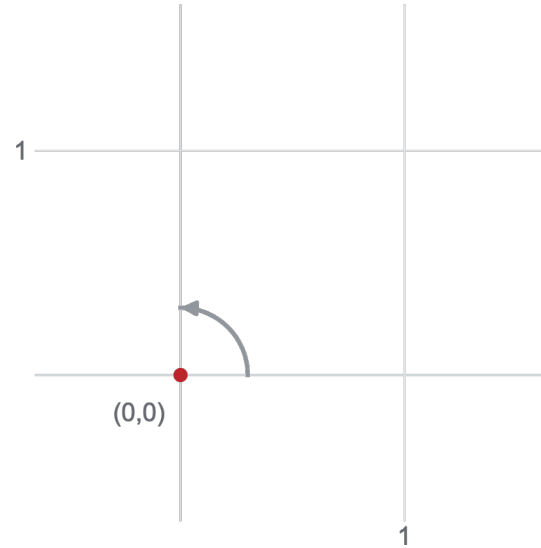
$$g = ((0, 0), R_1), \quad h = ((1, 0), I)$$

Translate, then rotate



$$h \cdot (0, 0) = (1, 0)$$

Rotate, then translate

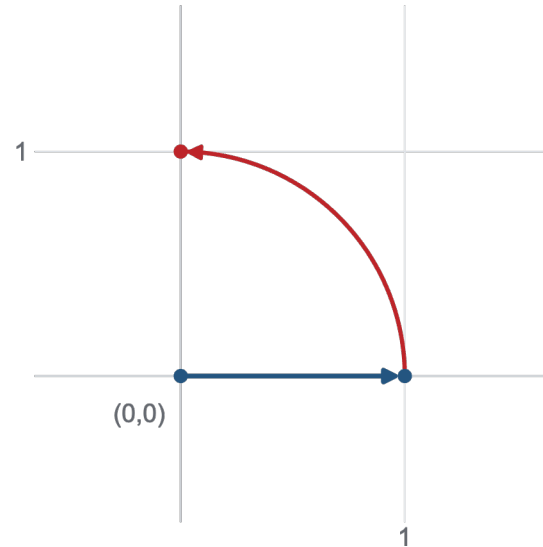


$$g \cdot (0, 0) = (0, 0)$$

Composition order

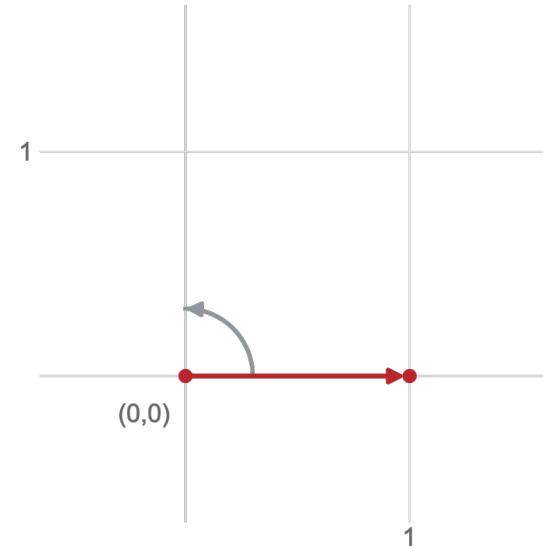
$$g = ((0, 0), R_1), \quad h = ((1, 0), I)$$

Translate, then rotate



$$(gh) \cdot (0, 0) = (0, 1)$$

Rotate, then translate



$$(hg) \cdot (0, 0) = (1, 0)$$

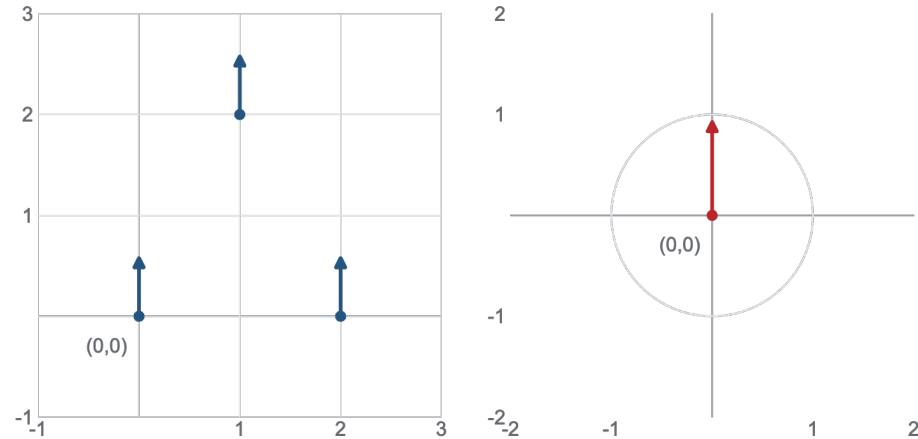
Orientation projection

Discard the location of a pose and retain its rotation.

$$\pi : G \rightarrow K, \quad \pi((t, R)) = R$$

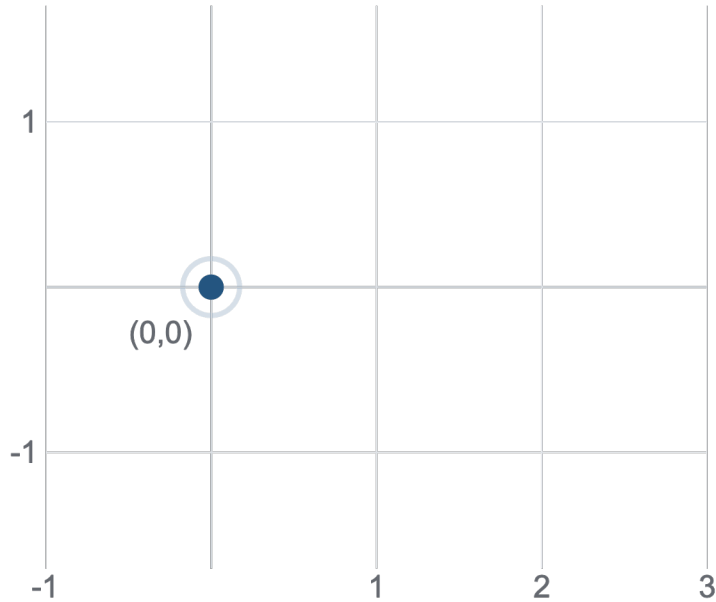
$$p4 : K = C_4, \quad \text{SE}(2) : K = \text{SO}(2)$$

Different locations Same rotation



Spatial motion and inverse sampling

A stored scalar value moves with its spatial location.



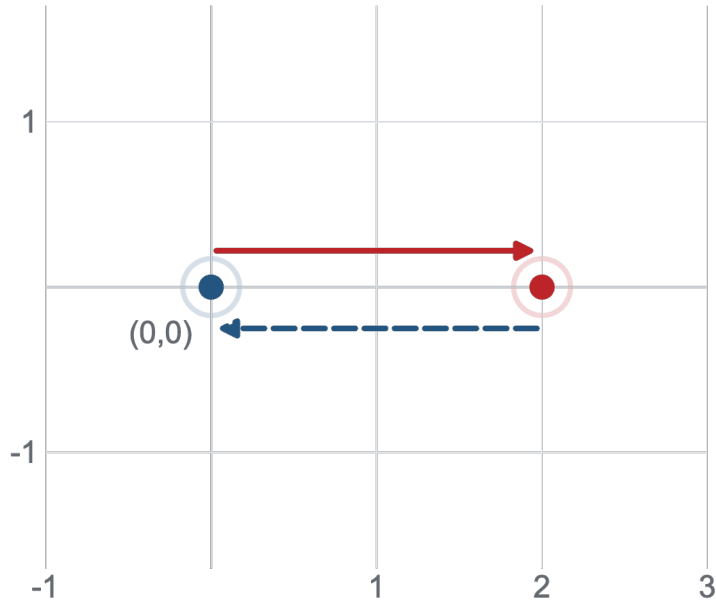
$$g = ((2, 0), I), \quad g \cdot (0, 0) = (2, 0)$$

$$[T_g f](y) = f(g^{-1} \cdot y)$$

The original peak sits at the origin.

Spatial motion and inverse sampling

A stored scalar value moves with its spatial location.



$$g = ((2, 0), I), \quad g \cdot (0, 0) = (2, 0)$$

$$[T_g f](y) = f(g^{-1} \cdot y)$$

$$[T_g f](2, 0) = f(0, 0)$$

Read the old coordinate to fill the new one.

Left-regular representation and composition

Definition: left-regular representation

A function stores one vector at each group element.

$$z : G \rightarrow \mathbb{R}^C, \quad [L_a z](g) = z(a^{-1}g)$$

Each operator is linear and invertible. It reindexes the whole function.

$$[L_a z](ag) = z(g), \quad a, g \in G$$

Definition: representation law (homomorphism)

Operator composition respects the group product.

$$L_a L_b = L_{ab}, \quad L_e = I, \quad L_a^{-1} = L_{a^{-1}}$$

Apply the rightmost operator first, as with the **group action**.

Outline

1. Groups and actions

- Choose the CNN symmetry
- Motion groups and actions
- **How CNN features transform**
- Symmetrization by averaging
- Image domains and homogeneous spaces
- Normality and quotient groups

2. Group convolutions

3. Steerable kernels

4. Typed features and 3D networks

5. Implementation and verification (optional)

Representations and channel types

The **whole-field representation** obeys the **representation law (homomorphism)**.

$$\sigma : G \rightarrow \text{GL}(V), \quad \sigma(gh) = \sigma(g)\sigma(h)$$

A feature map stores a vector at each location.

Rotating the input moves locations and transforms each stored vector.

$$\rho : K \rightarrow \text{GL}(\mathbb{R}^C) \quad K \text{ is the } \mathbf{orientation\ group}.$$

Next: two examples with different channel meanings.

1. Planar velocity

$$\rho(R) = R$$

Rotate two velocity components.

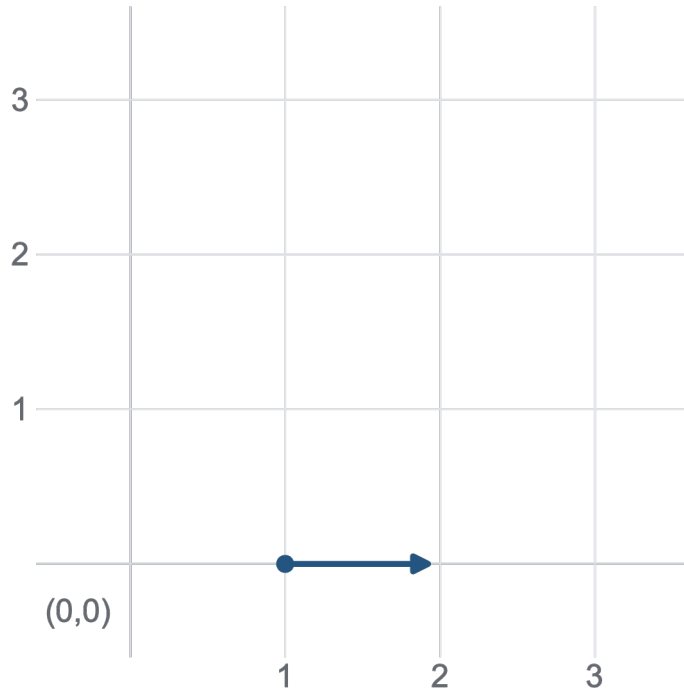
2. Directional contrasts

$$\rho(R_1) = P$$

Permute four orientation channels.

Example 1: a planar velocity field

$$g = ((2, 1), R_1), \quad x = (1, 0), \quad f(x) = (1, 0)^\top$$

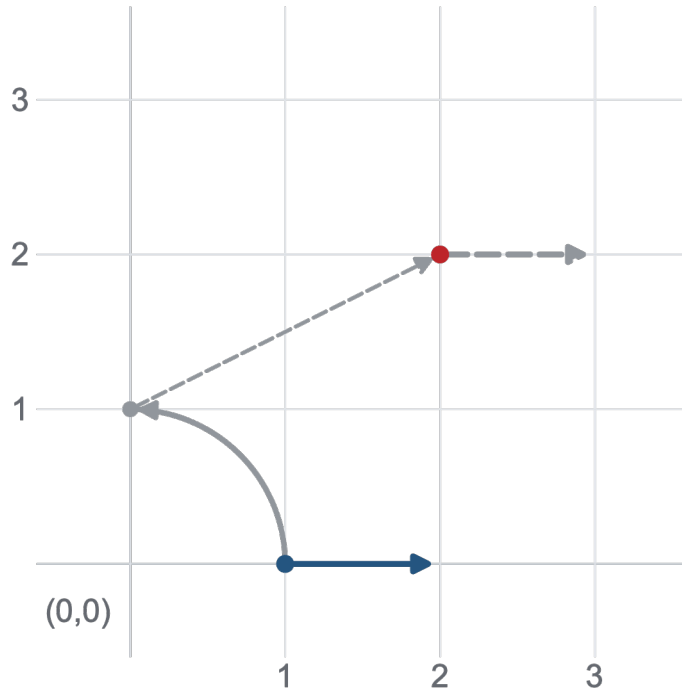


1. Original location and velocity

$$f(x) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Example 1: a planar velocity field

$$g = ((2, 1), R_1), \quad x = (1, 0), \quad f(x) = (1, 0)^\top$$



2. Rotate the location, then translate

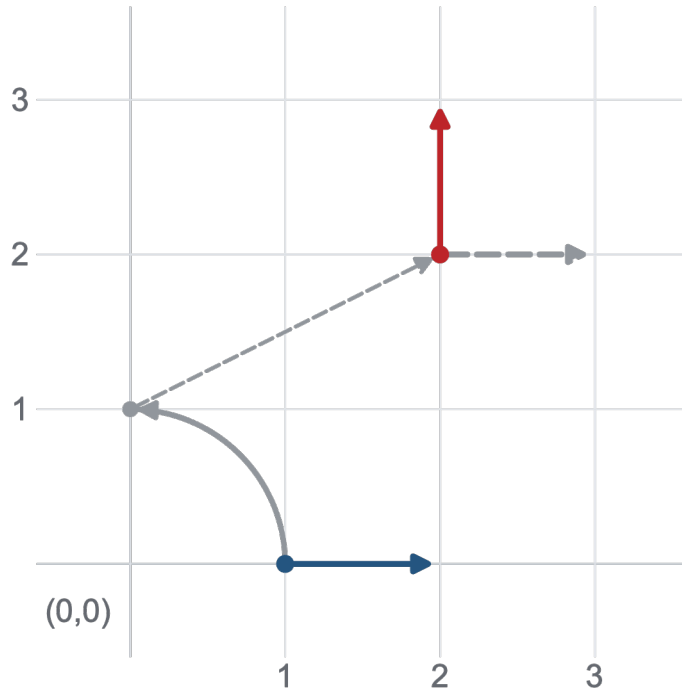
$$R_1 x = (0, 1)$$

$$y = R_1 x + (2, 1) = (2, 2)$$

The carried velocity still needs its channel rotation.

Example 1: a planar velocity field

$$g = ((2, 1), R_1), \quad x = (1, 0), \quad f(x) = (1, 0)^\top$$



3. Rotate the velocity components

$$\tilde{f}(y) = R_1 f(x) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$[T_g f](y) = \rho(\pi(g)) f(g^{-1} \cdot y)$$

For scalar fields such as temperature, only the location changes.

Example 2: directional contrasts

At each location, encode neighboring intensity contrasts as a four-vector.

$$z(t, R_r) = f(t + R_r e_1) - f(t), \quad R_r = R(r\pi/2), \quad e_1 = (1, 0)^\top$$

$$\mathbf{z}(t) = (z(t, R_0), z(t, R_1), z(t, R_2), z(t, R_3))^\top$$

Local pixel values

0	1	0
2	0	4
0	3	0

$$t = (2, 1)$$

Start from one scalar intensity per pixel.

Subtract the center from each neighbor.

Order: right, up, left, down.

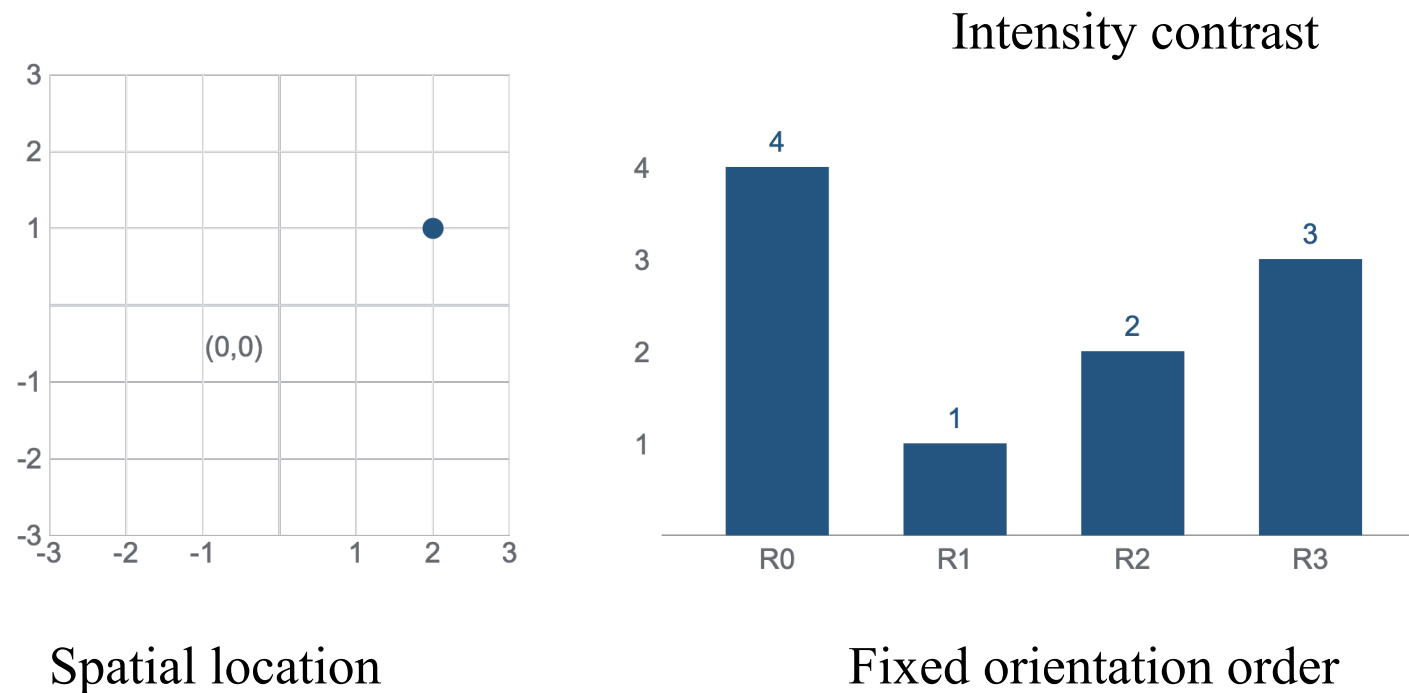
$$\mathbf{z}((2, 1)) = (4, 1, 2, 3)^\top$$

Every location uses its own neighbors.

Four orientation channels at each location

Scalar pose responses form a four-vector at each grid location.

$$\mathbf{z}(t) = (z(t, R_0), z(t, R_1), z(t, R_2), z(t, R_3))^{\top}$$



$$\mathbf{z}((2, 1)) = (4, 1, 2, 3)^{\top}$$

Rotating the image moves the contrast pattern

Rotate the entire image counterclockwise by a quarter-turn.

$$a = ((0, 0), R_1), \quad \tilde{f}(x) = f(a^{-1} \cdot x)$$

Before rotation

0	1	0
2	0	4
0	3	0

$$t = (2, 1)$$

After rotation

0	4	0
1	0	3
0	2	0

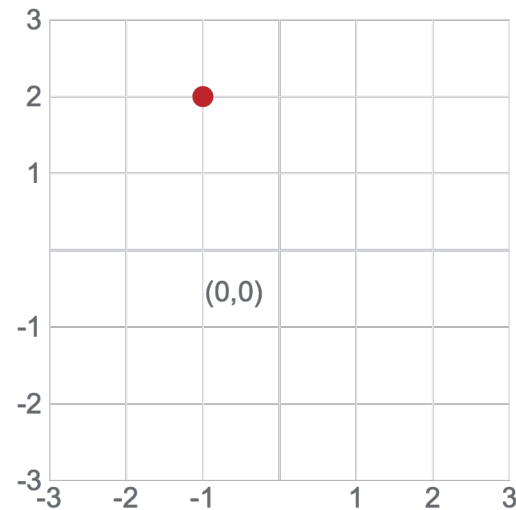
$$R_1 t = (-1, 2)$$

The right neighbor becomes the upper neighbor. Its contrast stays 4.

Rotating the image shifts the orientation channels

Keep orientation labels fixed: right, up, left, down.

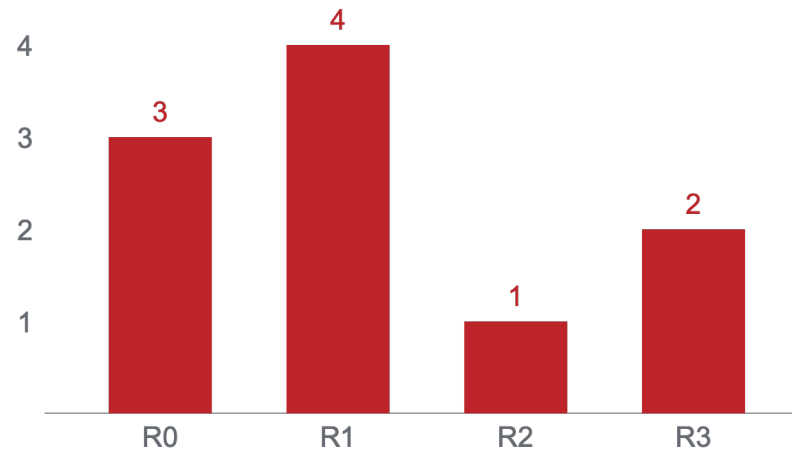
$$([L_a z](R_1 t, R_0), \dots, [L_a z](R_1 t, R_3))^{\top} = (3, 4, 1, 2)^{\top}$$



Moved spatial location

$$z(t, R_0) = 4 \implies [L_a z](R_1 t, R_1) = 4$$

Intensity contrast



Fixed orientation order

The left-regular action in vector storage

Stack the transformed responses in their fixed orientation order.

The same **left-regular action** becomes a cyclic permutation of the entries.

$$\begin{pmatrix} [L_a z](R_1 t, R_0) \\ [L_a z](R_1 t, R_1) \\ [L_a z](R_1 t, R_2) \\ [L_a z](R_1 t, R_3) \end{pmatrix} = P \mathbf{z}(t), \quad P = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Fill each new orientation channel from the one that rotated into it.

$$[L_a z](R_1 t, R_r) = z(t, R_{(r-1) \bmod 4})$$

This reindexing follows from the directional detector definition.

Outline

1. Groups and actions

- Choose the CNN symmetry
- Motion groups and actions
- How CNN features transform
- **Symmetrization by averaging**
- Image domains and homogeneous spaces
- Normality and quotient groups

2. Group convolutions

3. Steerable kernels

4. Typed features and 3D networks

5. Implementation and verification (optional)

Averaging: invariance or equivariance?

For a **finite rotation group**, choose the average by the output transformation.

Classification: average class scores

$$F_{\text{inv}}(v) = \frac{1}{|G|} \sum_{g \in G} F(\sigma_{\text{in}}(g)v)$$

Rotating the input permutes the terms, so the result is **invariant**.

Segmentation: align masks, then average

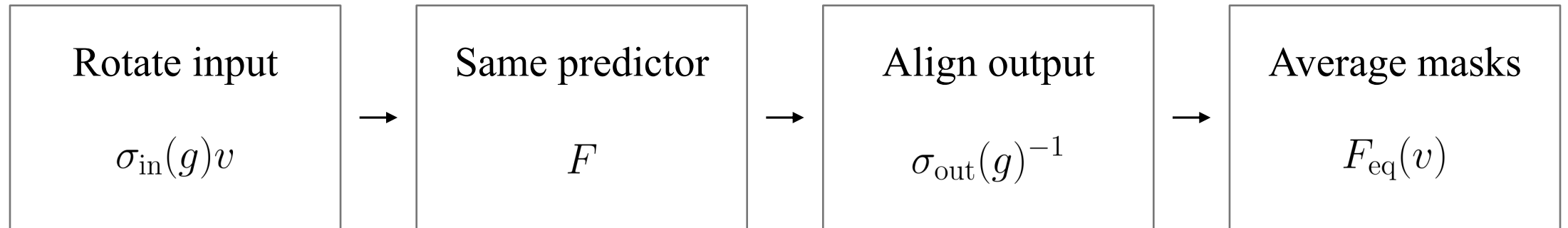
$$F_{\text{eq}}(v) = \frac{1}{|G|} \sum_{g \in G} \sigma_{\text{out}}(g)^{-1} F(\sigma_{\text{in}}(g)v)$$

Undo each output rotation to compare corresponding mask pixels.

The aligned prediction rotates with the input: this is **equivariance**.

Symmetrization as a CNN baseline

Segmentation example: align every predicted mask to the original image.



Aligned masks describe the same original pixels.

Four quarter-turns require four evaluations of the same predictor.

Use this baseline to compare accuracy and runtime with equivariant CNNs.

Finite-group symmetrization

Use a **finite group** with **linear input and output representations**.

$$F_{\text{eq}}(v) = \frac{1}{|G|} \sum_{g \in G} \sigma_{\text{out}}(g)^{-1} F(\sigma_{\text{in}}(g)v)$$

Meaning of one aligned prediction

v	Complete model input, such as the whole image
$\sigma_{\text{in}}(g)$	Transform the input
F	Apply the base predictor
$\sigma_{\text{out}}(g)^{-1}$	Undo the output transformation

Output linearity lets its transformation pass through the average.

Symmetrization guarantees equivariance

Recall the aligned average:

$$F_{\text{eq}}(v) = \frac{1}{|G|} \sum_{g \in G} \sigma_{\text{out}}(g)^{-1} F(\sigma_{\text{in}}(g)v)$$

Theorem: finite-group symmetrization

For **linear input and output representations**, the average is equivariant:

$$F_{\text{eq}}(\sigma_{\text{in}}(h)v) = \sigma_{\text{out}}(h)F_{\text{eq}}(v), \quad h \in G$$

An already **equivariant predictor** is unchanged by averaging.

$$F_{\text{eq}} = F$$

Assume exact transformations and the complete **finite-group average**.

Symmetrization proof: expand the average

1. Expand the aligned average at a transformed input.

$$F_{\text{eq}}(\sigma_{\text{in}}(h)v) = \frac{1}{|G|} \sum_{g \in G} \sigma_{\text{out}}(g)^{-1} F(\sigma_{\text{in}}(g)\sigma_{\text{in}}(h)v)$$

2. Use the representation law (homomorphism property).

$$\sigma_{\text{in}}(g)\sigma_{\text{in}}(h) = \sigma_{\text{in}}(gh)$$

$$F_{\text{eq}}(\sigma_{\text{in}}(h)v) = \frac{1}{|G|} \sum_{g \in G} \sigma_{\text{out}}(g)^{-1} F(\sigma_{\text{in}}(gh)v)$$

Only the input transformations are combined.

The predictor may be nonlinear and non-equivariant.

Symmetrization proof: reindex the sum

3. Reindex the same finite sum.

$$k = gh, \quad g = kh^{-1}$$

$$\sigma_{\text{out}}(kh^{-1})^{-1} = \sigma_{\text{out}}(h)\sigma_{\text{out}}(k)^{-1}$$

4. Factor out the common output transformation.

$$\begin{aligned} F_{\text{eq}}(\sigma_{\text{in}}(h)v) &= \sigma_{\text{out}}(h) \frac{1}{|G|} \sum_k \sigma_{\text{out}}(k)^{-1} F(\sigma_{\text{in}}(k)v) \\ &= \sigma_{\text{out}}(h) F_{\text{eq}}(v) \end{aligned}$$

Right composition permutes the group elements, so no terms change.

Outline

1. Groups and actions

- Choose the CNN symmetry
- Motion groups and actions
- How CNN features transform
- Symmetrization by averaging
- **Image domains and homogeneous spaces**
- Normality and quotient groups

2. Group convolutions

3. Steerable kernels

4. Typed features and 3D networks

5. Implementation and verification (optional)

Subgroups and homomorphisms

A **subgroup** contains the identity and is closed under products and inverses.

$$H = \{((0, 0), R) : R \in K\}, \quad C_4 \leq p4, \quad \text{SO}(2) \leq \text{SE}(2)$$

Subgroup notation $H \leq G$ reads “is a **subgroup** of.”

Orientation projection $\pi : G \rightarrow K, \quad \pi((t, R)) = R$

A **homomorphism** preserves the **group product**.

$$\pi(gh) = \pi((t + Ru, RS)) = RS = \pi(g)\pi(h)$$

A bijective **homomorphism** is an **isomorphism**.

G-spaces: domains with an action

A **G-space** is a set equipped with a **group action**.

$$\alpha : G \times X \longrightarrow X, \quad (g, x) \longmapsto g \cdot x$$

The identity fixes each point; group products compose the motions.

$$\begin{aligned} e \cdot x &= x, & g \cdot (h \cdot x) &= (gh) \cdot x \\ g, h &\in G, & x \in X, & \quad e \in G \end{aligned}$$

For image domains, the points are locations and the actions are motions.

$$g = (t, R), \quad g \cdot x = t + Rx$$

The set and the action belong together in this definition.

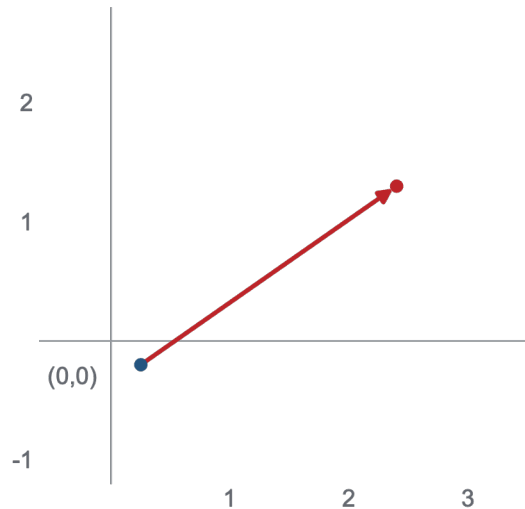
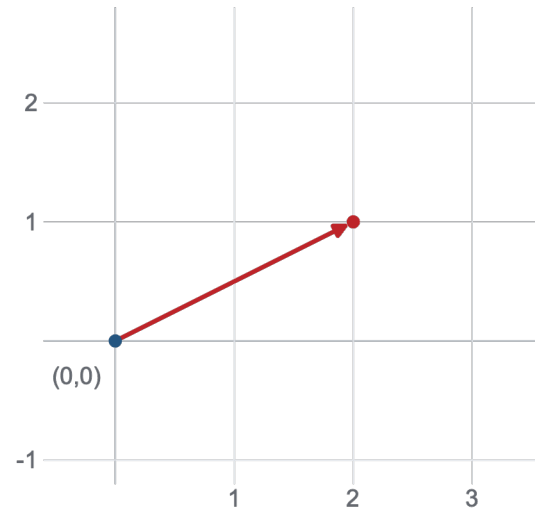
Homogeneous image domains

A nonempty domain is **homogeneous** when any point can reach any other.

$$\forall x, y \in X, \quad \exists g \in G : \quad g \cdot x = y$$

$$g = (y - x, I) : \quad g \cdot x = y$$

$$G = p4, \quad X = \mathbb{Z}^2 \quad G = \text{SE}(2), \quad X = \mathbb{R}^2$$



Orbits and stabilizers

Choose one point in the domain.

$$x_0 \in X$$

Orbit: reachable points

$$\text{Orb}_G(x_0) = \{g \cdot x_0 : g \in G\} \subseteq X$$

A set of points in the domain.

Stabilizer: motions fixing the point

$$\text{Stab}_G(x_0) = \{g \in G : g \cdot x_0 = x_0\} \leq G$$

A **subgroup** of the motion group.

$$x_0 = (0, 0) : \quad \text{Orb}_G(x_0) = X$$

Translations reach every location in either image domain.

$$G = p4, \quad X = \mathbb{Z}^2$$

or

$$G = \text{SE}(2), \quad X = \mathbb{R}^2$$

The origin stabilizer

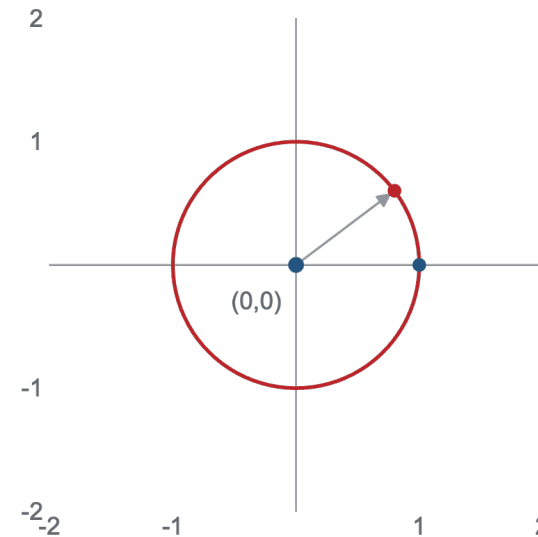
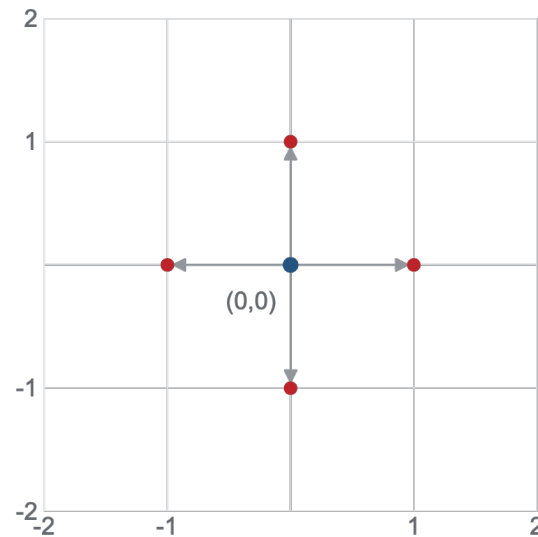
Exactly the rotations about the origin fix the origin.

$$H = \text{Stab}_G((0, 0)) = \{((0, 0), R) : R \in K\}$$

$$G = p4, \quad K = C_4 \quad G = \text{SE}(2), \quad K = \text{SO}(2)$$

The red **orbit** is

$$\text{Orb}_H((1, 0))$$



Left: four points.

Right: a circle.

A **stabilizer** fixes the chosen point; it can move other points.

Left cosets and quotient spaces

A **left coset** collects equivalent representatives of one label.

$$H \leq G, \quad g \in G : \quad gH = \{gh : h \in H\}$$

Any element of the coset can serve as its **representative**.

$$(gh)H = gH \quad (h \in H)$$

$$g_1H = g_2H \quad \Longleftrightarrow \quad g_1^{-1}g_2 \in H$$

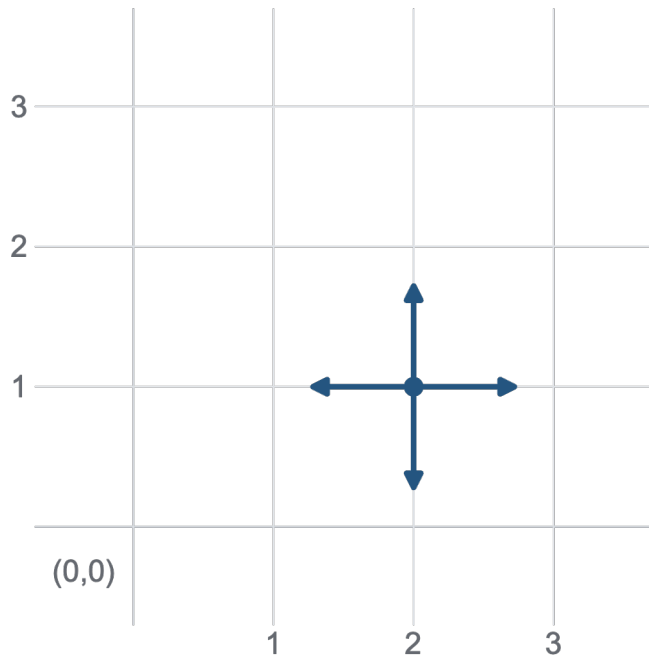
The **quotient set** contains these cosets as its elements.

$$G/H = \{gH : g \in G\}$$

For origin rotations, one coset keeps location and forgets orientation.

One location, many poses

A **group product** on the right changes orientation while keeping the location.



$$t = (2, 1), \quad H = \{((0, 0), S) : S \in C_4\}$$

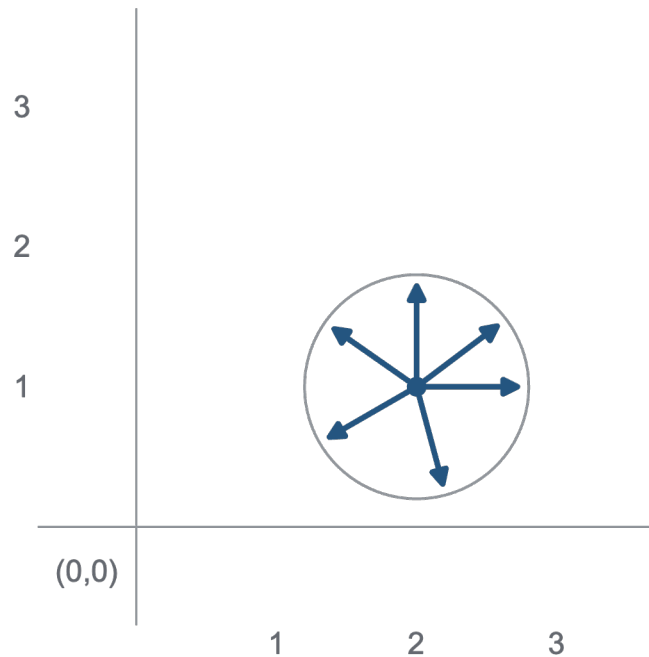
$$g = (t, R), \quad h = ((0, 0), S), \quad gh = (t, RS)$$

$$gH = \{(t, Q) : Q \in C_4\}$$

Four poses share one grid location.

One location, many poses

A **group product** on the right changes orientation while keeping the location.



$$t = (2, 1), \quad H = \{((0, 0), S) : S \in \text{SO}(2)\}$$

$$g = (t, R), \quad h = ((0, 0), S), \quad gh = (t, RS)$$

$$gH = \{(t, Q) : Q \in \text{SO}(2)\}$$

Every angle belongs to the same location coset.

Cosets as location labels

Assume a **transitive action**. Fix a reference point and its **stabilizer**.

$$H = \text{Stab}_G(x_0), \quad \Phi : G/H \longrightarrow X, \quad \Phi(gH) = g \cdot x_0$$

Each **coset** names one location, regardless of the chosen **representative**.

$$g_1H = g_2H \iff g_1 \cdot x_0 = g_2 \cdot x_0$$

For both CNN motion groups, use rotations about the origin as the **stabilizer**.

$$x_0 = (0, 0) : \quad \Phi((t, R)H) = t$$

Every heading at this pixel names the same location coset.

$$\Phi(((2, 1), R)H) = (2, 1), \quad R \in K$$

$$p4/C_4 \simeq \mathbb{Z}^2, \quad \text{SE}(2)/\text{SO}(2) \simeq \mathbb{R}^2$$

Isomorphism of group spaces

A **G-space isomorphism** is a **bijection** preserving the **group action**.

$$a \cdot (gH) = (ag)H, \quad a \in G$$

The correspondence reads the image location of a coset.

$$\Phi(a \cdot (gH)) = a \cdot \Phi(gH)$$

In either motion group, a quarter-turn moves the whole location coset.

$$g = ((2, 1), R), \quad a = ((0, 0), R_1), \quad \Phi(gH) = (2, 1)$$

$$\Phi((ag)H) = (-1, 2) = a \cdot \Phi(gH)$$

Coset labels and pixel coordinates obey the same motion rule.

Outline

1. Groups and actions

- Choose the CNN symmetry
- Motion groups and actions
- How CNN features transform
- Symmetrization by averaging
- Image domains and homogeneous spaces
- **Normality and quotient groups**

2. Group convolutions

3. Steerable kernels

4. Typed features and 3D networks

5. Implementation and verification (optional)

Two poses, one location label

The origin rotation **subgroup** identifies all headings at the origin.

$$H = \text{Stab}_G((0, 0)) = \{((0, 0), R) : R \in K\}$$

Identity pose

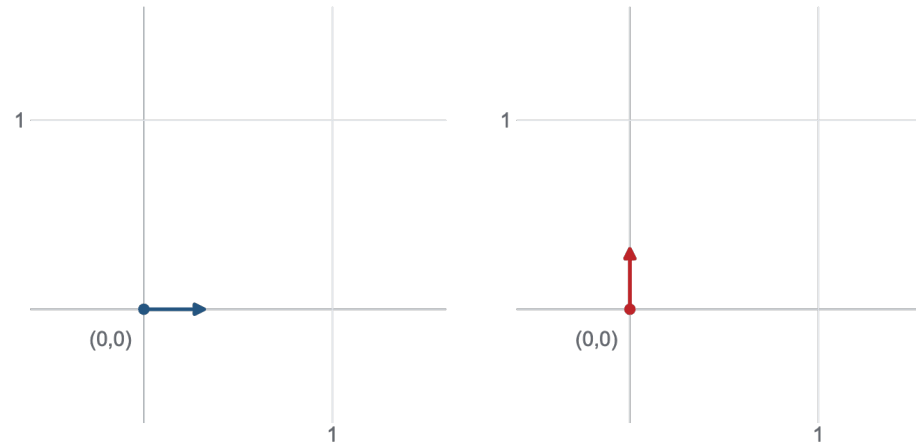
$$e = ((0, 0), I)$$

Quarter-turn pose

$$q = ((0, 0), R_1)$$

Quarter-turn matrix

$$R_1 = R(\pi/2)$$



Both poses fix the origin.

$$eH = qH = H$$

A proposed product on coset labels

We are testing the **proposed operation** on **coset labels**:

$$(g_1H) \star (g_2H) := (g_1g_2)H$$

$$\tau = ((1, 0), I), \quad eH = qH = H, \quad H \neq \tau H$$

Representatives chosen	Input cosets	Result from the rule
$(g_1, g_2) = (e, \tau)$	$(H, \tau H)$	$(e\tau)H = \tau H$ Location $(1, 0)$
$(g_1, g_2) = (q, \tau)$	$(H, \tau H)$	$(q\tau)H$ Location $(0, 1)$

A **well-defined** rule must give the same output for the same input pair.

The proposed rule gives two different outputs

Required for well-definedness:

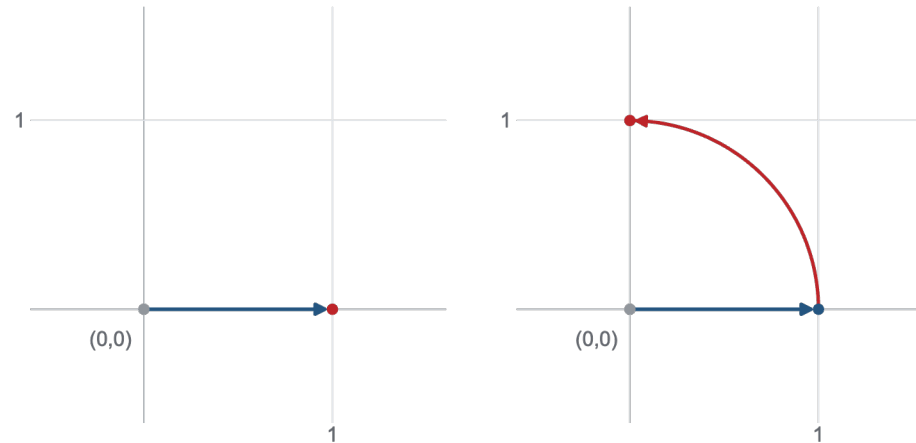
$$eH = qH \implies (e\tau)H = (q\tau)H$$

Identity outer pose Quarter-turn outer pose

$$e\tau = ((1, 0), I) \qquad q\tau = ((0, 1), R_1)$$

Actual outputs:

$$(e\tau)H \neq (q\tau)H$$



Same input pair.
Different results.

The rule needs
the discarded
heading.

This rule cannot define a product on location cosets.

Normal subgroups and quotient groups

Cosets already describe locations. Can we also take a product of the coset labels?

A **normal subgroup** contains every version of its operations after a frame change.

$$gHg^{-1} = \{ghg^{-1} : h \in H\} = H \quad (\forall g \in G), \quad H \trianglelefteq G$$

The rule is independent of **representatives** for every pair exactly when this holds.

$$(g_1H) \star (g_2H) = (g_1g_2)H$$

Conjugation changes the frame of a **subgroup** operation.

With this product, the **quotient set** becomes a **quotient group**.

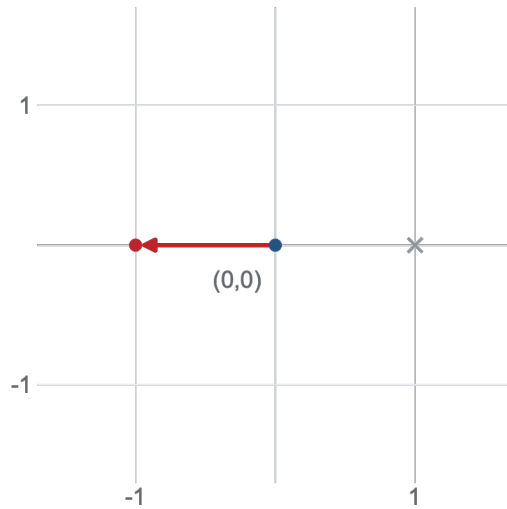
Origin rotations under a frame change

Test the origin-rotation **subgroup** in both CNN motion groups.

$$H = \{((0, 0), Q) : Q \in K\}, \quad (G, K) \in \{(p4, C_4), (SE(2), SO(2))\}$$

Shift the frame right by one. The local operation is a quarter-turn.

$$g = ((1, 0), I), \quad h = ((0, 0), R_1) \in H$$



1. Undo the frame shift

$$(0, 0) \xrightarrow{g^{-1}} (-1, 0)$$

Track the origin to check whether the final operation still fixes it.

Origin rotations under a frame change

Test the origin-rotation **subgroup** in both CNN motion groups.

$$H = \{((0, 0), Q) : Q \in K\}, \quad (G, K) \in \{(p4, C_4), (SE(2), SO(2))\}$$

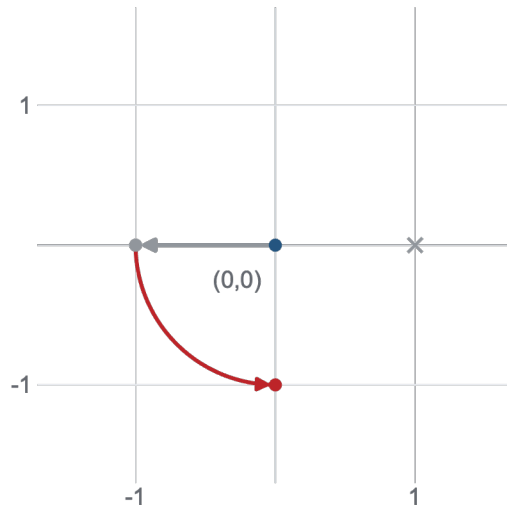
Shift the frame right by one. The local operation is a quarter-turn.

$$g = ((1, 0), I), \quad h = ((0, 0), R_1) \in H$$

2. Apply the local rotation

$$(-1, 0) \xrightarrow{h} (0, -1)$$

Rotate in the shifted coordinates. We still have to restore the frame.



Origin rotations under a frame change

Test the origin-rotation **subgroup** in both CNN motion groups.

$$H = \{((0, 0), Q) : Q \in K\}, \quad (G, K) \in \{(p4, C_4), (\text{SE}(2), \text{SO}(2))\}$$

Shift the frame right by one. The local operation is a quarter-turn.

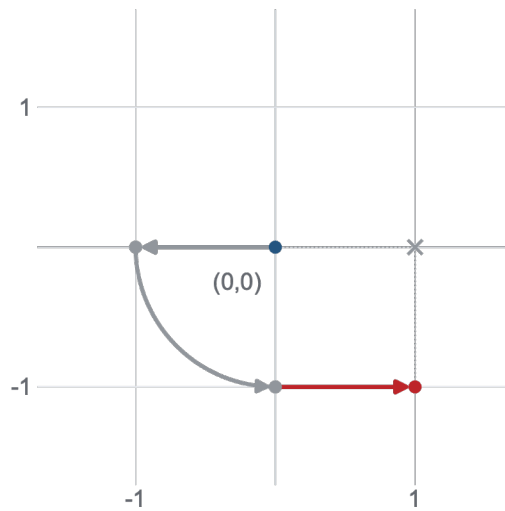
$$g = ((1, 0), I), \quad h = ((0, 0), R_1) \in H$$

3. Restore the frame

$$ghg^{-1} = ((1, -1), R_1) \notin H$$

The resulting rotation fixes the shifted center and moves the origin.

Result: the rotation stabilizer is non-normal.



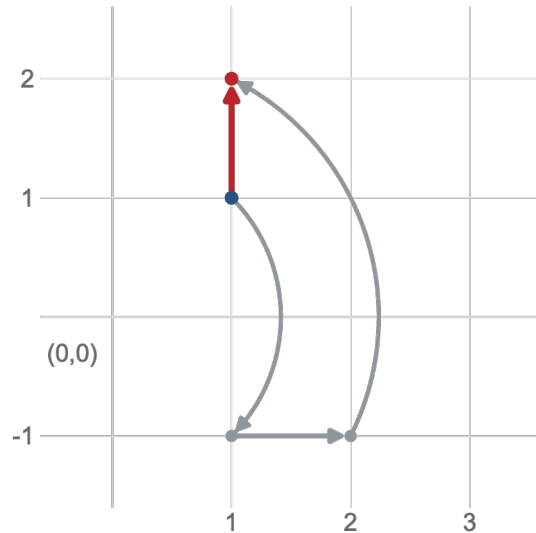
Translations under a frame change

We now test the **translation subgroup**.

Choose a translation in the subgroup and a rotation for the **frame change**.

$$N = \{(u, I) : u \in X\}, \quad X \in \{\mathbb{Z}^2, \mathbb{R}^2\}$$

$$g = ((0, 0), R_1), \quad h = ((1, 0), I) \in N$$



Track one point, in inverse-first order.

$$(1, 1) \xrightarrow{g^{-1}} (1, -1) \xrightarrow{h} (2, -1) \xrightarrow{g} (1, 2)$$

$$ghg^{-1} = ((0, 1), I) \in N$$

A rightward translation becomes an upward translation.

The translation subgroup is normal

The same result must hold for every pose and every translation.

$$g = (t, R), \quad h = (u, I) \in N, \quad t, u \in X, \quad R \in K$$

$$(ghg^{-1}) \cdot x = t + R(R^{-1}(x - t) + u) = x + Ru$$

Rotations map the set of allowed displacement vectors onto itself.

$$ghg^{-1} = (Ru, I), \quad gNg^{-1} = N, \quad N \trianglelefteq G$$

CNN interpretation

Discarding location still leaves orientation labels that compose consistently.

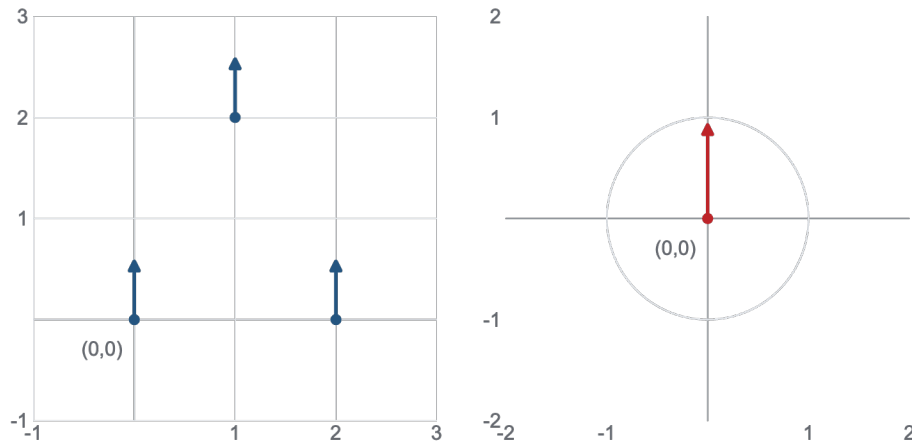
$$p4/N \simeq C_4, \quad \text{SE}(2)/N \simeq \text{SO}(2)$$

The quotient that retains orientation

A pose has a location and an orientation. Keep only its orientation.

$$\pi : G \rightarrow K, \quad \pi((t, R)) = R$$

Different locations, same orientation



Orientation labels compose.

$$\pi((t, R)(u, S)) = RS$$

$$\pi(g_1 g_2) = \pi(g_1) \pi(g_2)$$

The projection is a **homomorphism**: it preserves composition.

$$(t, R)N = \{(w, R) : w \in X\}$$

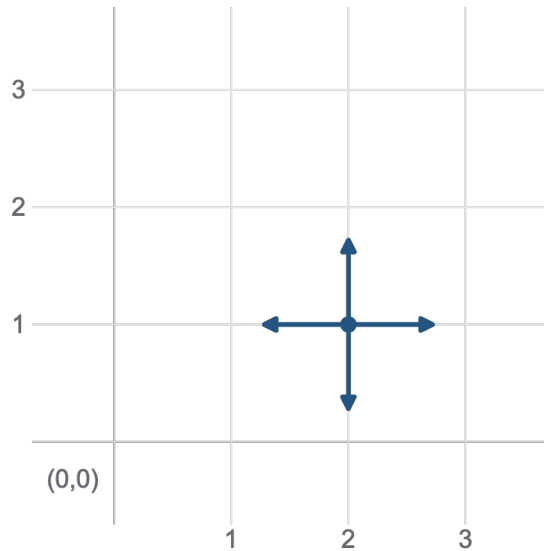
One pixel indexes a whole location coset

Recall: the **stabilizer** contains poses that fix the origin.

$$H = \{((0, 0), Q) : Q \in K\}, \quad (G, K) \in \{(p4, C_4), (SE(2), SO(2))\}$$

$$t = (2, 1)$$

$$(t, R)H = \{(t, Q) : Q \in K\}$$



Every heading at this location names the same **coset**.

$$\Phi((t, R)H) = t$$

One pixel indexes this entire coset. It does not select a heading.

These location cosets describe the input image domain.

CNN feature domains and their actions

Image locations form a **homogeneous space** even when the **stabilizer** is **non-normal**.

$$G = p4 : \quad X = \mathbb{Z}^2 \simeq G/H$$

$$G = \text{SE}(2) : \quad X = \mathbb{R}^2 \simeq G/H$$

Image field

$$f : X \rightarrow \mathbb{R}^{C_{\text{in}}}$$

Scalar channels at each image location

Spatial action

$$[T_a f](x) = f(a^{-1} \cdot x)$$

Every scalar channel moves to the transformed image location.

Questions?

Groups and actions

Next: Group convolutions

References

- [1] Bekkers. Group Equivariant Deep Learning. University of Amsterdam, Lectures 1.1–1.3.
- [2] Bronstein et al. (2021). Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges.
- [3] Cohen & Welling (2016). Group Equivariant Convolutional Networks. ICML.