MSc Artificial Intelligence
Master Thesis

A Probabilistic Approach to Learning the Degree of Equivariance in Steerable CNNs

by
Lars Veefkind
11630876

August 9, 2024

48EC
November 2022 - November 2023

Supervisor:
G. Cesa

Examiner:
Dr E.J. Bekkers

Second reader:
G. Cesa

Universiteit van Amsterdam

Abstract

Steerable convolutional neural networks (SCNNs) model geometric symmetries by imposing equivariance constraints on the learnable weights, with improved performance on tasks that exhibit these symmetries as a result. However, if the required symmetry is unknown or varies between features, it is easy to overconstrain the weights, subsequently reducing performance.

In this thesis, we introduce a probabilistic approach to learn the degree of equivariance within the flexible framework of SCNNs. We parameterise the degree of equivariance as a likelihood distribution over the transformation group, expressed in terms of its Fourier coefficients denoting the contribution of basis-functions. By introducing some specialised regularisation, we obtain an interpretable likelihood distribution denoting the degree of equivariance throughout the network. Additionally, our approach allows for a layer-wise or a shared degree of equivariance. Our method yields several advantages compared to existing solutions: Firstly, the SCNNs framework allows our solution to be applied to a plethora of equivariant networks. Second, through our parameterisation, equivariance can be learnt for any subgroup of any compact group without the need for additional layers.

We evaluate our approach on a simple 2D vector dataset, a custom double-digit MNIST dataset with controllable symmetries, and several 3D biomedical datasets. We show that our method can significantly improve performance on datasets with unknown or mixed symmetries, while maintaining high generalisation capabilities and data-efficiency. Furthermore, we show that the learnt likelihood distributions are an accurate and interpretable representation of the underlying equivariance.

Acknowledgements

First of all, I would like to thank Gabriele Cesa, my supervisor, for his invaluable guidance and support, and for sharing his extensive knowledge. It has been an incredible fun, educational and satisfying experience. I would also like to thank my family for supporting me and allowing me to ramble on about things they might not understand. Finally, I want to thank Nienke Reints for her extensive proofreading and precious feedback.

Contents

Chapter 1

Introduction

Natural images and structures often exhibit geometrical symmetries of various types, such as rotations, translations, and reflections. Figure 1.1 provides illustrative examples of these symmetries. Since LeCun et al. [20] introduced translation-equivariant Convolutional Neural Networks (CNNs), the integration of other types of symmetries into deep learning architectures has been an active area of research [6, 9, 36, 5, 7, 37, 41, 16, 2]. The aim is to instil these models with invariance or equivariance properties as a form of inductive bias.

(a) Human organs. Image credit: freepik.com.

(b) Field of sunflowers. Image credit: MikeLynch, CC BY-SA 3.

(c) Peacock. Image credit: freepik.com.

Figure 1.1: Examples of natural images exhibiting rotational and/or reflective symmetries. For each of these images, the level of symmetry varies between the features. In Figure 1.1a, the textures of low-level features, such as the blood vessels in the lungs, do not exhibit a preferred orientation, although the orientations of the organs themselves are fixed with respect to each other. Furthermore, while the arrangement of some organs demonstrates approximate vertical symmetry, this is not the case for every organ. In Figure 1.1b, the sunflower heads may rotate freely, but the stems invariably extend downward. Lastly, in the illustration shown in Figure 1.1c, the peacock's feathers radiate outward from the body in a range of directions, yet there is a noticeable absence of feathers extending straight down or nearly straight down.

Regular CNNs obtain their translation equivariance through the properties of the convolution operator that slides a learnable stationary filter over the spatial domain of an input signal. Therefore, the same operation is applied on each location, with translation equivariance as a result. Group Convolutional Neural Networks (GCNNs) [6] generalise the application of the convolution operator to not just the spatial domain, but also the domains of geometric transformation groups. As a result, GCNNs often improve general performance, data-efficiency, and generalisation capabilities [40, 42, 36, 38].

However, due to discretisation requirements, early GCNNs were only able to obtain equivariance with respect to discrete groups. Furthermore, obtaining equivariance with respect to large discrete groups resulted in a significant impact on memory requirements and computational performance. To address these shortcomings, several other works have been proposed, both within and outside the framework of GCNNs [7, 42, 38, 2, 26, 41, 40]. One of these approaches is the framework of steerable CNNs (SCNNs), introduced by Cohen and Welling [7]. The theory of SCNNs describes a general notion of equivariance, which is enforced through kernel constraints on the learnable parameters. Although early

SCNNs required a separate solution for each transformation group, Weiler and Cesa [36] proposed a generalised solution for the kernel constraint, applicable to any compact transformation group in 2D. Subsequently, Cesa et al. [5] extended this solution to nD. As a result, these SCNNs are agnostic of any specific types of features, and are therefore suitable to model many varieties of equivariant networks, including but not limited to GCNNs [6, 40, 26, 41], Harmonic Networks [42], Vector Field Networks [25], and early SCNNs [7, 37].

1.1 Problem Statement

While incorporating higher degrees of equivariance can be a powerful inductive bias, over-constraining a model can lead to diminished performance by preventing the use of potentially significant features [35, 27, 34]. Therefore, it is crucial to select an appropriate transformation group that is neither too large nor too small when developing a model. The required degree of equivariance often varies among features, typically correlating with their size. As shown in Figure 1.1, smaller features generally demonstrate more symmetries than larger ones. While this hierarchical structure of features is observed in CNNs, choosing the correct transformation group a priori is challenging and requires substantial knowledge about the dataset and the layers' receptive fields.

To address these complexities, various strategies have been proposed to autonomously learn the relaxation of equivariance constraints on a layer-by-layer basis. Among these are Residual Pathway Priors (RPPs) [8], non-stationary GCNNs [32], and Partial GCNNs [28]. Despite the adaptability of the SCNN framework, none has been specifically developed within this framework. Therefore, to the best of our knowledge, RPP is the only approach that is directly applicable for SCNNs, providing a generalised method for modulating equivariance. However, RPPs achieve this modulation through residual connections that are non-equivariant, or only partially equivariant. Since there is no separate parameterisation that describes the degree of equivariance, such as a likelihood distribution over the transformation group [28], it is impossible to obtain an interpretable degree of equivariance without performing computationally expensive analyses. We cover these methods in more detail in Chapter 4.

1.2 Contributions

Building upon the SCNN framework developed by Cesa et al. [5], we propose two general solutions to obtain a learnable degree of equivariance in SCNNs, as well as equivariant MLPs (E-MLPs). The first approach, which is comparable to RPPs, serves as a proof of concept and is not easily interpretable. Here, we allow the model to break equivariance by allowing it to use connections that are not available in a fully equivariant setting. Our second and main approach similarly gradually learns to use these initially unavailable connections, but parameterises this using an interpretable likelihood over the transformation group.

To achieve this, we use the fact that the kernel constraints can be interpreted as an implicit uniform averaging over the transformation group. Subsequently, we show that through the existing framework of SCNNs, this uniform averaging can be modelled in terms of Fourier coefficients describing the contribution of group-dependent basis-functions through a uniform likelihood distribution. By separately parameterising the Fourier coefficients and ensuring that the resulting distribution is normalised and aligned, we are able to learn an interpretable likelihood distribution over any compact group in nD, including continuous groups, through backpropagation. This likelihood distribution corresponds to the inherent degree of equivariance of the model and can easily be extracted through the inverse Fourier transform without requiring any forward passes. In addition to allowing this on a layer-by-layer basis, our implementation allows a single likelihood distribution to be shared between multiple layers, resulting in a shared degree of equivariance between these layers.

Additionally, to increase the accuracy and therefore interpretability of the learnt likelihood distributions between consecutive layers, we propose to use KL-divergence [17] between the likelihood distri-

butions of these layers, as this prevents a layer from incorrectly regaining previously lost equivariance. Finally, both to provide a means of regularisation on the flexibility of the likelihood distribution and a reduction in computational performance, our approach allows the user to tune the bandlimiting of the Fourier coefficients of the likelihood distribution as a hyper-parameter.

Finally, our method incorporates a hyper-parameter that adjusts the bandlimiting of the Fourier coefficients associated with the likelihood distribution. This serves two purposes: first, it acts as a form of regularisation, constraining the flexibility of the likelihood distribution to prevent overfitting and the interpretability of the likelihood distribution; and second, it improves computational efficiency by reducing the computational complexity. By using fewer Fourier coefficients the likelihood distribution contains fewer basis-functions, and thus has a reduced complexity.

To evaluate our approaches, we perform an elaborate series of experiments in which we compare our approach to several baselines. Here we include regular SCNNs with pre-tuned layer-wise degrees of equivariance through a group restriction [36], and RPP, as, to our knowledge, this approach is the only alternative approach that is directly applicable on SCNNs. In these experiments, we evaluate the impact on task-specific performance in biomedical classification in `MedMNIST` [44], double-digit number classification on an adapted version of `MNIST` [21], and regression on a custom dataset containing vectors. Furthermore, we investigate the interpretability and correctness of the learnt underlying likelihood distributions. We also study the influence of bandlimiting and KL-divergence, both in terms of performance and interpretability. Finally, since data-efficiency and generalisation capabilities are two main advantages of equivariant networks [40, 42, 36, 38], we compare these aspects between our approach and the baselines. Through these experiments, we offer the following contributions:

- We develop a *preliminary* approach to learning the degree of equivariance and show that it can significantly improve performance compared to fully equivariant E-MLPs on datasets that lack the symmetry of the equivariance group.
- We develop a *probabilistic* approach and show that it results in more reliable performance compared to the preliminary approach due to additional weight sharing and consistency.
- We show that, on datasets with varying symmetries, our probabilistic approach has the capacity to outperform fully equivariant SCNNs, regular CNNs, and RPPs on relatively symmetric datasets, while performing on par with RPPs on non-symmetric datasets.
- Our findings reveal that through the addition of two regularisation terms, the learnt likelihood distributions are an accurate representation of the underlying degree of equivariance in SCNNs, thus making the degree of equivariance easily interpretable.
- Our results indicate that sharing the degree of equivariance in subsequent linear layers can result in more accurate and therefore interpretable likelihood distributions.
- We show that through bandlimiting, the flexibility of the likelihood distribution can be controlled, serving as a means to regularise the breaking of equivariance.
- We demonstrate that while our probabilistic approach sacrifices a small degree of generalisability compared to SCNNs with a fixed degree of equivariance, it surpasses RPPs when it comes to generalising from discrete to continuous group symmetries.
- We show that our approach achieves a relatively high degree of data-efficiency on datasets that are fairly symmetric.

1.3 Thesis Outline

The subsequent chapters of this thesis are organised as follows. Chapter 2 provides an overview of the required mathematical preliminaries, such as group theory, representation theory, equivariance, and the Fourier transform. Subsequently, in Chapter 3 we discuss equivariant neural networks, starting at GCNNs and working our way towards SCNNs. As some of the related research depends on the

theories and methods provided in Chapters 2 and 3, Chapter 4 contains a more detailed overview of the related work. Afterwards, Chapter 5 contains the theoretical details and substantiation of our proposed approaches. The implementation details are presented in Chapter 6. Next, Chapter 7 contains an overview of our evaluation methodology, covering both our datasets and model configurations. Our results and evaluation are presented in Chapter 8. Finally, we conclude this thesis in Chapter 9.

Chapter 2

Mathematical Preliminaries

In this chapter, we cover the mathematical preliminaries required for the work in this thesis. We will start with the basics of Group Theory and Representation Theory, and work our way towards Equivariance and Invariance.

2.1 Group Theory

Definition 1: Group

A *group* is a pair $(G, \cdot)$, containing a set G and a binary operation $\cdot$, also called the *group law*:

$$\cdot : G \times G \to G, (h, g) \mapsto h \cdot g$$

satisfying the following axioms of group theory:

- Associativity: $\forall a, b, c \in G :$ $\quad a \cdot (b \cdot c) = (a \cdot b) \cdot c$
- Identity: $\forall e \in G, \forall g \in G :$ $\quad g \cdot e = e \cdot g = g.$
- Inverse: $\forall g \in G, \exists g^{-1} \in G :$ $\quad g \cdot g^{-1} = g^{-1} \cdot g = e.$

It can be proven that the inverse element g^{-1} of an element g, and the identity element e are unique.

To reduce notation it is common to write hg instead of $h \cdot g$ and to refer to the whole group with G instead of $(G, \cdot)$ as long as this is not ambiguous. It is also common to use the power notation to abbreviate the combination of the element g with itself n times:

$$g^n = \underbrace{g \cdot g \cdot g \cdots g}_{n \text{ times}}$$

Example 1: General Linear Group

The *general linear group* $(GL(\mathbb{R}^n), \cdot)$ is the group with the matrix multiplication $\cdot$ as group law and the set $GL(\mathbb{R}^n)$ of all invertible $n \times n$ matrices:

$$GL(\mathbb{R}^n) = \left\{ O \in \mathbb{R}^{n \times n} \mid O^{-1}O = \mathbf{I}_n \right\}$$

The identity element e is the identity matrix $\mathbf{I}_n$ of size $n \times n$. Since the matrix multiplication is associative, this definition satisfies the axioms from Definition 1.

Beyond the structure of a group itself, it is useful to consider the cardinality of the set G within a group, formally termed as the order of a group.

Definition 2: Order of a Group

The *order* of a group G is the *cardinality* of its set G and it is indicated by $|G|$.

Given the order, groups naturally fall into two categories: finite and infinite groups.

Definition 3: Finite Group

A *finite group* is a group with a finite number of elements, and therefore a finite cardinality.

Therefore, an infinite group is a group G with cardinality $|G| = \infty$. An example of a finite group is the set of integers module n.

Example 2: $\mathbb{Z}/n\mathbb{Z}$

The set $\{e = 0, 1, 2, \ldots, n-1\}$ with the associative group law sum modulo n:

$$+ : (a, b) \mapsto (a + b) \mod n$$

forms the group $\mathbb{Z}/n\mathbb{Z}$ of integers modulo n.

1. Identity: The element 0 serves as the identity element for this group. For any integer a in $\mathbb{Z}/n\mathbb{Z}$, $a + 0 \equiv a \mod n$. Thus, adding 0 to any element does not change its value in this set.
2. Inverse Element: Every element a in $\mathbb{Z}/n\mathbb{Z}$ has an inverse b such that $a + b \equiv 0 \mod n$. For instance, when $n = 10$, the element 4 has the inverse 6 because $(4 + 6) \mod 10 \equiv 0 \mod 10$.

The group has order $|\mathbb{Z}/n\mathbb{Z}| = n \in \mathbb{N}^+$, and is therefore a *finite group*.

Now that we have introduced the basic concepts of group theory, we move onto more advanced concepts. These involve relationships between groups, and transformations of groups.

Definition 4: Group Homomorphism

Given two groups $(G, \cdot)$ and $(H, *)$, a map $f : G \to H$, mapping an element in group G to an element in group H, is a *group homomorphism* from G to H if:

$$\forall a, b \in G: \quad f(a \cdot b) = f(a) * f(b)$$

A special case of a homomorphism is an *isomorphism*, which adds the condition of *bijectivity*.

Definition 5: Group Isomorphism

A group homomorphism f from G to H is a group *isomorphism* if it is bijective (surjective and injective), i.e. if and only if:

$$\forall h \in H, \ \exists!\ g \in G: \quad f(g) = h.$$

If we have two groups G and H and there exists an isomorphism between the two, then the two groups are considered to be isomorphic.

Definition 6: Group Automorphism

A group homomorphism f from G to G itself is called a group endomorphism. A group endomorphism which is also bijective (i.e., an isomorphism) is called a group automorphism.

Groups can be useful for describing space-based symmetries. In this work we focus on describing or

negating symmetries in $\mathbb{R}^2$ and $\mathbb{R}^3$. Mathematically, these symmetries are described as group actions acting on a specific space.

Definition 7: Group Action and G-space

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

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

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

For any group $(G, \cdot)$, its group law $\cdot : G \times G \to G$ trivially defines a group action of the group on itself.

Definition 8: 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:

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

where $\cdot$ denotes the group action from Def 7.

One of the main symmetries we are interested in are rotational symmetries. The Cyclic Group can be used to model rotations, but it is limited to a fixed number of discrete rotations.

Example 3: Cyclic Group

The collection of all complex N-th roots of unity, $\left\{e^{ik\frac{2\pi}{N}} | 0 \leq k \leq n\right\}$, forms a group under multiplication. This group is isomorphic to the group $\mathbb{Z}/n\mathbb{Z}$, as depicted in Ex. 2. A homomorphism f can be established between these two groups and can be defined as

$$f : e^{ik\frac{2\pi}{n}} \mapsto k.$$

This finite group, generally denoted as C_N, is referred to as the Cyclic Group of order $|C_N| = N$. On a more abstract level, this group can be described as

$$C_N = \left\{e = g^0, g, g^2, \ldots, g^{n-1} \mid g^m = g^n \iff m \equiv n \bmod N\right\}.$$

Here, the elements of the group are indicated using power notation.
The Cyclic Group appears frequently in this work, manifesting itself as discrete rotations when acting on the plane $\mathbb{R}^2$, or as a reflection when acting on rotation groups.

In contrast to the Cyclic Group, the Special Orthogonal Group contains continuous rotations.

Example 4: Special Orthogonal Group

The Special Orthogonal Group is the group of all continuous rotations. This infinite group is denoted as $SO(n)$, where n denotes the dimensionality of the space on which the group acts. The most notable examples are $SO(2)$ and $SO(3)$, denoting planar and volumetric rotations respectively. The action of a rotation $r_\Theta \in SO(n)$ with the rotation angles (or singular angle in the case of $SO(2)$) parameterized by the set Θ on the respective space can be defined as:

$$\forall x \in X = \mathbb{R}^n: \quad r_\Theta, x \mapsto r_\Theta \cdot x = \psi(\Theta)x$$

Where $\psi(\Theta)$ is the rotation matrix, i.e., for rotations in $SO(2)$ with $\Theta = \{\theta\}$:

$$\psi(\Theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix},$$

and $\psi(\Theta)x$ is the regular matrix-vector product.

The $SO(n)$ groups are therefore the group of all $n \times n$ orthogonal real matrices with positive determinant

$$SO(n) = \left\{ O \in \mathbb{R}^{n \times n} \mid O^\top O = \mathbf{I}_n \text{ and } \det(O) = 1 \right\}.$$

In this work, we often consider groups whose elements are completely included in another larger group. These groups are called subgroups.

Definition 9: Subgroup

Given a group $(G, \cdot)$, a non-empty subset $H \subseteq G$ is a subgroup of G if it forms a group $(H, *)$ under the same group law, restricted to H.
For H to be a subgroup of G, it is required and sufficient that the restricted group law and the inverse are closed in H:

- $\forall a, b \in H: \quad a * b \in H$
- $\forall h \in H: \quad h^{-1} \in H$

In this case, it is common to write $H \leq G$.

Definition 10: Coset

Consider the group G and subgroup $H \leq G$. A *left coset* of H in G is $gH = \{gh \mid h \in H\}$ for an element $g \in G$. Likewise, a *right coset* of H in G is $Hg = \{hg \mid h \in H\}$ for an element $g \in G$.

Therefore, a coset of an element g is the set of all $g' \in G$ which are obtainable through the respective left or right action of an element $h \in H < G$ on the element g.

Example 5: Left Coset of $SO(2)$ in $O(2)$

Let us consider the *left coset* of $SO(2)$ in $O(2)$.
Given $SO(2)$ as our subgroup, consider a reflection $g \in O(2)$ which acts as a pure reflection along the x-axis. The matrix representation of this reflection is:

$$g = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

The left coset generated by this reflection with respect to $SO(2)$ will be:

$$gSO(2) = \{g \cdot r_\Theta \mid r_\Theta \in SO(2)\}$$

Given our representation of rotations in $SO(2)$ from Ex. 4, for any rotation $\psi(\Theta) \in SO(2)$, the result of the matrix multiplication $g \cdot \psi(\Theta)$ will be a reflection followed by a rotation by θ. Since $\det(g) = -1$ and $\det(\psi(\Theta)) = 1$, the resulting coset is as follows:

$$gH = gSO(2) = \left\{O \in \mathbb{R}^{2\times 2} \mid O^\top O = \mathbf{I}_n \text{ and } \det(O) = -1\right\}.$$

Definition 11: Normal Subgroup

Given a group G with a subgroup $H \leq G$, H is a normal subgroup of G iff:

$$\forall g \in G\colon \quad gH = Hg$$

If this holds, the fact that H is a normal subgroup of G can be denoted as $H \trianglelefteq G$.

To illustrate the concept of a subgroup, let us consider the cyclic group C_N and its relation to the group of all continuous rotations in a plane, $SO(2)$.

Example 6: C_N as a Subgroup of $SO(2)$

In Examples 4 and 3 we defined the group of all continuous rotations in a plane, $SO(2)$, and the cyclic group C_N respectively. The cyclic group C_N, being a finite group of rotations, is actually a subgroup of the group $SO(2)$ of infinite rotations. It represents the group of N rotations by integer multiples of an angle equal to $\frac{2\pi}{N}$.
From Ex. 3, we know that C_N is isomorphic to the group of N-th roots of unity under multiplication, which can be viewed as rotations in the complex plane, denoted by $\mathbb{C}$. We can relate the elements of C_N to $SO(2)$ through the following mapping function:

$$C_N \mapsto SO(2), \quad g^k \mapsto r_{k\frac{2\pi}{N}}.$$

For example, if we take the cyclic group C_4, together with the element $g^3 \in C_4$, then this element corresponds to $r_{3\frac{2\pi}{4}} = r_{\frac{3\pi}{2}}$, or a rotation of 270 degrees.

Occasionally we want to describe certain groups as combinations of two or more smaller groups. There are two types of group products allowing for such decompositions; the *direct product* and *semi-direct product*. First, lets consider the *direct product*.

Definition 12: Direct Product

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

$$(k_1, h_1) \cdot (k_2, h_2) = (k_1 * k_2, h_1 + h_2).$$

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

Given a direct product set $K \times H$, we find that the subsets $\{(e_K, h) | h \in H\}$ and $\{(k, e_H) | k \in K\}$ form distinct normal subgroups. Furthermore, these subsets are isomorphic to H and K, respectively. This structure enables us to uniquely decompose any element (k, h) of the set $K \times H$ as a product of an element from K and an element from H. For instance, we can represent (k, h) as either $(e_K, h) \cdot (k, e_H)$ or $(k, e_H) \cdot (e_K, h)$. A notable characteristic of these sets is that all elements of K commute with the elements of H. We can observe the structure of a direct-product in the decomposition of groups of translations.

Example 7: Decomposition of Translations

The group of translations on the plane, denoted as $(\mathbb{R}^2, +)$, can be decomposed into its constituent vertical and horizontal translations. As a result, we can say that the group $(\mathbb{R}^2, +)$ is isomorphic to the direct product $(\mathbb{R}, +) \times (\mathbb{R}, +)$, which represents two instances of the group $(\mathbb{R}, +)$, signifying translations along a line. Following the same line of thought, we can use the direct product between more than two groups. For example, for translations in three-dimensional space, represented by the group $(\mathbb{R}^3, +)$. This group is isomorphic to the direct product $(\mathbb{R}, +) \times (\mathbb{R}, +) \times (\mathbb{R}, +)$, which can be viewed as three instances of the group $(\mathbb{R}, +)$.

While the direct product decomposes a group in the product of two normal subgroups whose elements commute with each other, the semi-direct product only requires one of the subgroups to be normal subgroups.

Definition 13: Semi-Direct Product

Considering two groups $(N, *)$ and $(H, +)$ and a group action $\phi : H \times N \to N$ 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:

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

It is important to note here that the resulting group depends on the map ϕ and that changing the map leads to a change in the group.
Like in a direct product, any element of a semi-direct product can be uniquely identified by a pair of elements on the two subgroups.

Here, only the group N is required to be a normal subgroup of G. Additionally, when ϕ is the identity map on N for any $h \in H$, we obtain the more specific definition for the previously defined direct product. The semi-direct product plays a role in the decomposition of the special euclidean- and orthogonal groups.

Example 8: Orthogonal Group $O(n)$

In Ex. 4 we introduced the Special Euclidean Group $SO(n)$, the group of continuous rotations in n-dimensional space. Another common group in this work is the Orthogonal Group $O(n)$. In addition to rotations, this group also contains reflections in the n-dimensional space. Like in $SO(n)$, the action of a rotation r_Θ in $O(n)$ is defined as:

$$\forall x \in X = \mathbb{R}^n: \quad r_\Theta, x \mapsto r_\Theta \cdot x = \psi(\Theta)x$$

Where $\psi(\Theta)$ is the rotation matrix.
A potential reflection may or may not reflect the point x around the first axis. This operation can be represented by inverting the first coordinate of a point, depending on the value of a parameter p that dictates whether or not a reflection occurs. For $p = -1$, a reflection occurs, while for $p = 1$, the original point is preserved. Thus we have that for a given a potential reflection operator f_p where $p \in \{-1, 1\}$, the mapping is defined as follows:

$$\forall x \in X = \mathbb{R}^n: \quad f_p, x \mapsto f_p \cdot x = \begin{bmatrix} p & 0 \\ 0 & \mathbf{I}_{n-1} \end{bmatrix} x$$

Here, $p = -1$ results in a reflection, whereas $p = 1$ leaves the point unchanged and $\mathbf{I}_{n-1}$ is the $(n-1) \times (n-1)$ identity matrix. Therefore, this group is the group of all $n \times n$ orthogonal real matrices:

$$O(n) = \left\{ O \in \mathbb{R}^{n \times n} \mid O^\top O = \mathbf{I}_n \right\}$$

With $\forall O \in O(n) \quad \det(O) = \pm 1$.

Let us consider the planar case $n = 2$, hence $G = O(2)$. $O(2)$ can be decomposed as a semi-direct product of $SO(2)$ and the cyclic group C_2. Any element of $O(2)$ can be identified by a pair (r_Θ, f_p), where $r_\Theta \in SO(2)$ and $f_p \in C_2$. The resulting product between these two elements is:

$$\begin{aligned}
(r_{\Theta_1}, f_{p_1}) \cdot (r_{\Theta_2}, f_{p_2}) &= r_{\Theta_1} f_{p_1} r_{\Theta_2} f_{p_2} \\
&= r_{\Theta_1} r_{\phi(f_{p_1})\Theta_2} f_{p_1} f_{p_2} \\
&= (r_{\Theta_1} r_{\phi(f_{p_1})\Theta_2}, f_{p_1} f_{p_2}) \\
&= (r_{\Theta_1} r_{\phi(f_{p_1})\Theta_2}, f_{p_1} f_{p_2})
\end{aligned}$$

The action ϕ can be identified as:

$$\phi : C_2 \times SO(2) \to (r_{\Theta_1}, f_{p_2}), SO(2) \mapsto r_{\phi(f_{p_2})\Theta_1}$$

Hence, $O(n)$ can be decomposed as a semi-direct product:

$$O(2) = C_2 \rtimes_\phi SO(2)$$

Here, the action of the C_2 group can be interpreted as a rotation of 0 or π around the x-axis in the coordinate system, where the 2-dimensional space is embedded in a 3-dimensional space.

Example 9: Special Euclidean Group $SE(n)$

The Special Euclidean Group, denoted as $SE(n)$, is the group of all translations and rotations in the n dimensional space. Groups $SO(n)$ (Ex. 4) and $(\mathbb{R}^n, +)$ can be chosen as the respective subgroup and normal subgroups of $SE(n)$. Any element of $SE(n)$ can be identified by a pair (t_v, r_Θ), where $t_v \in \mathbb{R}^n$ and $r_\Theta \in SO(n)$. The resulting product between these two elements is:

$$\begin{aligned}(t_{v_1}, r_{\Theta_1}) \cdot (t_{v_2}, r_{\Theta_2}) &= t_{v_1} r_{\Theta_1} t_{v_2} r_{\Theta_2} \\ &= t_{v_1} t_{\psi(\Theta_1)v_2} r_{\Theta_1} r_{\Theta_2} \\ &= (t_{v_1} t_{\psi(\Theta_1)v_2}, r_{\Theta_1} r_{\Theta_2}) \\ &= (t_{v_1 + \psi(\Theta_1)v_2}, r_{\Theta_1} r_{\Theta_2})\end{aligned}$$

The action ϕ can be identified as:

$$\phi : (\mathbb{R}^n, +) \times SO(n) \to (\mathbb{R}^n, +), (t_{v_2}, r_{\Theta_1}) \mapsto t_{\psi(\Theta_1)v_2}$$

Hence the $SE(n)$ can be decomposed as a semi-direct product:

$$SE(n) = (\mathbb{R}^n, +) \rtimes_\phi SO(n)$$

2.2 Group Representation Theory

In this section, we will focus on *linear group representations*. These group actions model group elements G as matrices, making them suitable for acting on G-spaces that are vector spaces. This is particularly useful for deep learning, since both the data and intermediate features are generally represented as vectors, and therefore form a vector G-space.

Definition 14: Linear Group Representation

A Linear Group Representation ρ of a group G on a vector space (representation space) V is a group homomorphism from G to the general linear group $GL(V)$, i.e., it is a map

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

In the rest of this work, the shorthand *representation* or *group representation* will be used for *linear group representation*. Perhaps the simplest example of a *representation* is the *trivial representation*.

Example 10: Trivial Representation

The *trivial representation* $\rho : G \to GL(\mathbb{R})$ maps any group element to the identity. Therefore,

$$\forall g \in G: \quad \rho(g) = 1.$$

The groups used in this work often contain rotations, which are modelled as rotation matrices.

Example 11: Rotation Matrices

As shown in Example 4, an example of a representation of the group $SO(2)$ are the two-dimensional rotation matrices:

$$\psi : SO(2) \to GL(\mathbb{R}^2), r_\theta \mapsto \psi(r_\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

Representations that act similarly to one another are called *equivalent representations*.

Definition 15: Equivalent Representations

Two representations ρ and ρ' on a vector space V are called equivalent, or isomorphic if and only if they are related by a change of basis $Q \in GL(V)$, that is:

$$\forall g \in G: \quad \rho'(g) = Q\rho(g)Q^{-1}$$

It is straightforward that *equivalent representations* behave similarly, since their composition is independent of the basis:

$$\rho'(g_1)\rho'(g_2) = Q\rho(g_1)Q^{-1}Q\rho(g_2)Q^{-1} = Q\rho(g_1)\rho(g_2)Q^{-1}$$

A collection of representations can be combined into a single representation through the *direct sum*.

Definition 16: Direct Sum

Given two representations $\rho_1 : G \to GL(V_1)$ and $\rho_2 : G \to GL(V_2)$, their direct sum $\rho_1 \oplus \rho_2 : G \to GL(V_1 \oplus V_2)$ is defined as:

$$(\rho_1 \oplus \rho_2)(g) = \begin{bmatrix} \rho_1(g) & 0 \\ 0 & \rho_2(g) \end{bmatrix}.$$

Here, the corresponding matrices of the two representations are combined by simply taking the direct sum between them. Therefore, the action of the direct sum is given by the independent actions of ρ_1 and ρ_2 on the orthogonal subspaces V_1 and V_2, respectively, in $V_1 \oplus V_2$.

When consecutively taking the *direct sum* between multiple representations, we use $\bigoplus$, e.g.:

$$\bigoplus_i \rho_i(g) = \rho_1(g) \oplus \rho_2(g) \oplus \rho_3(g) \oplus \dots.$$

Suppose $\rho : G \to GL(V)$ is a group representation and V_1 is a subspace of V such that $\forall g \in G : \rho(g)(V_1) \subseteq V_1$. Then V_1 is an invariant subspace under the action of ρ. We can find an orthogonal complement V_2 of V_1 such that $V = V_1 \oplus V_2$. If we choose a basis of V consistent with this decomposition, then the matrix of $\rho(g)$ with respect to this basis has a block-diagonal form:

$$\rho(g) = \begin{bmatrix} \rho_1(g) & 0 \\ 0 & \rho_2(g) \end{bmatrix},$$

where $\rho_1 : G \to GL(V_1)$ and $\rho_2 : G \to GL(V_2)$ are the restrictions of ρ to V_1 and V_2 respectively. In this case, we say that ρ is the direct sum of the representations ρ_1 and ρ_2. Representations that do not leave a subspace invariant are called *irreducible representations*, which turn out to be useful building blocks for equivariant networks.

Definition 17: Irreducible Representation

A representation is irreducible if it does not contain any non-trivial invariant subspaces. *Irreps* is often used as shorthand for irreducible representations. We often denote irreps as ψ rather than ρ.

Given a group G, the set of irreducible representations of G is often denoted as $\hat{G}$. The *trivial representation* (Ex. 10) is always in the set of irreducible representations $\hat{G}$.

In this work, in addition to translations, we will mostly focus on the rotation and reflection groups, both the discrete and continuous settings. These groups are all subsets of $O(n)$. Let us provide an overview of these subgroups in the 2D case $H \leq O(2)$:

Example 12: Irreducible Representations of Subgroups $H \leq O(2)$

Here we denote the irreps of the Cyclic group C_N, Dihedral group D_N, Special Orthogonal group $SO(2)$ and Orthogonal group $O(n)$ in the real case. For any of these groups, we will denote the irreps as $\psi_i^H(p)$, where H is the considered subgroup, i denotes some (group-specific and potentially multi-) index, and p is the parametrisation of the irrep.

Special Orthogonal Group $SO(2)$ The Special Orthogonal group (Ex. 4) is the group of all continuous planar rotations. $SO(2)$ has one 1-dimensional representation (the *trivial representation*), and infinitely many 2×2 rotation matrices as irreps.

- trivial representation: $\psi_0^{SO(2)}(r_\theta) = 1$
- non-trivial irreps: $\psi_k^{SO(2)}(r_\theta) = \begin{bmatrix} \cos(k\theta) & -\sin(k\theta) \\ \sin(k\theta) & \cos(k\theta) \end{bmatrix} = \psi(k\theta), \quad k \in \mathbb{N}^+$

where $\psi(k\theta)$ is the rotation matrix from Ex. 4.

Orthogonal Group $O(2)$ The Orthogonal group (Ex. 8) is the group of planar reflections and continuous rotations with index $i = (r, k)$, where $r \in \{0, 1\}$ denotes a possible reflection. In addition to the trivial representation, it has another 1-dimensional irreducible representation that performs the reflection.

- trivial representation: $\psi_{0,0}^{O(2)}(r_\theta p) = 1$
- reflection irrep: $\psi_{1,0}^{O(2)}(r_\theta p) = f, \quad p \in \{-1, 1\}$
- other irreps: $\psi_{1,k}^{O(2)}(r_\theta p) = \begin{bmatrix} \cos(k\theta) & -\sin(k\theta) \\ \sin(k\theta) & \cos(k\theta) \end{bmatrix} \begin{bmatrix} f & 0 \\ 0 & 1 \end{bmatrix} = \psi(k\theta) f_p, \ k \in \mathbb{N}^+, p \in \{-1, 1\}$

Note that irreps $\psi_{0,k}^{O(2)}, k > 0, k \in \mathbb{N}^+$ do not exist.

Cyclic Group C_N The Cyclic group (Ex. 3) is the group of N discrete planar rotations. Its irreps are similar to the irreps of the special orthogonal group, only limited to $k \leq \lfloor \frac{N}{2} \rfloor$, where $\lfloor \cdot \rfloor$ is the flooring operation.

- trivial representation: $\psi_0^{C_N}(r_\theta p) = 1$
- non-trivial irreps: $\psi_k^{SO(2)}(r_\theta) = \begin{bmatrix} \cos(k\theta) & -\sin(k\theta) \\ \sin(k\theta) & \cos(k\theta) \end{bmatrix} = \psi(k\theta), \quad k \in \left\{1, \dots, \lfloor \frac{N-1}{2} \rfloor\right\}$

In the case that N is even, there is an additional 1-dimensional irrep:

- N-even irrep: $\psi_{N/2}^{SO(2)}(r_\theta) = \cos(\frac{N}{2}\theta) = \pm 1$

Dihedral Group D_N The Dihehedral group is the group of planar reflections and N discrete planar rotations. Like the $O(2)$ group, it always has two 1-dimensional representations, with an additional two 1-dimensional irreps if N is even:

- trivial representation: $\psi_{0,0}^{D_N}(r_\theta p) = 1$
- reflection irrep: $\psi_{1,0}^{D_N}(r_\theta p) = p, \quad p \in \{-1, 1\}$
- other irreps: $\psi_{1,k}^{D_N}(r_\theta p) = \begin{bmatrix} \cos(k\theta) & -\sin(k\theta) \\ \sin(k\theta) & \cos(k\theta) \end{bmatrix} \begin{bmatrix} f & 0 \\ 0 & 1 \end{bmatrix} = \psi(k\theta) f_p,$
  $k \in \left\{1, \dots, \lfloor \frac{N-1}{2} \rfloor\right\}, p \in \{-1, 1\}$

And the N-even irreps:

- N-even irrep: $\psi_{0,N/2}^{SO(2)}(r_\theta p) = \cos(\frac{N}{2}\theta) = \pm 1$
- N-even reflection irrep: $\psi_{1,N/2}^{SO(2)}(r_\theta p) = f_p \psi_{0,N/2}^{SO(2)}(r_\theta p) = p \cos(\frac{N}{2}\theta) = \pm 1$

Upon closer inspection, it is notable that the irreps in the example above correlate with some form of frequency. For each of these groups, the irrep with index k evaluated at a rotation of θ results, up to a possible reflection, in a rotation matrix corresponding to a rotation of $k\theta$. Therefore, we often refer to these irreps as *irreps of frequency* k. A similar intuition applies for the irreps of subgroups $H = O(3)$

and $H = SO(3)$, since the irreps for these groups are similar to the irreps in this example, except that $O(3)$ also has non-reflective irreps $\psi_{0,k}^{O(3)}, k \in \mathbb{N}^+$. These irreps are therefore indiced by two separate independent frequencies $r \in \{0, 1\}$ and $k \in \mathbb{N}^+$ respectively. The concept of viewing these irreps as frequencies is relevant for future sections.

It turns out that any reducible representation can be decomposed into a direct sum of *irreps*.

Theorem 1: Irreps Decomposition (Peter-Weyl theorem part 1)

Any unitary (or orthogonal) representation $\rho : G \to V$ of a compact group[a] G over a field with characteristic zero (e.g. the complex $\mathbb{C}^n$ and real $\mathbb{R}^n$ fields) is a direct sum of irreducible representations. Each irrep corresponds to an invariant subspace of the vector space V with respect to the action of ρ. In particular, any real linear representation $\rho : G \to \mathbb{R}^n$ of a compact group G can be decomposed as

$$\rho(g) = Q \left[\bigoplus_{i \in I} \psi_i(g) \right] Q^{-1},$$

where I is an index set that specifies the irreducible representations ψ_i contained in ρ (with possible repetitions) and Q is a change of basis. Alternatively, we can write this equation more explicitly using the multiplicity s for each irrep $\psi_j \in \widehat{G}$:

$$\rho(g) = Q \left[\bigoplus_{j \in \widehat{G}} \bigoplus_{s}^{[j]} \psi_j(g) \right] Q^{-1},$$

where $[j]$ denotes the multiplicity of j in the decomposition.

[a] The exact definition of the compact group is outside the scope of this thesis. For this work, it suffices to state that a complex group has a 'compact 'topology, where the domain of the group is not infinitely large. The group $(\mathbb{R}^n, +)$ for example is *not* compact, since a translation can be infinitely large. $SO(2)$ is compact, since the rotation θ is at most 2π.

For proofs, it is often only required to consider the irreps. This will prove to be useful to solve the kernel constraint for steerable CNNs in Section 3.4. A more specific method of irreps decomposition will be introduced in Section 2.2.1.

A type of representation that is particularly useful for finite groups is the *regular representation*.

Definition 18: Regular Representation

Let G be a finite group, and let $\mathbb{R}^{|G|}$ be a vector space. Each basis vector $\mathbf{e}_g$ of $\mathbb{R}^{|G|}$ is associated with an element $g \in G$.
The regular representation of G, denoted as $\rho_{\text{reg}}^G : G \to GL(\mathbb{R}^{|G|})$, is defined as the map that takes an element $\widetilde{g} \in G$ and associates it with a permutation matrix $\rho_{\text{reg}}^G(\widetilde{g})$ in $GL(\mathbb{R}^{|G|})$.
The action of $\rho_{\text{reg}}^G(\widetilde{g})$ on a basis vector $\mathbf{e}_g$ is defined as:

$$\rho_{\text{reg}}^G(\widetilde{g}) \cdot \mathbf{e}_g = \mathbf{e}_{\widetilde{g}g}.$$

This means that $\rho_{\text{reg}}^G(\widetilde{g})$ permutes the basis vector $\mathbf{e}_g$ to the basis vector $\mathbf{e}_{\widetilde{g}g}$.

Example 13: Regular Representation of D_2

For the Dihedral group D_2, we represent each element of the group by a 4×4 permutation matrix. The complete regular representation of D_2 is as follows:

g	e	r_π	s_x	s_y
$\rho_{\text{reg}}^{D_2}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

In this representation, the i-th basis vector of the four-dimensional vector space $\mathbb{R}^4$ is associated with the i-th element in the sequence e, r_π, s, sr_π of D_2. Specifically, the enumeration is defined as $e \to 1$, $r_\pi \to 2$, $s \to 3$, $sr_\pi \to 4$. Thus, the i-th axis of $\mathbb{R}^4$ corresponds to the transformation associated with the i-th element of D_2.

The *regular representations* directly map group elements to a matrix that performs the action of the group on a vector space. However, deep learning models are comprised of mapping functions. Therefore, it can be useful to apply a group directly to a function by transforming the domain. The *left-regular representation* and *right-regular representation* define such mappings.

Definition 19: Left and Right-Regular Representations

Given a group G and a function $f : G \to Y$, where Y is any field, the *left-regular representation* $\mathscr{L}$ acts on f by "shifting" its domain from the left. For a group element g, the action $\mathscr{L}_g$ on f results in a new function $\mathscr{L}_g f : G \to Y$ defined by:

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

Similarly, the *right-regular representation* $\mathscr{R}$ acts on f by shifting its domain from the right, thus:

$$(\mathscr{R}_g f)(h) = f(hg) \quad \forall h, g \in G,$$

In the context of deep learning, it can be useful to vary the considered group throughout the network. To achieve this, the *restricted representation* can be used.

Definition 20: Restricted Representation

Any representation $\rho : G \to GL(\mathbb{R}^n)$ can be uniquely restricted to a representation of a subgroup H of G by restricting its domain to H:

$$\text{Res}_H^G(\rho) : H \to GL(\mathbb{R}^n), \quad h \mapsto \rho|_H(h)$$

If the representation ρ of G is an irrep, the resulting *restricted representation* $\rho|_H$ is not necessarily an irrep, but can be decomposed into a direct-sum of H irreps (Thm. 1).

2.2.1 Tensor Product Representation and its Decomposition

The tensor product and its decomposition in irreps will both show to be important for regular steerable networks, as well as for the extension towards learnable equivariance.

Definition 21: Tensor Product of Representations

Given two representations $\rho_1 : G \to GL(V_1)$ and $\rho_2 : G \to GL(V_2)$, the *tensor product of representations*, denoted as $\rho_1 \otimes \rho_2 : G \to GL(V_1 \otimes V_2)$, is defined as:

$$(\rho_1 \otimes \rho_2)(g) = \rho_1(g) \otimes \rho_2(g)$$

for every $g \in G$. Here, the tensor product of the two representations is obtained by taking the Kronecker product $\otimes$ of their corresponding matrices:

Definition 22: Kronecker Product

Given a matrix $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times q}$, the resulting Kronecker product block matrix $(A \otimes B) \in \mathbb{R}^{pm \times qn}$ is constructed as follows:

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{bmatrix}$$

or, more precisely:

$$A \otimes B = \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & \dots & a_{11}b_{1q} & \dots & \dots & a_{1n}b_{11} & a_{1n}b_{12} & \dots & a_{1n}b_{1q} \\ a_{11}b_{21} & a_{11}b_{22} & \dots & a_{11}b_{2q} & \dots & \dots & a_{1n}b_{21} & a_{1n}b_{22} & \dots & a_{1n}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & a_{11}b_{p2} & \dots & a_{11}b_{pq} & \dots & \dots & a_{1n}b_{p1} & a_{1n}b_{p2} & \dots & a_{1n}b_{pq} \\ \vdots & \vdots & & \vdots & \ddots & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & \ddots & \vdots & \vdots & & \vdots \\ a_{m1}b_{11} & a_{m1}b_{12} & \dots & a_{m1}b_{1q} & \dots & \dots & a_{mn}b_{11} & a_{mn}b_{12} & \dots & a_{mn}b_{1q} \\ a_{m1}b_{21} & a_{m1}b_{22} & \dots & a_{m1}b_{2q} & \dots & \dots & a_{mn}b_{21} & a_{mn}b_{22} & \dots & a_{mn}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{p1} & a_{m1}b_{p2} & \dots & a_{m1}b_{pq} & \dots & \dots & a_{mn}b_{p1} & a_{mn}b_{p2} & \dots & a_{mn}b_{pq} \end{bmatrix}$$

This product is a special case of the regular tensor product.

Thus, the tensor product representation maps the space of matrices to the space of the Kronecker product $V_1 \otimes V_2$.

A tensor product between two irreps does not necessarily result in an irrep. However, through the *Clebsch-Gordan decomposition* they can be decomposed into a direct-sum of irreps.

Theorem 2: Clebsch-Gordan Decomposition

Given two irreducible representations $\rho_l : G \to GL(V_1)$ and $\rho_J : G \to GL(V_2)$ of a compact group G, their tensor product representation $\rho_l \otimes \rho_J$ is not necessarily irreducible. However, as stated in Theorem 1, any representation can be decomposed into a *direct sum* of irreducible representations. The *Clebsch-Gordan decomposition* is the irreps decomposition for such tensor products of irreducible representations:

$$(\rho_l \otimes \rho_J)(g) = \left[C^{lJ}\right]^\top \left(\bigoplus_j \bigoplus_s^{[j(Jl)]} \psi_j(g) \right) C^{lJ}$$

Here C^{lJ} is the change of basis, $\psi_j : G \to GL(V_j)$ are the irreps in the decomposition, and $[j(Jl)]$ denotes the *multiplicity* of the irrep ψ_j.

Visually, this decomposition looks as follows in block-matrix form:

$$(\rho_l \otimes \rho_J)(g) = \underbrace{\begin{bmatrix} [C^{lJ}_{j_1}]^\top \\ [C^{lJ}_{j_2}]^\top \\ \vdots \\ [C^{lJ}_{j_n}]^\top \end{bmatrix}}_{[C^{lJ}]^\top} \underbrace{\begin{bmatrix} \psi_{j_1}(g) & 0 & \cdots & 0 \\ 0 & \psi_{j_2}(g) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \psi_{j_n}(g) \end{bmatrix}}_{\left(\bigoplus_j \bigoplus_s^{[j(Jl)]} \psi_j(g)\right)} \underbrace{\begin{bmatrix} C^{lJ}_{j_1} & C^{lJ}_{j_2} & \cdots & C^{lJ}_{j_n} \end{bmatrix}}_{C^{lJ}}$$

Here, $C^{lJ}_{j_1}$ through $C^{lJ}_{j_n}$ are blocks of columns in C^{lJ}. Note that, due to the block-diagonal nature, these blocks act only on one specific irrep (as visualised with the colours). Therefore, we can re-write this equation replacing the direct-sums with regular summations:

$$(\rho_l \otimes \rho_J)(g) = \sum_j \sum_s^{[j(Jl)]} \mathrm{CG}_s^{j(Jl)} \, \psi_j(g)$$

Where $\mathrm{CG}_s^{j(Jl)} \in \mathbb{R}^{d_j \times d_l d_J}$ are the resulting *Clebsch-Gordan coefficients*.

In the context of the group $SO(2)$, the *Clebsch-Gordan decomposition* of the tensor product between two irreps with frequencies k_1 and k_2 results in a decomposition of up to two irreps of the frequencies $|k_1 - k_2|$ and $k_1 + k_2$ respectively. Notably, the trivial representation (where all frequencies are zero) appears only in the tensor product between equal irreps, $k_1 = k_2$, since only in that case $|k_1 - k_2| = 0$. For the group $SO(3)$, the decomposition includes not only the lowest and highest frequencies but also all frequencies in between $|k_1 - k_2|$ and $k_1 + k_2$. It is important to note that for other groups, the decomposition pattern might differ. However, a similar pattern can be observed for groups such as $O(n)$, C_n, and D_n.

2.3 Equivariance and Schur's Lemma

The aim of Steerable Networks is to become equivariant, or perhaps invariant, to a specific set of transformations. Therefore, in this section, we mathematically define the concepts of *equivariance* and the more special case of *invariance* in terms of representation- and group theory. Furthermore, we use these definitions to introduce the concept of intertwiners, and *Schur's Representation Lemma*.

Definition 23: Equivariance

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

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

In other words, when an operation is equivariant with respect to group G, or *G-equivariant*, when the input transforms the out transforms predictably. Equivariance is therefore a generalisation of the more specific *invariance*.

Definition 24: Invariance

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

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

Therefore, invariance is a special case of equivariance, where the action of G on the set Y is the trivial action:

$$\forall y \in Y, \forall g \in G: \quad g \cdot y = y.$$

Functions that perform an equivariant mapping between representations are called *intertwiners*. These are particularly relevant for equivariant deep learning, as the mapping between features must be equivariant.

Definition 25: Intertwiner

Consider the group G and the two representations $\rho : G \to GL(V_1)$ and $\rho_2 : G \to GL(V_2)$. A linear map W from V_1 to V_2 is an intertwiner between ρ_1 and ρ_2 if it is an equivariant map:

$$\forall \boldsymbol{v} \in V_1, \forall g \in G: \quad W\rho_1(g)\boldsymbol{v} = \rho_2(g)W\boldsymbol{v}$$

from which follows that:

$$\forall g \in G: \quad W\rho_1(g) = \rho_2(g)W.$$

In fact, it turns out that two irreducible representations ρ_1 and ρ_2 must be equivalent representations if W is not a null map.

Theorem 3: Schur's Representation Lemma

Consider group G along with two of its irreps $\psi_1 : G \to V_1$ and $\psi_2 : G \to V_2$, and a linear map $W : V_1 \to V_2$ satisfying $\forall g \in G: \quad \psi_2(g)W = W\psi_1(g)$. Then either one of the following must hold:

- W is the null map
- W is an isomorphism, and therefore ψ_1 and ψ_2 are equivalent representations (Def. 15) where W is the change of basis (Q from Def. 15) between ψ_1 and ψ_2.

2.4 Peter-Weyl Theorem: Fourier Transform and Steerable Basis

In this section we discuss the *Peter-Weyl theorem*, along with two relevant applications. While we have already seen in Ex. 12 that the irreps of certain groups represent frequencies, through the *Peter-Weyl theorem* we build upon this notion by relating irreps to the Fourier transform; a popular technique in signal processing. This allows us to describe a continuous signal in terms of its frequencies. Additionally, for steerable CNNs we require a way to parameterise the kernels. Using the *Peter-Weyl theorem*, a convenient parameterisation of kernels can be built.

Theorem 4: Peter-Weyl Theorem

Let G be a compact group and $\rho : G \to GL(V)$ a unitary representation. Then ρ decomposes into a direct sum of finite-dimensional and unitary irreducible representations of G (Thm. 1). Moreover, let $L^2(G)$ be the vector space of square integrable functions over G. Consider the regular representation of G on $L^2(G)$ given by its left-action (Def. 19):

$$f \mapsto \rho(g) : [\mathscr{L}_g f](h) = f(g^{-1}h)$$

This representation is unitary, and therefore decomposes according to Thm. 1. In particular, the matrix coefficients of the irreps of G span the vector space $L^2(G)$. If complex valued irreps are considered, then their matrix coefficients form a basis for this space.
Finally, an orthonormal basis for *complex* square integrable functions in $L^2(G)$ can be explicitly built as $\left\{\sqrt{d_\psi}[\psi(g)]_{ij} \mid \psi \in \widehat{G}, 1 \leq i, j \leq d_\psi\right\}$.

It is important to note that this theorem only holds as such for the complex case. In the real case, there can be some redundancy in the irreps. Consider, for example, the 2-dimensional irreps for $SO(2)$ (Ex. 12). Here, the second column of an irrep $\psi_k^{SO(2)}$ can be obtained by multiplying the first column with the anti-symmetric matrix $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$:

$$\begin{bmatrix} -\sin(k\theta) \\ \cos(k\theta) \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cos(k\theta) \\ \sin(k\theta) \end{bmatrix}.$$

The relation between the irreps and their non-redundant columns is modelled by the *endomorphism basis*.

Definition 26: Endomorphism Basis

Let G be a compact group and $\psi_i : G \to GL(V)$ be a real irrep of G. We define $\overline{\psi}_i : G \to \mathbb{R}^{d_i \times n_i}$ to contain the non-redundant columns of ψ_i. Here, $n_i = \frac{d_i}{m_i}$, where $m_i = [0(ii)]$ is the multiplicity of the trivial representation in the Clebsch-Gordan decomposition of the tensor product $\psi_i \otimes \psi_i$ (Thm. 2).

The *endomorphism space* End_{ψ_i} consists of all linear endomorphisms $c : V \to V$ such that $c \circ \psi_i(g) = \psi_i(g) \circ c$ for all $g \in G$. A particular basis $\mathcal{C}_{\psi_i} = \{c_r^i \mid c_r^i \in \mathbb{R}^{d_i \times d_i}, r \leq m_i\}\ \forall \psi_i \in \widehat{G}$ spanning this endormorphism space can be constructed such that the elements c_r^i describe how the non-redundant columns $\widetilde{\psi}_i$ relate to the entire irrep ψ_i. See [5, Appendix C] for more details.

To transform the non-redundant columns $\overline{\psi}_i$ into the full irrep ψ_i, we introduce the function $\mathcal{R}_{\psi_i} : \mathbb{R}^{d_i \times n_i} \to \mathbb{R}^{d_i \times d_i}$. This function accepts $\overline{\psi}_i(g)$ as its input. Leveraging the endomorphism basis $\mathcal{C}_{\psi_i}$, it produces the entire irrep $\psi_i(g)$ by sequentially left-multiplying the elements c_r^i with the non-redundant columns $\overline{\psi}_i(g)$:

$$\psi_i(g) = \left[\ c_0^i \overline{\psi}_i(g) \mid c_1^i \overline{\psi}_i(g) \mid \ldots \mid c_{m_i}^i \overline{\psi}_i(g)\ \right]$$

We can also construct an inverse $\mathcal{R}_{\psi_i}^{-1}$, which performs the opposite operation.

For example, in the case of irreps $\psi_0^{SO(2)}$ and $\psi_k^{SO(2)}$, the endomorphism bases are defined as fol-

lows:

$$\mathcal{C}_{\psi_0^{SO(2)}} = \left\{ \begin{bmatrix} 1 \end{bmatrix} \right\}$$
$$\mathcal{C}_{\psi_k^{SO(2)}} = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \right\}.$$

On the contrary, the endomorphism basis for any irrep $\psi^{O(2)}$ of $O(2)$ only contains the identity matrix, since these irreps do not contain redundant columns.

$$\mathcal{C}_{\psi^{O(2)}} = \left\{ I_{d_{\psi^{O(2)}} \times d_{\psi^{O(2)}}} \right\}.$$

Cesa et al. [5, Appendix C.3] have shown that in such cases we only need to consider the first n non-redundant columns of the irreps ψ to build an orthonormal basis. Therefore, using real irreps, an orthonormal basis can be explicitly built as follows:

$$\left\{ \sqrt{d_\psi} \psi_i(g)_j \mid \psi_i \in \widehat{G}, j \leq n_i \right\},$$

replacing d_ψ with n_i, for example $n_{\psi_k^{SO(2)}} = \frac{2}{2} = 1$ or $n_{\psi_{1,k}^{O(2)}} = \frac{2}{1} = 2$ for $SO(2)$ and $O(2)$ respectively.

The Peter-Weyl theorem plays a vital role in the Fourier transform and the steerable basis.

Fourier Transform Since, for a compact group G, the matrix entries of the irreps both span the space of square integrable functions $L^2(G)$ and constitute an orthonormal basis, these irreps can be used to express a function $f : G \to \mathbb{R}$ through the *inverse Fourier transform*.

Definition 27: Inverse Fourier Transform

Let G be a group and $f : G \to \mathbb{R}$ be a function defined over G. The *inverse Fourier transform* allows us to express f as a linear combination of the irreps of G. This can be formally stated as:

$$f(g) = \sum_{\psi_j \in \widehat{G}} \sqrt{d_j} \mathrm{Tr} \left(\psi_j(g)^\top \mathcal{R}_{\psi_j} \left(\widehat{f}(\psi_j) \right) \right),$$

where g is an element of G, $\widehat{G}$ is the set of irreducible representations of G, d_j is the dimension of the irrep ψ_j, and $\widehat{f}(\psi_j)$ denotes the Fourier coefficient corresponding to the irrep ψ_j which characterises the contribution of ψ_j to the function f.

Similarly, we can also express the Fourier coefficients $\widehat{f}(\psi_j)$ with the *Fourier transform*.

Definition 28: Fourier Transform

Given a group G and a function $f : G \to \mathbb{R}$, the *Fourier transform* of f is given as:

$$\widehat{f}(\psi_j) = \sqrt{d_j} \int_G f(g) \mathcal{R}_{\psi_j}^{-1} \left(\psi_j(g) \right) dg$$

if G is an infinite group, and

$$\widehat{f}(\psi_j) = \sqrt{d_j} \frac{1}{|G|} \sum_{g \in G} f(g) \mathcal{R}_{\psi_j}^{-1} \left(\psi_j(g) \right)$$

if G is a finite group. Here, $\widehat{f}(\psi_j)$ denotes the Fourier coefficient corresponding to the irrep ψ_j.

The *Fourier transform* and its inverse of a signal x and Fourier coefficients $\widehat{x}$ are often denoted as $\mathcal{F}(x)$ and $\mathcal{F}^{-1}(x)$ respectively, where $\mathcal{F}^{-1}(\mathcal{F}(x)) = x$.

Steerable Basis Convolution kernels, which are square integrable functions, belong to the space $L^2(\mathbb{R}^n)$. Parameterising such functions is conveniently achieved using steerable bases[5, 18].

Definition 29: H-Steerable Basis

Given a compact group H with a unitary action on $\mathbb{R}^n$, an H-steerable basis for $L^2(\mathbb{R}^n)$ is a collection of orthogonal functions:

$$\left\{Y_j^{km} \colon \mathbb{R}^n \to \mathbb{R} \;\middle|\; \psi_j \in \widehat{H}, m \leq d_j\right\}.$$

This is an application of the *Peter-Weyl theorem* (Thm. 4), by considering the action of H on $L^2(\mathbb{R}^n)$. Therefore, the stack $\left\{Y_j^{km}\right\}_{m=1}^{d_j}$, denoted by $Y_j^k : \mathbb{R}^n \to \mathbb{R}^{d_j}$, has the following defining property:

$$Y_j^k(g \cdot \boldsymbol{x}) = \psi_j(g) Y_j^k(\boldsymbol{x}) \quad \forall g \in G,\ \boldsymbol{x} \in \mathbb{R}^n$$

For groups $G = (\mathbb{R}^n, +) \rtimes H$, with compact subgroup H, it has been shown that the well-known *Wigner-Eckart theorem* can be generalised to build such G-steerable bases [18].

Chapter 3

Equivariant Neural Networks

In this chapter, we cover *Equivariant Neural Networks*, in particular Steerable Neural Networks. In Section 3.1 we first discuss Group Convolutions, which is a more intuitive extension of regular convolutions to obtain equivariance. In Sections 3.2 and 3.3 we discuss Steerable CNNs (SCNNs) and equivariant MLPs (E-MLPs), respectively. Finally, in Section 3.4 we discuss how the kernel constraint is solved for these types of networks.

3.1 Group Convolutional Networks

Before we investigate the SCNN framework, it is important to provide an overview of the more conventional Group Convolutional Neural Networks (GCNNs) [6]. GCNN are an extension of traditional CNNs, replacing the conventional convolution operation, which is usually used over planar or volumetric images, with a group convolution. It is important to note that, in practice, as well as in this work, cross-correlations are used instead of convolutions. In most literature, as well as in this work, the terms will be used interchangeably.

3.1.1 Regular Convolutions

In the traditional setting, given an input signal $f : \mathbb{R}^n \to \mathbb{R}$ along with a filter $k : \mathbb{R}^n \to \mathbb{R}$, the cross-correlation is defined as:

$$[k * f](\boldsymbol{x}) = \int_{\widetilde{\boldsymbol{x}} \in \mathbb{R}^n} k(\widetilde{\boldsymbol{x}} - \boldsymbol{x}) f(\widetilde{\boldsymbol{x}}) d\widetilde{\boldsymbol{x}}. \tag{3.1}$$

Since the input signals of neural networks are discretised to be $f : \mathbb{Z}^n \to \mathbb{R}$, the kernel is also discretised as $k : \mathbb{Z}^n \to \mathbb{R}$. In the case of 3D convolutions, this produces a kernel $X \times Y \times Z$, where X, Y, and Z correspond to the kernel sizes in each dimension. As a result, the integral is replaced with a sum:

$$\begin{aligned}[k * f](\boldsymbol{x}) &= \sum_{\widetilde{\boldsymbol{x}} \in \mathbb{Z}^n} k(\widetilde{\boldsymbol{x}} - \boldsymbol{x}) f(\widetilde{\boldsymbol{x}}) d\widetilde{\boldsymbol{x}} \\ &= \sum_{\widetilde{\boldsymbol{x}} \in \mathbb{Z}^n} k(\widetilde{\boldsymbol{x}} - \boldsymbol{x}) f(\widetilde{\boldsymbol{x}}).\end{aligned}$$

Here we can safely remove $d\widetilde{\boldsymbol{x}}$, since we assume that all points are evenly spaced and evenly sized, and therefore $d\widetilde{\boldsymbol{x}} = 1$. For simplicity, we will use the continuous setting (Eq. 3.1) in the remainder of this work.

Visually, the cross-correlation can be interpreted as sliding a kernel k over all possible translations $x \in \mathbb{R}^n$ in the signal, where the resulting output is the inner product between the kernel and the space covered by the kernel in the signal. Therefore, the output space is defined in the space of all translations $x \in (\mathbb{R}^n, +)$. Since the kernel is stationary, i.e. the same kernel is applied for each translation, it comes naturally that cross-correlations are equivariant to translations: a translation in the input results in the same translation of the output. We show this more formally:

$$
\begin{aligned}
[k * \mathscr{L}_{t\in(\mathbb{R}^n,+)} f](\boldsymbol{x}) &= \int_{\widetilde{\boldsymbol{x}}\in\mathbb{R}^n} k(\widetilde{\boldsymbol{x}} - \boldsymbol{x}) f(t^{-1}\cdot\widetilde{\boldsymbol{x}}) d\widetilde{\boldsymbol{x}} \\
&= \int_{\widetilde{\boldsymbol{x}}\in\mathbb{R}^n} k(\widetilde{\boldsymbol{x}} - \boldsymbol{x}) f(\widetilde{\boldsymbol{x}} - \boldsymbol{t}) d\widetilde{\boldsymbol{x}} \\
\text{substituting } \widetilde{\boldsymbol{x}} \text{ for } \widetilde{\boldsymbol{x}} + \boldsymbol{t} & \\
&= \int_{\widetilde{\boldsymbol{x}}\in\mathbb{R}^n} k(\widetilde{\boldsymbol{x}} + \boldsymbol{t} - \boldsymbol{x}) f(\widetilde{\boldsymbol{x}}) d\widetilde{\boldsymbol{x}} \\
&= \int_{\widetilde{\boldsymbol{x}}\in\mathbb{R}^n} k(\widetilde{\boldsymbol{x}} - (\boldsymbol{x} - \boldsymbol{t})) f(\widetilde{\boldsymbol{x}}) d\widetilde{\boldsymbol{x}} \\
&= [k * f](\boldsymbol{x} - \boldsymbol{t}) \\
&= \mathscr{L}_{t\in(\mathbb{R}^n,+)} [k * f](\boldsymbol{x}).
\end{aligned}
$$

This meets the requirement for equivariance (Def. 23).

Invariance can be obtained in the discrete setting by reducing the output size of the cross-correlation to a singular point. This can be achieved with a pooling operation, which is performed after regular cross-correlation. Such pooling operations reduce the dimensionality by replacing a block of values in the cross-correlation output by some feature derived from these values, such as the maximum, minimum, or mean value. Taking this block to be the size of the entire output results in an output size of one, in turn, resulting in invariance.

3.1.2 Group Convolutions

Introduced by Cohen and Welling [6], group convolutions generalise this notion of obtaining translation equivariance to other groups. Rather than only applying a mapping across all elements in the translation group $(\mathbb{R}^n, +)$, group convolutions can obtain equivariance to any group G by applying the mapping across all elements in G.

Definition 30: Group Convolution

Given a group G, an input space V, a signal $f : V \to \mathbb{R}$ and a kernel $f : V \to \mathbb{R}$, the *group convolution*, or more accurately group cross-correlation, is defined as:

$$
\begin{aligned}
[k *_G f](g) &= \int_{\widetilde{g}\in G} k(g^{-1}\widetilde{g}) f(\widetilde{g}) d\widetilde{g} \\
&= \int_{\widetilde{g}\in G} \mathscr{L}_g k(\widetilde{g}) f(\widetilde{g}) d\widetilde{g}
\end{aligned}
$$

If $G = (\mathbb{R}^n, +)$ we consequently obtain the regular cross-correlation from Eq. 3.1. Similarly to how the regular cross-correlation defines its output space in the space of all translations, the *group convolution* has its output space defined over all $g \in G$. Therefore, the *group convolution* is equivariant with respect to the group action of G on the input f:

$$
\begin{aligned}
[k *_G \mathscr{L}_{h\in G} f](g) &= \left[k *_G (f \cdot h^{-1})\right](g) \\
&= \int_{\widetilde{g}\in G} k(g^{-1}\widetilde{g}) f(h^{-1}\widetilde{g}) d\widetilde{g} \\
\text{substituting } \widetilde{g} \text{ for } h^{-1}g & \\
&= \int_{\widetilde{g}\in G} k(g^{-1}(h\widetilde{g})) f(\widetilde{g}) d\widetilde{g} \\
&= \int_{\widetilde{g}\in G} k((h^{-1}g)^{-1}\widetilde{g}) f(\widetilde{g}) d\widetilde{g} \\
&= [k *_G f](h^{-1}g) = \mathscr{L}_{h\in G} [k *_G f](g).
\end{aligned}
$$

Again satisfying the requirement for equivariance (Def. 23).

Generally, in equivariant neural networks, we are interested in attaining equivariance to a transformation group (such as rotations) in addition to the translational equivariance that is acquired from the standard convolution. Therefore, we write the group G as $G = (\mathbb{R}^n, +) \rtimes H$, using the semi-direct product, where H is the additional transformation group. This allows us to replace the integral in the group convolution with two integrals; one over the translation group $(\mathbb{R}^n, +)$ and one over the group H. In this case, the output space can be uniquely defined as a pair (t, h), with $t \in (\mathbb{R}^n, +)$ and $h \in H$. Furthermore, the signal $f(t, h)$ and the kernel $k(t, h)$ are now separately indexed by the two elements of this pair.

$$\left[k *_{(\mathbb{R}^n,+)\rtimes H} f\right](t,h) = \int_{\widetilde{t}\in(\mathbb{R}^n,+)} \int_{\widetilde{h}\in H} \mathscr{L}_t \mathscr{L}_h k(\widetilde{t},\widetilde{h}) f(\widetilde{t},\widetilde{h}) \frac{1}{|h|} d\widetilde{t} d\widetilde{h} \tag{3.2}$$

$$= \int_{\widetilde{t}\in(\mathbb{R}^n,+)} \int_{\widetilde{h}\in H} k(h^{-1}(\widetilde{t}-t), h^{-1}\widetilde{h}) f(\widetilde{t},\widetilde{h}) \frac{1}{|h|} d\widetilde{t} d\widetilde{h}. \tag{3.3}$$

Here $|h|$ is the absolute value of the determinant of the matrix-representation of the group element h. The division by this value is necessary due to the potential change in volume induced by the group action. In the case of compact groups, the group action does not impose a change of volume, and therefore this fraction is equal to 1.

For most deep learning tasks, the input is defined in $(\mathbb{R}^n, +)$, whereas the input of the group convolution is defined in $\mathbb{R}^n \rtimes H$. Therefore, the input first must be *lifted* from $\mathbb{R}^n$ to $\mathbb{R}^n \rtimes H$. This is achieved with the lifting convolution:

$$\left[f *^{\text{lift}}_{(\mathbb{R}^n,+)\rtimes H} k\right](t,h) = \int_{\widetilde{t}\in(\mathbb{R}^n,+)} \mathscr{L}_h k(\widetilde{t}-t) f(\widetilde{t}) \frac{1}{|h|} d\widetilde{t}$$

$$= \int_{\widetilde{t}\in(\mathbb{R}^n,+)} k(h^{-1}(\widetilde{t}-t)) f(\widetilde{t}) \frac{1}{|h|} d\widetilde{t}.$$

Note that this is a special case of the regular group convolution, where the integral over the subgroup H of the input from Eqs. 3.2 and 3.3 is dropped.

3.1.3 Discretisation and Limitations of Group Convolutions

Like in the case of regular convolution, group convolutions need to be discretised to be used in deep learning models. Therefore, Eq. 3.3 becomes:

$$\left[k *_{(\mathbb{R}^n,+)\rtimes H} f\right](t,h) = \sum_{\widetilde{t}\in(\mathbb{R}^n,+)} \sum_{\widetilde{h}\in H} k(h^{-1}(\widetilde{t}-t), h^{-1}\widetilde{h}) f(\widetilde{t},\widetilde{h}) \frac{1}{|h|}.$$

As there is now a summation over the group H, it is required that H is a finite group. While we often require some form of rotational equivariance, group convolutions are restricted to the modelling of discrete rotations, e.g. via the *cyclic group* C_N, and are therefore only truly equivariant to these discrete rotations.

Although it is possible to obtain equivariance to a group that is close to a continuous group such as $H = SO(2)$ by selecting N to be a large value, this quickly results in a significant increase in memory consumption and computational complexity of the convolution operation. This is due to the fact that the size of the additional dimension is equal to the order $|H|$ of the group H, which, in the case of the cyclic group, is equal to $|C_N| = N$. Therefore, the total size of the feature field scales linearly with the order of the discretised group H, and the complexity of the convolution scales exponentially.

3.2 Steerable CNNs

In this section, we discuss steerable CNNs (SCNNs) [7, 36, 5]. SCNNs are based on the concepts of GCNNs. As illustrated in the previous section, GCNNs obtain a discretised response over the subgroup

H at every spatial location by separately computing the correlation locally for every transformed version of the kernel. Instead, the aim of SCNNs is to obtain the Fourier coefficients of this response over the subgroup H at each spatial location in terms of carefully chosen equivariant basis-functions. Therefore, these Fourier coefficients represent a continuous function over the subgroup H. This allows us to use infinite groups, while also reducing the memory consumption for discrete groups. It is important to note here that we will restrict ourselves to compact subgroups H of $G = (\mathbb{R}^n, +) \rtimes H$.

3.2.1 Feature Fields

In GCNNs, a layer l_G that is equivariant to the group $G = (\mathbb{R}^n, +) \rtimes H$ maps the input from either the space $\mathbb{R}^n \rtimes H$ or $\mathbb{R}^n$ to real values $\mathbb{R}$. Consequently, intermediate features of GCNNs can be perceived as scalar fields that possess an added dimension for the subgroup (H), separating them from regular CNNs.

Alternatively, layer l_G can also be seen as mapping $l_G : \mathbb{R}^n \to \mathbb{R}^{|H|}$, where it assigns an $|H|$-dimensional feature vector $f(x) \in \mathbb{R}^{|H|}$ to each spatial point $\boldsymbol{x} \in \mathbb{R}^n$. These $|H|$-dimensional feature vectors are also called *group fibers*. Under GCNNs, these feature vectors adhere to the transformation law of the regular representation of G. For a given $g = (t, h) \in G$, the feature field of the layer l_G transforms as:

$$\left[\mathscr{L}_g l_G\right](\boldsymbol{x}) = \rho_{\text{reg}}^H(h) l_G(g^{-1}\boldsymbol{x}) \tag{3.4}$$

Here, $\rho_{\text{reg}}^H : H \to GL(\mathbb{R}^{|H|})$ maps the $|H| \times |H|$ permutation matrices of the regular representation of subgroup H to each element $h \in H$. As such, we refer to the feature fields in GCNNs as *regular fields*. The action of $g = (t, h)$ on the regular field can be visualised as relocating the group fibers through the action of the entire group G (e.g. both rotation and translation) and transforming the group fibers themselves using only the action of h in $g = (t, h)$.

SCNNs extend this concept. They introduce steerable feature fields $f : \mathbb{R}^n \to \mathbb{R}^c$, where each point $\boldsymbol{x} \in \mathbb{R}^n$ is mapped to a c-dimensional feature vector $f(x) \in \mathbb{R}^c$. Every steerable feature field type associates with a group representation ρ of subgroup H (resulting in a ρ-field) and a transformation law[1], explaining the field's transformation under the action of group G. Similar to the transformation in Eq. 3.4, a steerable G-equivariant layer's ρ_H-field $\widehat{l}_G : \mathbb{R}^n \to \mathbb{R}^c$ transforms under $g = (t, h) \in G = (\mathbb{R}^n, +) \rtimes H$ as:

$$\left[\mathscr{L}_g \widehat{l}_G\right](\boldsymbol{x}) = \rho_H(h) \widehat{l}_G(g^{-1}\boldsymbol{x}) \tag{3.5}$$

These representations are often modelled using irreps, particularly for infinite groups H. Using a particular irrep $\psi_1 : H \to GL(\mathbb{R}^c)$ results in a feature field (or ψ_1-field) described as above. In addition to regular fields and irrep fields, there are several other feature fields which will be prevalent in this work.

Scalar Fields *Scalar fields* $f_s : \mathbb{R}^n \to \mathbb{R}$ are functions that assign a single value to each spatial point in $\mathbb{R}^n$. Notable examples are grey-scale images or single-channel voxel grids. Since these values do not describe any direction, the group action of an element $g = (t, h) \in G = (\mathbb{R}^n, +) \rtimes H$ only moves the points in $\mathbb{R}^n$ to a new location in $\mathbb{R}^n$. The group action on the group fibers is therefore modelled through the trivial representation $\rho(h) = 1 \ \forall h \in H$. Therefore, the action of $g = (t, h)$ on a scalar

[1] The way this transformation law works is modelled through the *induced representation* $\text{Ind}_H^G \rho$ of $G = (\mathbb{R}^2, +) \rtimes H$ [23]. This representation describes how the group G and its subgroups H and $(\mathbb{R}^n, +)$ act separately on a particular feature field.

field $f_s : \mathbb{R}^n \to \mathbb{R}$ can be represented as:

$$\begin{aligned}[\mathscr{L}_g f_s](\boldsymbol{x}) &= \rho_H(h) f_s(g^{-1}\boldsymbol{x}) \\ &= f_s(h^{-1}(\boldsymbol{x} - \boldsymbol{t})).\end{aligned}$$

Scalar fields are, therefore, also referred to as *trivial fields*.

Vector Fields Unlike scalar fields, *vector fields* $f_v : \mathbb{R}^n \to \mathbb{R}^n$ associate a directional vector $f_v(\boldsymbol{x}) \in \mathbb{R}^n$ to each spatial point $\boldsymbol{x} \in \mathbb{R}^n$. These vectors transform according to the standard representation of the group H (e.g. rotation matrices in the case of $O(2)$). Therefore, the action of $g = (t, h)$ on a vector field can be represented as:

$$\begin{aligned}[\mathscr{L}_g f_v](\boldsymbol{x}) &= \rho_H(h) f_v(g^{-1}\boldsymbol{x}) \\ &= h \cdot f_v(h^{-1}(\boldsymbol{x} - \boldsymbol{t})).\end{aligned}$$

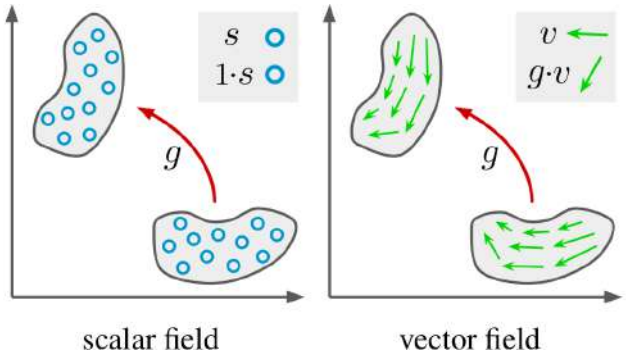


Figure 3.1: Comparison of the effect of a transformation $g \in SE(2)$ on scalar and vector fields. In the scalar field, the features only move and rotate with respect to the image reference. In the vector field, the directions of the features (vectors) also change accordingly. Figure adapted from [36].

As a result, the vectors are re-located and their directions are changed[2] according to the group action of $h \in H$. Due to the use of the standard representation, vector fields are alternatively named *standard fields*. Vector fields are often used to describe the gradients or optical flows of scalar fields. See Figure 3.1 for a visual comparison between vector fields and scalar fields.

Combined Fields Through the direct-sum, it is possible to create a feature field, or *combined field*, that is comprised of multiple feature fields $f_i : \mathbb{R}^n \to \mathbb{R}^{m_i}$ of their own specific type and therefore representation $\rho_i : H \to GL(\mathbb{R}^{m_i})$. This is comparable to using multiple channels, and therefore creating multiple features, in a regular CNN/GCNN. The combined feature field is then simply created by stacking the individual features f_i, where the feature field transforms under the representation constructed through the direct sum $\rho = \bigoplus_i \rho_i$. Due to the block-diagonal nature of the direct sum, this ensures that each feature field f_i transforms independently from the other feature fields. Therefore, the resulting combined feature field $f : \mathbb{R}^n \to \mathbb{R}^c$ contains group fibers of dimension $c = \sum_i m_i$ and the corresponding representation performs the mapping $\rho : H \to GL(\mathbb{R}^c)$.

For most applications, an SCNN's input and output feature field type (and therefore representations) are determined by the task. For example, consider 2D RGB images $f : \mathbb{R}^2 \to \mathbb{R}^3$ as input. These images contain three independent scalar fields to describe the colours for each spatial location. Thus, it would be appropriate to use a stack of three trivial fields as a combined feature field. This results in a ρ-field where $\rho(h) = \bigoplus_{i=1}^3 \rho_i(h) = I_{3\times3}$, since $\rho_i(h) = 1 \quad \forall h \in H$. Likewise, when the task is to predict 2D direction vectors, it would be suitable to pick a single standard field as output.

Irrep Fields A particularly useful type of field for intermediate features is the *irrep field*, especially for infinite subgroups H. Given an irrep $\psi_k : H \to \mathrm{GL}(\mathbb{R}^{d_k})$, an associated irrep field is defined as a field $f_{\psi_k} : \mathbb{R}^n \to \mathbb{R}^{d_k}$, where $f_{\psi_k}(\boldsymbol{x})$ is a d_k-dimensional feature vector at each spatial point $\boldsymbol{x} \in \mathbb{R}^n$.

Under the action of $g = (t, h) \in G = (\mathbb{R}^n, +) \rtimes H$, the irrep field transforms according to the following rule:

$$[\mathscr{L}_g f_{\psi_k}](\boldsymbol{x}) = \psi_k(h) f_{\psi_k}(g^{-1}\boldsymbol{x}).$$

[2] When considering non-compact groups H the vector's magnitudes can also change. But in this framework for SCNNs, we only consider compact groups H. Therefore, only the directions can change.

Irrep fields of different types are often combined into a combined irreps field. The features within these fields adhere to the transformation laws of their respective irreps, or direct-sum of irreps, which makes them particularly useful in applications where the properties of infinite groups H are important. In fact, in Section 3.4.1 we show that for solving the kernel constraints it is convenient to decompose any ρ-field into an irreps field due to the properties of irreps. Since the features of an irreps field transform according to the respective irreps, when considering a combined field of irreps up to a frequency L, the corresponding features are comparable to the Fourier coefficients (Eq. 28) describing the continuous signal over the group H at each spatial point. Therefore, through proper application of the Fourier and inverse Fourier transform, one can convert such a combined irreps field to a *band-limited regular-field* $\rho^H_{\text{bl-reg}} : H \to \mathbb{R}^N$, where N is the number of sampled points by the inverse Fourier transform, and vice versa.

3.2.2 Steerable Convolution

In a neural network, the individual layers perform a mapping between features. As described in the previous section, SCNNs use feature fields to describe the types and transformation laws of intermediate features, and these types can vary from layer to layer. To build an equivariant network with feature fields, it is important to ensure that the mapping of the feature type is equivariant and therefore guarantees that the action of the group G on the input commutes with the layers of the network. In other words, transforming the input feature field using its accompanying transformation law with an element $g \in G$ should result in each consecutive feature field transforming similarly according to its transformation law with element g. It was proven that the most general equivariant linear maps between input and output feature fields of types ρ_{in} and ρ_{out} respectively, are convolution with H-steerable kernels $k : \mathbb{R}^n \to \mathbb{R}^{c_{\text{out}} \times c_{\text{in}}}$, i.e. kernels that satisfy the following constraint [37]:

$$k(h\boldsymbol{x}) = \rho_{\text{out}}(h)k(\boldsymbol{x})\rho_{\text{in}}(h^{-1}) \quad \forall h \in H, \boldsymbol{x} \in \mathbb{R}^n \tag{3.6}$$

We refer to this constraint as the *kernel constraint* or *H-steerability constraint.*

Given the ρ_{in}- and ρ_{in}-fields $f_{\text{in}} : \mathbb{R}^n \to \mathbb{R}^{c_{\text{in}}}$ and $f_{\text{out}} : \mathbb{R}^n \to \mathbb{R}^{c_{\text{out}}}$ respectively, along with a Euclidean group $G = (\mathbb{R}^n, +) \rtimes (H \leq O(n))$ and an H-steerable kernel $k : \mathbb{R}^n \to \mathbb{R}^{c_{\text{out}} \times c_{\text{in}}}$ mapping from f_{in} to f_{out}, the steerable cross-correlation is defined similarly to the regular cross-correlation from Eq. 3.1:

$$f_{\text{out}}(\boldsymbol{x}) = [k * f_{\text{in}}](\boldsymbol{x}) = \int_{\widetilde{\boldsymbol{x}} \in \mathbb{R}^n} k(\widetilde{\boldsymbol{x}} - \boldsymbol{x}) f_{\text{in}} d\widetilde{\boldsymbol{x}}. \tag{3.7}$$

Similarly to the previously discussed types of convolution, this convolution (or cross-correlation) needs to be discretised to be used in deep learning frameworks. Due to the kernel k mapping of any point $\boldsymbol{x} \in \mathbb{R}^n$ to a $c_{\text{out}} \times c_{\text{in}}$ matrix, the resulting discretised kernel is of shape $c_{\text{out}} \times c_{\text{in}} \times X \times Y \times Z$ in the case of 3D convolutions.

Using a group element $g = (t, h) \in G$, it is possible to show that the cross-correlation defined in Eq. 3.7 satisfies the equivariance constraint:

$$\begin{aligned}
[k * (g \cdot f_{\text{in}})](\boldsymbol{x}) &= \int_{\widetilde{\boldsymbol{x}} \in \mathbb{R}^n} k(\widetilde{\boldsymbol{x}} - \boldsymbol{x}) [g \cdot f_{\text{in}}](\widetilde{\boldsymbol{x}}) d\widetilde{\boldsymbol{x}} \\
&= \int_{\widetilde{\boldsymbol{x}} \in \mathbb{R}^n} k(\widetilde{\boldsymbol{x}} - \boldsymbol{x}) \rho_{\text{in}}(h) f_{\text{in}}(h^{-1}(\widetilde{\boldsymbol{x}} - \boldsymbol{t})) d\widetilde{\boldsymbol{x}}
\end{aligned}$$

Using the kernel constraint from Eq 3.6, substitute the kernel $k(\boldsymbol{x})$ for $\rho_{\text{out}}(h)k(h^{-1}\boldsymbol{x})\rho_{\text{in}}(h^{-1})$

$$\begin{aligned}
&= \int_{\widetilde{\boldsymbol{x}} \in \mathbb{R}^n} \rho_{\text{out}}(h) k(h^{-1}(\widetilde{\boldsymbol{x}} - \boldsymbol{x})) \rho_{\text{in}}(h^{-1}) \rho_{\text{in}}(h) f_{\text{in}}(h^{-1}(\widetilde{\boldsymbol{x}} - \boldsymbol{t})) d\widetilde{\boldsymbol{x}} \\
&= \rho_{\text{out}}(h) \int_{\widetilde{\boldsymbol{x}} \in \mathbb{R}^n} k(h^{-1}(\widetilde{\boldsymbol{x}} - \boldsymbol{t}) - h^{-1}(\boldsymbol{x} - \boldsymbol{t})) f_{\text{in}}(h^{-1}(\widetilde{\boldsymbol{x}} - \boldsymbol{t})) d\widetilde{\boldsymbol{x}}
\end{aligned}$$

Substituting $\widetilde{\boldsymbol{x}}' = g^{-1}\widetilde{\boldsymbol{x}} = h^{-1}(\widetilde{\boldsymbol{x}} - \boldsymbol{t})$

$$
\begin{aligned}
&= \rho_{\text{out}}(h) \int_{\widetilde{\boldsymbol{x}} \in \mathbb{R}^n} k(\widetilde{\boldsymbol{x}}' - h^{-1}(\boldsymbol{x} - \boldsymbol{t})) f_{\text{in}}(\widetilde{\boldsymbol{x}}') d\widetilde{\boldsymbol{x}}' \\
&= \rho_{\text{out}}(h) f_{\text{out}}(h^{-1}(\boldsymbol{x} - \boldsymbol{t})) \\
&= \rho_{\text{out}}(h) f_{\text{out}}(g^{-1}\boldsymbol{x})
\end{aligned}
$$

From Eq. 3.5

$$= \left[\mathscr{L}_g f_{\text{out}}\right](\boldsymbol{x}).$$

Therefore, a cross-correlation using an H-steerable kernel meets the equivariance constraint and thus: $[k * (g \cdot f_{\text{in}})] = g \cdot [k * f_{\text{in}}]$.

In Section 3.4 we provide the necessary details to construct kernels that satisfy the kernel constraint for compact groups $H \leq O(n)$.

3.3 Equivariant MLP

Convolutional networks model their features as some feature field $f : \mathbb{R}^n \to \mathbb{R}^c$, where the square integrable convolution kernel $k : \left[L^2(\mathbb{R}^n)\right]^{c_{\text{in}}} \to \left[L^2(\mathbb{R}^n)\right]^{c_{\text{out}}}$ maps between feature fields. Conversely, linear layers model their features without any spatial domain, or which can be interpreted as a feature field $f \in \mathbb{R}^c$ with only one spatial point[3]. These linear layers are parameterised by some weight matrix $W : \mathbb{R}^{c_{\text{in}}} \to \mathbb{R}^{c_{\text{out}}}$. To create H-equivariant MLPs [9], the integral and the indexing by $\boldsymbol{x}$ from the steerable cross-correlation in Eq. 3.7 need to be removed. Therefore, given a linear layer with f_{in} and f_{out} as respective input vector spaces and weight matrix W, the linear map is performed as follows:

$$f_{\text{out}} = W f_{\text{in}}$$

Here, the matrix W is a $c_{\text{out}} \times c_{\text{in}}$ matrix.
Despite the lack of spatial domain, it can be interesting to create H-equivariant MLPs. For example, at the end of an SCNN after the spatial dimensionality has been reduced to 1 through downsampling or pooling, or for tasks where the inputs are simply a singular feature (which can be comprised of multiple feature types). Similarly to the steerability constraint for convolutions, under group $H \leq O(n)$, the H-steerability constraint for a weight matrix W mapping between features of types ρ_{in} and ρ_{out} is as follows:

$$W = \rho_{\text{out}}(h) W \rho_{\text{in}}(h^{-1}) \quad \forall h \in H, \tag{3.8}$$

satisfying the equivariance and interwinter constraints (Def. 23 & 25).

3.4 Solving the Kernel Constraints

In this section we show a derivation of how both the kernel constraints for equivariant MLPs and steerable convolutions are solved through a linear projection. This derivation is equivalent in outcome to the approach in [5], but follows a different perspective. In the process, it will become clear that this projection is dependent on an averaging over the group H, which will become relevant in the future. However, before solving the kernel constraint, it is convenient to first decompose the linear mapping into smaller linear maps using irreducible representations through the irreps decomposition.

[3] Alternatively, although not technically correct, one can almost see this as a mapping $f : \mathbb{R}^0 \to \mathbb{R}^c$.

3.4.1 Irreps Decomposition

The steerability constraints from Def. 3.6 and 3.8 depend on the input and output representations ρ_{in} and ρ_{out}, and are therefore dependent on the type of feature field. This requires the constraint to be solved independently for each pair of input and output types, which can result in a large computational overhead for mixed feature fields. However, [5, 36] propose to decompose ρ_{in} and ρ_{out} into a direct sum of irreps, which is possible for any representation of a compact group H (Thm. 1). Therefore, the kernel constraint from 3.6 can be written as:

$$k(h\boldsymbol{x}) = Q_{\text{out}}^{-1}\left[\bigoplus_{i\in I_{\text{out}}} \psi_i(h)\right] Q_{\text{out}} k(\boldsymbol{x}) Q_{\text{in}}^{-1}\left[\bigoplus_{j\in I_{\text{in}}} \psi_j(h)^{-1}\right] Q_{\text{in}} \quad \forall h \in H, \boldsymbol{x} \in \mathbb{R}^n.$$

Applying a change of variable: $\widetilde{k} = Q_{\text{out}} k Q_{\text{in}}^{-1}$, and noting that compact representations are orthogonal, hence $\forall h \in H: \quad \psi(h)^{-1} = \psi(h)^\top$:

$$\widetilde{k}(h\boldsymbol{x}) = \left[\bigoplus_{i\in I_{\text{out}}} \psi_i(h)\right] \widetilde{k}(\boldsymbol{x}) \left[\bigoplus_{j\in I_{\text{in}}} \psi_j(h)^\top\right] \quad \forall h \in H, \boldsymbol{x} \in \mathbb{R}^n.$$

Due to the block-diagonal nature of the direct-sum, we can visualise this equation as follows:

$$\underbrace{\begin{bmatrix} \widetilde{k}^{i_1 j_1}(h\boldsymbol{x}) & \widetilde{k}^{i_1 j_2}(h\boldsymbol{x}) & \dots \\ \widetilde{k}^{i_2 j_1}(h\boldsymbol{x}) & \widetilde{k}^{i_2 j_2}(h\boldsymbol{x}) & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}}_{\widetilde{k}(h\boldsymbol{x})} = \underbrace{\begin{bmatrix} \psi_{i_1}(h) & & \\ & \psi_{i_2}(h) & \\ & & \ddots \end{bmatrix}}_{\left[\bigoplus_{i\in I_{\text{out}}} \psi_i(h)\right]} \underbrace{\begin{bmatrix} \widetilde{k}^{i_1 j_1}(\boldsymbol{x}) & \widetilde{k}^{i_1 j_2}(\boldsymbol{x}) & \dots \\ \widetilde{k}^{i_2 j_1}(\boldsymbol{x}) & \widetilde{k}^{i_2 j_2}(\boldsymbol{x}) & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}}_{\widetilde{k}(\boldsymbol{x})} \underbrace{\begin{bmatrix} \psi_{j_1}(h)^\top & & \\ & \psi_{j_2}(h)^\top & \\ & & \ddots \end{bmatrix}}_{\left[\bigoplus_{j\in I_{\text{in}}} \psi_j(h)^\top\right]}.$$

Thus, this constraint decomposes into multiple independent constraints:

$$\widetilde{k}^{ij}(h\boldsymbol{x}) = \psi_i(h) \widetilde{k}^{ij}(\boldsymbol{x}) \psi_j(h)^\top \quad \forall h \in H, \boldsymbol{x} \in \mathbb{R}^n,$$

where the index i and j denote $i \in I_{\text{out}}$ and $j \in I_{\text{in}}$ respectively.

The same strategy can be applied to the MLP kernel constraint. Therefore, regardless of chosen input- and output-representation types, the kernel constraint can always be decomposed into multiple kernel constraints using irreducible representations and a change of basis. Consequently, as suggested by [36], when further analysing the kernel constraints, we assume irreducible representations as input and output representations. To remove clutter, we simply use k and W rather than $\widetilde{k}^{ij}$ and $\widetilde{W}^{ij}$ respectively.

3.4.2 Equivariant MLP

The aforementioned irreps decomposition can equally be defined to the MLP's kernel constraint. Therefore, from Eq. 3.8, the kernel constraint for H-equivariant MLPs mapping from the input representation ψ_l to the output representation ψ_J using a weight matrix $W \in \mathbb{R}^{d_j \times d_l}$ is as follows:

$$W = \psi_J(h) W \psi_l(h)^\top \quad \forall h \in H \tag{3.9}$$

This constraint is a linear constraint; all the possible solutions to this constraint live in a vector space which is a subspace of all matrices/linear operators. Therefore, this constraint can be solved by a projection map ξ_H, which projects an unconstrained non-equivariant weight matrix $\widehat{W}$ to an H-equivariant matrix W:

$$W = \xi_H(\widehat{W}) = \int_{h\in H} \psi_J(h) \widehat{W} \psi_l(h)^\top dh, \tag{3.10}$$

This projection indeed satisfies the kernel constraint:

$$\begin{aligned}\psi_J(h')W\psi_l(h')^\top &= \psi_J(h')\int_{h\in H}\psi_J(h)\widehat{W}\psi_l(h)^\top dh\psi_l(h')^\top \\ &= \int_{h\in H}\psi_J(h')\psi_J(h)\widehat{W}\psi_l(h)^\top\psi_l(h')^\top dh \\ &= \int_{h\in H}\psi_J(h'h)\widehat{W}\psi_l(h'h)^\top dh\end{aligned}$$

Substituting $h'h$ for h

$$\begin{aligned}\psi_J(h')W\psi_l(h')^\top &= \int_{h\in H}\psi_J(h)\widehat{W}\psi_l(h)^\top dh \\ &= W.\end{aligned}$$

To ensure that the projection map ξ_H is flexible enough to generate any H-equivariant matrix, we demonstrate that this map is surjective. In other words, we show that for any equivariant matrix W, the projection $\xi_H(W)$ yields W itself. Given an equivariant matrix W:

$$\xi_H(W) = \int_{h\in H}\psi_J(h)W\psi_l(h)^\top dh$$

Since W is equivariant we can substitute $\psi_J(h)W\psi_l(h)^\top$ for W (Eq. 3.9)

$$\begin{aligned}\xi_H(W) &= \int_{h\in H} W dh \\ &= W\end{aligned}$$

This proves that $\xi_H(W) = W$ when W is already an equivariant matrix. Therefore, the projection map ξ_H is surjective, enabling it to generate any H-equivariant matrix. This flexibility is crucial when integrating the projection into a learnable MLP, as it ensures the network's ability to learn any equivariant mapping.

To further solve the kernel constraint, we can safely add a uniform likelihood distribution $\mu(h) = 1 \quad \forall h \in H$ to the projection in Eq. 3.10:

$$W = \xi_H(\widehat{W}) = \int_{h\in H}\mu(h)\psi_J(h)\widehat{W}\psi_l(h)^\top dh. \tag{3.11}$$

This integral is essentially an averaging operator over the group H.

Now it is convenient to vectorise W through columnwise vectorisation $\mathrm{vec}(\cdot)$, where $\mathrm{vec}\,(ABC) = \left(C^\top \otimes A\right)\mathrm{vec}\,(B)$:

$$\underbrace{\mathrm{vec}\,(W)}_{\in\mathbb{R}^{d_j d_l}} = \mathrm{vec}\left(\int_{h\in H}\mu(h)\psi_J(h)\widehat{W}\psi_l(h)^\top dh\right) = \int_{h\in H}\mu(h)(\psi_l\otimes\psi_J)(h)\,\mathrm{vec}\left(\widehat{W}\right)dh.$$

Decomposing the tensor product $(\psi_l \otimes \psi_J)$ into a direct sum of irreps through the Clebsch-Gordan decomposition (Thm. 2):

$$\mathrm{vec}\,(W) = \int_{h\in H}\mu(h)\underbrace{\left(Q\left(\bigoplus_i\bigoplus_r^{[i(lJ)]}\psi_i(h)\right)Q^\top\right)}_{(\psi_l\otimes\psi_J)(h)}\mathrm{vec}\left(\widehat{W}\right)dh. \tag{3.12}$$

Here, the change of basis Q and matrix $\widehat{W}$ do not depend on h, thus:

$$\mathrm{vec}\,(W) = Q\bigoplus_i\bigoplus_r^{[i(lJ)]}\left(\int_{h\in H}\mu(h)\psi_i(h)dh\right)Q^\top\,\mathrm{vec}\left(\widehat{W}\right) \tag{3.13}$$

If we consider the irreps ψ_i as basis-functions, as described in Theorem 4, we can describe the function $\mu(h)$ in terms of its Fourier coefficients (Def. 28):

$$\widehat{\mu}(\psi_i) = \sqrt{d_i} \int_{h \in H} \mu(h) \mathcal{R}_{\psi_i}^{-1} \left(\psi_i(h)\right) dh \quad \forall \psi_i \in \widehat{H}, \tag{3.14}$$

where d_i is the dimensionality of irrep ψ_i, $\mathcal{R}_{\psi_i}^{-1}$ converts an irrep to only its non-redundant columns (Def. 26). Therefore, we reformulate the integral in Equation 3.13 in terms of its Fourier coefficients:

$$\operatorname{vec}(W) = \operatorname{vec}\left(\xi_H\left(\widehat{W}\right)\right) = Q \left(\bigoplus_i \bigoplus_r^{[i(lJ)]} \frac{\mathcal{R}_{\psi_i}\left(\widehat{\mu}(\psi_i)\right)}{\sqrt{d_i}} \right) Q^\top \operatorname{vec}\left(\widehat{W}\right) \tag{3.15}$$

It should be noted that, due to the uniformity of the likelihood distribution, the Fourier coefficient corresponding to the trivial representation (frequency zero), denoted as $\widehat{\mu}(\psi_0)$, is equal to 1. All other Fourier coefficients corresponding to different irreps, and therefore frequencies, are set to zeros. Thus, since the endomorphism basis $\mathcal{C}_{\psi_0} = \left\{\left[1\right]\right\}$,

$$\operatorname{vec}(W) = Q \underbrace{\begin{bmatrix} I_{[0(lJ)]\times[0(lJ)]} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}}_{\left(\bigoplus_i \bigoplus_r^{[i(lJ)]} \frac{\widehat{\mu}(\psi_i)}{\sqrt{d_i}}\right)} Q^\top \operatorname{vec}(\widehat{W}),$$

where $[0(lJ)]$ is the multiplicity of the trivial representation in the decomposition of $\psi_l \otimes \psi_J$. The resulting sparse block-diagonal matrix selects the relevant columns from the change of basis Q. Equivalently, using a sum, we have:

$$\operatorname{vec}(W) = \sum_r^{[0(lJ)]} Q_r Q_r^\top \operatorname{vec}(\widehat{W}),$$

where Q_r is the r-th column of Q. To reduce the number of parameters, we absorb $Q_r^\top \in \mathbb{R}^{1 \times d_l d_j}$ into the weights $\operatorname{vec}(\widehat{W}) \in \mathbb{R}^{d_l d_j}$, resulting in a single weight $\widehat{W}_r \in \mathbb{R}$.

$$\operatorname{vec}(W) = \sum_r^{[0(lJ)]} Q_r \underbrace{\widehat{W}_r}_{Q_r^\top \operatorname{vec}(\widehat{W})}. \tag{3.16}$$

When the input and output representations, ψ_l and ψ_J, are identical, the multiplicity $[0(lJ)]$ of the trivial representation in the Clebsch-Gordan decomposition of $\psi_l \otimes \psi_J$ is equal to the size of the endomorphism basis $|\operatorname{End}_{\psi_J}| = c_{\psi_J}$. In this case, it turns out that the Q_r vectors span the same space as the vectorised endomorphism basis $\mathcal{C}_{\psi_J} = \left\{c_r^J \in \mathbb{R}^{d_j \times d_j}\right\}$. Thus, Eq 3.16 can be re-written as follows:

$$\begin{aligned} \operatorname{vec}(W) &= \sum_r^{[0(lJ)]} \operatorname{vec}(c_r^J) \widehat{W}_r \\ W &= \sum_r^{[0(lJ)]} c_r^J \widehat{W}_r \end{aligned} \tag{3.17}$$

We refer to [5, Appendix B] for a full proof.

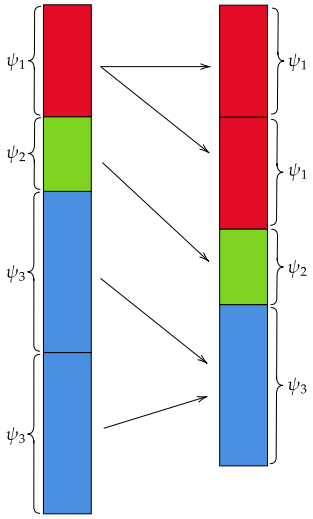


Figure 3.2: Mapping between different input and output representations. Representations are colour-coded. A mapping only (and always) exists between equal irreps.

Conversely, if $\psi_l \neq \psi_J$, the Clebsch-Gordan decomposition lacks a trivial representation and, therefore, multiplicity $[0(lJ)] = 0$. This leads to a null-map in Equations 3.15 through 3.16 as all emerging Fourier coefficients are zero. Consequently, the weight matrix $\widehat{W}$ is also projected to zeros, rendering it a null-map and making these weights redundant. See Figure 3.2 for a visualisation.

The observation made in the preceding paragraphs follows from *Schur's representation lemma* (Thm. 3). Specifically, this lemma establishes that W can act as an intertwiner only if the input and output representations are equivalent; otherwise, W becomes a null-map. This observation extends this by relating Schur's representation lemma to the properties of the tensor product, presenting the result in a manner that is particularly convenient for our purposes.

3.4.3 Steerable Convolution

We follow a similar approach for steerable convolutions. First, assuming input and output irreps, we re-write the kernel constraint from Eq. 3.6 to:

$$k(\boldsymbol{x}) = \psi_J(h)k(h^{-1}\boldsymbol{x})\psi_l(h)^\top \quad \forall h \in H, \boldsymbol{x} \in \mathbb{R}^n$$

Similarly to the MLP case, we insert a uniform likelihood distribution $\mu(h) = 1 \quad \forall h \in H$ and use a linear projection $\Pi_H(\widehat{K})$, which projects the unconstrained square integrable kernel $\widehat{K} \in L^2(\mathbb{R}^n)$ to an H-equivariant kernel k:

$$k(\boldsymbol{x}) = \Pi_H(\widehat{K})(\boldsymbol{x}) = \int_{h\in H} \mu(h)\psi_J \widehat{K}(h^{-1}\boldsymbol{x})\psi_l(g)^\top dg \tag{3.18}$$

Applying columnwise vectorisation:

$$\operatorname{vec}(k(\boldsymbol{x})) = \operatorname{vec}(\Pi_H(\widehat{K})(\boldsymbol{x})) = \int_{h\in H} \mu(h)(\psi_l \otimes \psi_J)(h) \operatorname{vec}\left(\widehat{K}(h^{-1}\boldsymbol{x})\right) dh \tag{3.19}$$

To parameterise the vectorised unconstrained kernel $\operatorname{vec}(\widehat{K}) : \mathbb{R}^n \to \mathbb{R}^{d_j d_l}$, we use an H-steerable basis $\mathcal{B} = \left\{Y_j^k : \mathbb{R}^n \to \mathbb{R}^{d_j} \mid \psi_j \in \widehat{H}, k\right\}$ (Def. 29), and a weight matrix $W_{j,k}$ of shape $d_j d_l \times d_j$:

$$\operatorname{vec}(\widehat{K}(\boldsymbol{x})) = \sum_{j,k} W_{j,k} Y_j^k(\boldsymbol{x}).$$

Noting that $Y_j^k(h \cdot \boldsymbol{x}) = \psi_j(h) Y_j^k(\boldsymbol{x})$ (Def. 29), we re-write Eq. 3.19 using this steerable basis parameterisation:

$$\operatorname{vec}(k(\boldsymbol{x})) = \sum_{j,k} \left[\int_{h\in H} \mu(h) \left(\psi_l \otimes \psi_J\right)(h) W_{jk} \psi_j(h)^\top dh\right] Y_j^k(\boldsymbol{x}). \tag{3.20}$$

We can decompose the tensor product using the *Clebsch-Gordan decomposition* (Thm. 2):

$$(\psi_l \otimes \psi_J)(h) = \sum_{j'} \sum_{s}^{[j'(Jl)]} \left[\mathrm{CG}_s^{j'(Jl)}\right]^\top \psi_{j'}(h) \, \mathrm{CG}_s^{j'(Jl)} .$$

Using this decomposition, Eq. 3.20 becomes:

$$\operatorname{vec}(k(\boldsymbol{x})) = \sum_{j,k} \left[\int_{h\in H} \mu(h) \sum_{j'} \sum_{s}^{[j'(Jl)]} \left[\mathrm{CG}_s^{j'(Jl)}\right]^\top \psi_{j'}(h) \, \mathrm{CG}_s^{j'(Jl)} \, W_{jk} \psi_j(h)^\top dh\right] Y_j^k(\boldsymbol{x}).$$

As the summations and the Clebsch-Gordan coefficients do not depend on h:

$$\text{vec}(k(\boldsymbol{x})) = \sum_{j,k} \sum_{j'} \sum_{s}^{[j'(Jl)]} \left[\text{CG}_s^{j'(Jl)}\right]^\top \left[\int_{h\in H} \mu(h)\psi_{j'}(h)\, \text{CG}_s^{j'(Jl)}\, W_{jk}\psi_j(h)^\top dh\right] Y_j^k(\boldsymbol{x}). \quad (3.21)$$

Subsequently, we define $W_{j,j',k,s} = \text{vec}\left(\text{CG}_s^{j'(Jl)}\, W_{jk}\right) \in \mathbb{R}^{d_j d_{j'}}$ and unvec$(\cdot)$, which reverses the columnwise vectorisation given by vec$(\cdot)$. Using the property $\text{vec}\,(ABC) = \left(C^\top \otimes A\right) \text{vec}\,(B)$, we re-write Eq. 3.21 to:

$$\text{vec}(k(\boldsymbol{x})) = \sum_{j,k} \sum_{j'} \sum_{s}^{[j'(Jl)]} \left[\text{CG}_s^{j'(Jl)}\right]^\top \text{unvec}\left[\int_{h\in H} \mu(h)\,(\psi_j \otimes \psi_{j'})\,(h) W_{j,j',k,s} dh\right] Y_j^k(\boldsymbol{x}) \quad (3.22)$$

Note the similarity between the integral here and the integral in Eq. 3.13 for the MLP case. Consequently, we repeat the steps from the MLP case starting from Eq. 3.12 where we decompose the tensor product through the *Clebsch-Gordan decomposition* (Thm. 2):

$$\begin{aligned}\int_{h\in H} \mu(h)\,(\psi_j \otimes \psi_{j'})\,(h) W_{j,j',k,s} dh &= \int_{h\in H} \mu(h) \left(Q \left(\bigoplus_i \bigoplus_r^{[i(jj')]} \psi_i(h)\right) Q^\top\right) W_{j,j',k,s} dh \\ &= Q \bigoplus_i \bigoplus_r^{[i(jj')]} \left(\int_{h\in H} \mu(h)\psi_i(h) dh\right) Q^\top W_{j,j',k,s}\end{aligned}$$

Noting that this equation is similar to Eq. 3.13, except that here the integral is defined over the irreps in $\psi_j \otimes \psi_{j'}$ rather than $\psi_l \otimes \psi_J$, we apply the same steps as performed in equations 3.14 through 3.17. Thus, if $\psi_j = \psi_{j'}$, yielding:

$$\int_{h\in H} \mu(h)\,(\psi_j \otimes \psi_{j'})\,(h) W_{j,j',k,s} dh = \sum_r^{0[jj']} \text{vec}(c_r^j) W_{j,k,s,r}, \quad (3.23)$$

where the columns Q_r of Q have been absorbed in the weight matrix $W_{j,k,s,r}$. Similarly to the MLP case, if $\psi_j \neq \psi_{j'}$, this equation results in a null-map and $W_{j,k,s,r}$ is only indexed by j and not j'.

Replacing the integral in Eq. 3.22 with Eq. 3.23, and replacing j' with j gives:

$$\text{vec}(k(\boldsymbol{x})) = \Pi_H(\widehat{K})(\boldsymbol{x}) = \sum_{j,k} \sum_{s}^{[j'(Jl)]} \left[\text{CG}_s^{j(Jl)}\right]^\top \text{unvec}\left[\sum_r^{0[jj]} \text{vec}(c_r^j) W_{j,k,s,r}\right] Y_j^k(\boldsymbol{x}) \quad (3.24)$$

$$= \sum_{j,k} \sum_{s}^{[j'(Jl)]} \left[\text{CG}_s^{j(Jl)}\right]^\top \left[\sum_r^{0[jj]} c_r^j W_{j,k,s,r}\right] Y_j^k(\boldsymbol{x}) \quad (3.25)$$

Thus, removing the summation over j', and therefore all linear maps when $\psi_j \neq \psi_{j'}$. This equation is analogous to Eq. 4 in [5], where the summations have been removed in favour of additional indexing.

3.5 Group Restriction

Each consecutive layer of a convolution network generally has its own receptive field [22, 29], and therefore processes features of a specific scale. In natural images, it is common that the exhibited symmetries vary across scales, where smaller features are often symmetric with respect to larger groups than larger features. While this thesis aims to propose a solution to automatically learn the required

degree of equivariance, Weiler and Cesa [36] proposed a *group restriction*. A group restriction allows the user to restrict the modelled equivariance from group H^* to a subgroup $H \leq H^*$ at a certain layer in the network, offering a manual tool to vary the degree of equivariance. Such a restriction requires the features produced by the last H^*-equivariant layer, transforming under group representation ρ^* of H^*, to be re-interpreted in the H-equivariant layer such that they transform under representation ρ of H.

The first step can be realised through the *restricted representation* (Def. 20): $\text{Res}_H^{H^*}\,\psi_{j^*} : H \to GL(\mathbb{R}^n)$, where $\psi_{j^*} : H^* \to GL(\mathbb{R}^n)$ is an irrep of H^*. The resulting restricted representation can be decomposed into a direct-sum of irreps (Thm. 1): $\text{Res}_H^{H^*}\,\psi_{j^*} = \sum_j \sum_t^{[jj^*]} \text{ID}_t^{jj^*}\,\psi_j$, where we use the block-diagonal structure to identify the blocks $\text{ID}_t^{jj^*}$ (Thm. 2). As shown by [5, Appendix B.3], using the H^*-steerable basis $\mathcal{B}^* = \left\{Y_{j^*}^{k^*}\right\}_{j^*}^{k^*}$, the restricted steerable basis $\mathcal{B}$ for H can now be built as follows, using the index $k = (k^* j^* t)$:

$$\mathcal{B} = \left\{ Y_j^{k^* j^* t} \mid Y_j^{k^* j^* t}(\boldsymbol{x}) = \text{ID}_t^{jj^*} \cdot Y_{j^*}^{k^*}(\boldsymbol{x}) \right\}_{j \in \widehat{H},\; Y_{j^*}^{k^*} \in \mathcal{B}^*,\; 1 \leq t \leq [jj^*]}. \tag{3.26}$$

This restricted steerable basis can now be used directly in Eq. 3.24 to create an $H \leq H^*$-equivariant kernel leveraging an H^*-steerable basis.

3.6 Non-Linearities

A GCNN layer maps an input in $\mathbb{R}^n \times GL(\mathbb{R}^{|H|})$ or $\mathbb{R}^n$ to regular features $GL(\mathbb{R}^{|H|})$. As shown by [36], this allows them to use regular pointwise non-linearities acting on the scalar values without breaking equivariance. Instead, a steerable layer might use other feature types, some of which do not allow the use of these pointwise nonlinearities without breaking equivariance.

Consider, for example, a vector field $f : \mathbb{R}^2 \to \mathbb{R}^2$ that transforms under $H = SO(2)$. Taking a single vector, e.g. $x = \begin{bmatrix} -1 \\ 0.5 \end{bmatrix}$, and rotation $h \in H$ of 180 degrees, it is straightforward to observe that pointwise ReLU does not satisfy the equivariance constraint:

$$\begin{aligned} \rho(h)\,\text{ReLU}\left(\begin{bmatrix} -1 \\ 0.5 \end{bmatrix}\right) &\neq \text{ReLU}\left(\rho(h)\begin{bmatrix} -1 \\ 0.5 \end{bmatrix}\right) \\ \rho(h)\begin{bmatrix} 0 \\ 0.5 \end{bmatrix} &\neq \text{ReLU}\begin{bmatrix} 1 \\ -0.5 \end{bmatrix} \\ \begin{bmatrix} 0 \\ -0.5 \end{bmatrix} &\neq \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{aligned}$$

Since non-linearities are essential to learning complex functions, alternative methods are required.

Fourier-based Since an SCNN often uses irrep fields as intermediate features for infinitely large groups H, which describe the signal over the group H in terms of its Fourier coefficients, one method to perform a non-linearity is to sample the Fourier coefficients for N points through the *inverse Fourier transform* (Def. 27) [5]. The resulting features are akin to a regular representation $\rho : G \to GL(\mathbb{R}^N)$, although limited to $N \leq |H|$ permutation axes. These features can then be transformed using regular pointwise non-linearities, after which the Fourier transform is performed again to obtain the original feature field. Consider an irrep feature field $f_{\text{in}} : \mathbb{R}^n \to \mathbb{R}^c$. The Fourier-based non-linearity is defined as follows:

$$f_{\text{out}}(\boldsymbol{x}) = \mathcal{F}\left(\gamma\left(\mathcal{F}^{-1}\left(f_{\text{in}}\left(\boldsymbol{x}\right)\right)\right)\right) \quad \forall \boldsymbol{x} \in \mathbb{R}^n,$$

where γ is some pointwise non-linearity function.

The Fourier transform and the inverse Fourier transform can be described using the FT $\in \mathbb{R}^{c \times N}$ and IFT $\in \mathbb{R}^{N \times c}$ matrices, respectively, where IFT $=$ FT†. Using these matrices, the Fourier-based activation function can be efficiently implemented as follows:

$$f_{\text{out}}(\boldsymbol{x}) = \text{FT}\, \gamma \left(\text{IFT}\, f_{\text{in}}(\boldsymbol{x})\right).$$

It is important to note that, depending on the chosen value for N, the method used for sampling the points, the group H and the non-linearity γ, this method is only *approximately* equivariant. This is due to the fact that activation functions like ReLU introduce sharp non-linearities in the signal, which result in high frequencies in the Fourier transform. In order to optimise equivariance, it is important that $N \geq c$, where c is the size of the combined feature size of the irrep field, and that the N points are sampled uniformly over the space of the group H. The latter can be complicated for certain groups, particularly 3D rotation groups such as $SO(3)$ and $O(3)$ due to the continuous and non-Euclidean space. For these groups, there are no methods to provide exact solutions for any arbitrary N, although methods pertaining to the Thomson problem are capable of providing approximately uniform sampling for such groups.

Since it is expensive to sample a sufficiently large number of points over large three-dimensional groups, such as $O(3)$ and $SO(3)$, Cesa et al. [5] propose to use features of the quotient space $Q = H/K$ rather than the full space H, where $K \leq H$ is a subgroup of H. Although the exact definition of a quotient space is outside the scope of this thesis, the quotient space $Q = H/K$ can be interpreted as the space of H with the subspace K factored out. We refer to [5, Appendix H.2] for more details. The Fourier-based non-linearity then only applies the non-linearity only over this quotient space $Q = H/K$, reducing computational complexity.

Norm-based As shown in [36], any non-linearity acting solely on the norm $||\cdot||_n$ on features transforming through unitary representations of H is H-equivariant. *Norm-ReLUs*, used in [37], are an example of such non-linearities. Here, the norms of the features $f(\boldsymbol{x})$ are transformed as follows: $||f_{\text{out}}(\boldsymbol{x})||_2 = \text{ReLU}(||f_{\text{in}}(\boldsymbol{x})||_2) \quad \forall \boldsymbol{x} \in \mathbb{R}^n$. Alternatively, Weiler et al. [37] introduced *Gated non-linearities*, which scale the norms of the feature vectors $f(\boldsymbol{x})$ using a learnable scalar field $s : \mathbb{R}^n \to \mathbb{R}$ through a sigmoid gate: $\frac{1}{1+e^{-s(\boldsymbol{x})}}$, resulting in: $||f_{\text{out}}(\boldsymbol{x})||_2 = (||f_{\text{in}}(\boldsymbol{x})||_2)\frac{1}{1+e^{-s(\boldsymbol{x})}}$. Although these types of non-linearities maintain perfect equivariance, Weiler and Cesa [36] have shown that they are generally outperformed by the Fourier-based non-linearities.

3.7 Partial Equivariance

The previous sections have mostly dealt with equivariance as a binary constraint, therefore, since some operations are equivariant or not equivariant. In this section, we discuss the notion of *partial* or *approximate equivariance*. We distinguish two causes for partial equivariance, inherent and learnable partial equivariance, and discuss some other methods to achieve learnable partial equivariance.

3.7.1 Inherent Partial Equivariance

Inherent partial equivariance refers to the phenomenon in which the perfect equivariance, often aimed at in the literature, is partially broken due to specific properties of the used data or models. Typically, researchers aim to ensure that the equivariance constraint holds exactly for all elements g in a chosen group G. However, achieving perfect equivariance is often thwarted by factors like the choice of non-linearities, such as Fourier-based non-linearities that are only approximately equivariant, or by the necessity of discretisation when working with discrete data like planar or volumetric grids.

This loss of equivariance can vary across the elements g in the group G, as well as the sampled location $\boldsymbol{x} \in \mathbb{R}^n$, making it useful to measure this loss, or *equivariance error*, over the group and location. This equivariance error quantifies the difference between the ideal and observed result from

Def. 23. Formally, given a group G, and map $\epsilon(g, \boldsymbol{x})$, which gives the mean equivariance error for each element g, the following holds for a partially equivariant map f:

$$\forall g \in G \; \forall \boldsymbol{x} \in \mathbb{R}^n \quad g \cdot f(\boldsymbol{x}) = f(g \cdot \boldsymbol{x}) \pm \epsilon(g, \boldsymbol{x}) \tag{3.27}$$

Alternatively, we collect all elements $g \in G$ for which $\epsilon(g, \boldsymbol{x}) = 0 \quad \forall \boldsymbol{x} \in \mathbb{R}^n$ in a set S. Note that this subset $S \leq G$, along with the group law of G, does not necessarily constitute a group. The mapping f is then only S-equivariant:

$$\forall g \in S \; \forall \boldsymbol{x} \in \mathbb{R}^n \quad g \cdot f(\boldsymbol{x}) \approx f(g \cdot \boldsymbol{x}),$$

and partially G-equivariant (Eq. 3.27). Note the use of $\approx$ in the latter equation; the constraint is not exact, as the equivariance is only guaranteed to hold approximately. This is a result of the subset S not necessarily being closed under group actions. Romero and Lohit [28] show that only if S is a proper subgroup of G, the equivariance constraint holds precisely.

A practical example illustrates this point well. In image processing, images are usually defined using square pixels. These pixels are completely symmetric to the groups D_N and C_N, where $N \in \{1, 2, 4\}$, corresponding to rotations of multiples of 90 degrees. Choosing a group $G = (\mathbb{R}^n, +) \rtimes H$ with a subgroup H that contains rotation elements not aligned with these multiples requires interpolation, resulting in a non-zero equivariance error. Additionally, translations also require interpolation for non-integer translations. Therefore, perfect equivariance can only be obtained for $\mathbb{Z}^n$ translations. As a result, for a $G = (\mathbb{R}^2, +) \rtimes O(2) = SE(2)$-equivariant model acting on pixels, the model is only perfectly $S = (\mathbb{Z}^2, +) \rtimes D_N$-equivariant and partially G-equivariant.

To alleviate these problems, one can use point clouds rather than pixels, which are continuously defined and do not require interpolation. Alternatively, non-standard pixel shapes can be used. For instance, Hoogeboom et al. [12] employed hexagonal pixels to achieve perfect equivariance for $N \in \{1, 2, 3, 6\}$, corresponding to multiples of 60 degrees.

Although these inherent sources of partial equivariance errors are measurable and often significant, in this study we regard a model that is modelled to be G-equivariant, without intentionally breaking equivariance, as being G-equivariant rather than making the distinction between G and S-equivariance.

3.7.2 Learnable Partial Equivariance

While achieving perfect G-equivariance can be beneficial for a G-symmetric task, it can be detrimental for tasks that are not perfectly G-symmetric [27, 35, 34]. Overconstraining a model to achieve perfect G-equivariance may impede the generation of key features, particularly for models that, through some type of pooling operation, force G-invariance. Therefore, it is important to correctly choose the subgroup G, such that the the model is not overly constrained. However, determining the optimal subgroup G a priori can be complicated, since it requires in-depth knowledge of the dataset, which is often not available.

Particularly for equivariant CNNs, where the hierarchical stack of layers ensures that each layer has its own receptive field [29, 22], resulting in each layer potentially requiring a different equivariance subgroup G due to varying symmetries across varying scales [39]. While group restrictions (Section 3.5) offer the ability to manually restrict the modelled equivariance for each layer, fine-tuning the optimal group for each layer can be a cumbersome task. Moreover, group restrictions are only able to restrict equivariance to a sub*group* H of the original group H^*, rather than to any subset S of transformations in group H^*. Petrache and Trivedi [27] show that a solution that is only equivariant with respect to a subset that does not qualify as a group can be sufficient. As a solution, several methods, including the methods presented in this thesis, aim to automatically learn the required degree of equivariance. An overview of some of those methods can be found in Section 4.2.

Chapter 4

Related Work

This chapter serves as a comprehensive review of the literature that is closely related to the research presented in this thesis. We begin by discussing the evolution and key contributions in the field of equivariant neural networks, focussing on the framework of Steerable CNNs (SCNNs), in Section 4.1. Following this, Section 4.2 explores the nuanced topic of learnable partial equivariance and its significance in various applications.

4.1 Equivariant Neural Networks

The concept of equivariant neural networks has garnered significant attention, beginning with the introduction of Group Convolutional Neural Networks (GCNNs) by Cohen and Welling [6]. Equivariant neural networks have been shown to improve performance, as well as generalisation and data-efficiency [40, 42, 36, 38]. Initially, these networks were designed to be equivariant to discrete 2D rotation groups [6, 12]. Over time, the scope of GCNNs has broadened to include continuous 2D rotation groups [42, 38, 2, 26], as well as 3D rotations [41, 40]. Various methods have been employed to achieve these advancements. For instance, Bekkers et al. [2] utilised kernel interpolation techniques to obtain $SO(2)$ equivariant models. On the other hand, Worrall et al. [42] employed circular harmonics instead of regular CNN kernels. To mitigate interpolation artefacts, Weiler et al. [38] introduced the concept of learning steerable filters, although at the cost of constraining the kernel space.

Building upon GCNNs, Steerable CNNs (SCNNs) were introduced to provide a more general framework to achieve equivariance [6]. In this framework, intermediate features are assigned both a feature type and a corresponding representation that dictate their behaviour under transformations. To ensure group equivariance, the kernels in this architecture are linearly constrained. The initial implementations of these constraints were specific to particular groups, such as D_4 by Cohen and Welling [6] and $SO(3)$ by Weiler et al. [37]. However, more recently Weiler and Cesa [36] provided a general solution for any subgroup $H \leq O(2)$ by decomposing arbitrary representations into irreducible ones. This was later extended to any compact group H for any arbitrary choice of dimensions by Cesa et al. [5].

4.2 Learnable Partial Equivariance

While the focus of earlier research was primarily on achieving or approximating strict equivariance, it has been observed that overly strict equivariances can sometimes be detrimental to model performance [27, 35, 34]. Wang et al. [34] differentiates between three types of equivariance: correct, incorrect, and extrinsic. Correct equivariance implies that the model's symmetry matches the ground truth function. Incorrect equivariance indicates a mismatch between the model's symmetry and the ground truth. Extrinsic equivariance occurs when the model transforms in-distribution data into out-of-distribution data. Moreover, Wang et al. [34] developed a general theory to guide the selection of equivariant models based on these categories. Furthermore, Weiler and Cesa [36] demonstrated that

performance can be improved by manually tuning group restrictions in convolutional layers, especially for tasks that exhibit varying symmetries at different scales.

The topic of learnable partial equivariance has gained traction in recent research, offering models the flexibility to adapt their equivariance properties to better suit the data [8, 28, 32, 33]. These approaches often focus on breaking the equivariance for either an entire group $G = (\mathbb{R}^n, +) \rtimes H$ or only for the subgroup H.

For instance, Romero and Lohit [28] parameterised a learnable probability distribution over the elements of the subgroup H of G. Rather than uniformly sampling group elements as in traditional GCNNs, they use this learnt probability distribution to guide the sampling of group elements, achieving a form of partial H-equivariance as a result. van der Ouderaa et al. [32] utilised non-stationary kernels to achieve partial G-equivariance. Proposed by Finzi et al. [8], Residual Pathway Priors (RPP) are a more versatile strategy that can be applied across a variety of equivariant networks. Based on the work of van der Ouderaa et al. [32] and Finzi et al. [8], van der Ouderaa et al. [33] took a more unified approach by employing Bayesian model selection using differentiable Laplace approximations. They learnt layer-wise equivariances as hyper-parameters, offering a comprehensive framework for discovering symmetries at different layers of the model.

In RPPs, each intermediate feature field comprises two layers: one that maintains G-equivariance and another that can be either non-equivariant or equivariant to a specific subgroup of G. This dual-layer structure provides the model with the flexibility to adapt its equivariance properties as needed. The contribution of the two (or more) layers is subsequently controlled by a predetermined prior variance that is enforced through regularisation. To learn the equivariance with respect to a specific subgroup S of H, RPPs require building a separate residual connection for each of these subgroups. This can be a complicated task, as continuous groups have an infinite number of subgroups, thus requiring an infinite number of residual layers. Furthermore, while the enumeration of such subgroups is straightforward in 2D rotation groups, this is not the case in the 3D rotation groups. Specifically, the collection of rotations –continuous or discrete– around any arbitrary 3D axis constitutes a subgroup of $SO(3)$. As the generation of a uniform grid over the space of 3D unit vectors is similarly complex as generating a uniform grid over $SO(3)$ itself, it remains a complex task when opting for a discretisation of the subgroups. As a result, using RPPs it is generally not feasible to learn equivariance with respect to a specific, but unknown subgroup S of a large and continuous group H.

In the following chapter, we discuss a *preliminary* and a *probabilistic* approach to learning partial equivariance in the framework of SCNNs, inspired by the approaches from Romero and Lohit [28] and Finzi et al. [8]. Similarly to RPPs [8], our approaches model the partially equivariant mapping as a fully equivariant mapping with an additional non-equivariant mapping, as discussed in Sections 5.2 and 5.2. However, our approaches offer additional weight sharing, ensuring that the manner of breaking equivariance is consistent throughout the group. Furthermore, similar to the work of Romero and Lohit [28], our *probabilistic* approach learns a likelihood over the group and uses the resulting likelihood as a sampling mechanism. However, there is a key distinction: whereas Romero and Lohit [28] directly samples this likelihood to perform group convolutions, in our case, the learnt likelihood is implicitly integrated within the SCNN framework. Through this likelihood distribution, our approach can model equivariance with respect to any subgroup S of H, or even subsets S that are not proper groups, without the need for additional layers.

Chapter 5

Method

In this chapter, we propose a method to obtain a learnable partial equivariance for SCNNs and equivariant MLPs. As in the previous sections, we assume a group $G = (\mathbb{R}^n, +) \rtimes (H \leq O(n))$, where we focus on learning the degree of equivariance with respect to the subgroup H. We begin by presenting a preliminary strategy in Section 5.1. Subsequently, in Section 5.2.1, we outline a more mathematically informed approach to model and parameterise partial H-equivariance.

5.1 Preliminary Approach

To demonstrate the feasibility of learning equivariance, we propose a preliminary approach to obtain learnable equivariance. Focussing on equivariant MLPs, as established in Section 3.4.2, an equivariant MLP serves as an intertwiner (Def. 25). Consequently, it only maps between the input irrep ψ_l and the output irrep ψ_J when $\psi_l = \psi_J$. This property is inferred by Eq. 3.17, where only in this case the multiplicity $[0(Jl)]$ is greater than 0, hence resulting in a null-map whenever $\psi_l \neq \psi_J$. To introduce a mechanism for breaking equivariance, we allow the projection to map between irreps that are not necessarily identical. Specifically, we re-write equation 3.17 to:

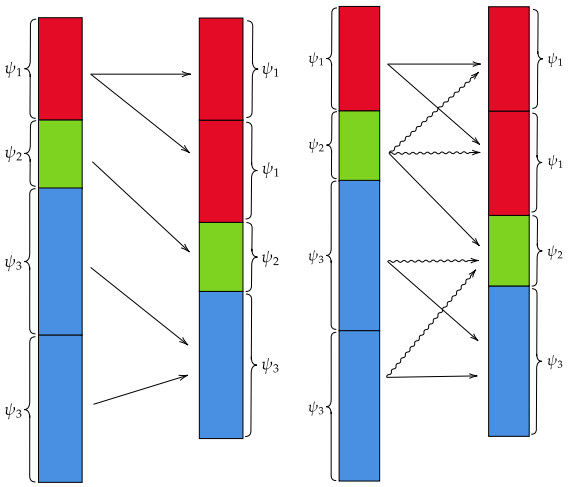
$$W = c^{lJ} \cdot \widehat{W},$$

where $\sum_r^{[0(JL)]} c_r^J$ is replaced by a single matrix $c^{lJ} \in \mathbb{R}^{d_J \times d_l}$. These matrices define the projection operation to map from $\widehat{W}$ to W for each irrep pair (ψ_l, ψ_J), rather than only providing a mapping through the endomorphism basis for equal irreps.

To learn the degree of equivariance, these matrices are initialised such that they provide an equivariant mapping as described in Section 3.4.2. Hence, at initialisation:

$$c^{lJ} = \begin{cases} I_{d_J \times d_l} & \psi_J = \psi_l \\ \mathbf{0}_{d_J \times d_l} & \psi_J \neq \psi_l \end{cases} \tag{5.1}$$

Subsequently, these matrices are registered as learnable weights to allow learning through backpropagation. This way, the projection starts as fully H-equivariant, rendering the weight matrices $\widehat{W}$ unused for non-matching irreps ψ_l and ψ_J.



(a) Initialisation **(b)** After training

Figure 5.1: An example the input and output mapping from the preliminary approach. The left shows an equivariant mapping at initialisation, mapping only equal (colour coded) irreps. Right shows a potential partially equivariant mapping after training, which also contains mappings between non matching irreps (zigzag lines).

However, as training progresses, the model gains the capacity to establish non-zero mappings between non-matching irreps, thereby breaking equivariance, and re-introducing the previously redundant weight matrices W which describe mappings between non-matching irrep pairs (ψ_l, ψ_J). See Figure 5.1 for a visualisation.

We apply the same approach to SCNNs, using matrices $c^{jj'}$, which depend on the irrep pairs $(\psi_{j'}, \psi_j)$ rather than (ψ_l, ψ_J). In this case, Eq. 3.25 becomes:

$$\text{vec}(k(\boldsymbol{x})) = \sum_{j,k} \sum_{j',s} \left[C_s^{j'(Jl)}\right]^\top \left[c^{jj'} \cdot W_{j,j',k,s}\right] Y_j^k(\boldsymbol{x}). \tag{5.2}$$

Here, we re-introduce the summation and indexing in $W_{j,j',k,s}$ over j'.

This approach is comparable to the Residual Pathway Prior (RPP) method from Finzi et al. [8], as both methods perform an equivariant mapping with an additional (learnable) non-equivariant mapping on top. Where our preliminary approach method differentiates itself is in the consistency of equivariance through additional weight sharing. Since RPPs use regular CNN or MLP layers to learn the non-equivariant mapping, there is no guarantee that it learns the same mapping between multiple instances of the same irrep pairs. Conversely, through our method of parameterisation, the preliminary approach does have this guarantee. Furthermore, unlike RPPs, our approach does not add additional layers. As a result, while the number of parameters increases compared to regular SCNNs during the training phase, the resulting filter sizes remain the same and during evaluation the computational cost is therefore the same as with regular SCNNs.

5.2 Probabilistic Approach

While the preliminary approach allows a steerable model to learn to gradually break equivariance, its parameterisation does not provide a straightforward method to inspect the exact degree of equivariance learnt over the group H using some function $H \to \mathbb{R}$. In this Section, we outline an alternative approach which directly models a continuous likelihood distribution over the group in terms of its Fourier coefficients.

5.2.1 Leveraging the Averaging Operator

Building on the preliminary approach outlined in Section 5.1, we consider the role of the uniform likelihood distribution within the context of Eqs. 3.11 and 3.18. These equations utilise the integral over group H as a uniform averaging mechanism, with all elements $h \in H$ equally weighted by the uniform likelihood distribution $\mu(h) = 1 \quad \forall h \in H$. This uniformity ensures perfect equivariance for all elements in H.

The introduction of a non-uniform likelihood distribution, denoted as $\lambda(h)$, disrupts this uniformity and, as a consequence, affects the resulting equivariance. Specifically, Eqs. 3.11 and 3.18 are modified as follows:

$$W = \xi_\lambda(\widehat{W}) = \int_{h \in H} \lambda(h) \psi_J(h) \widehat{W} \psi_l(h)^\top dh$$

and

$$k(\boldsymbol{x}) = \Pi_\lambda(\widehat{K})(\boldsymbol{x}) = \int_{h \in H} \lambda(h) \psi_J \widehat{K}(h^{-1}\boldsymbol{x}) \psi_l(g)^\top dg.$$

The projection operators ξ_λ and Π_λ indicate that the projection is no longer uniformly performed over H, but rather uses a non-uniform likelihood λ. In the following two paragraphs, we show how such a non-uniform projection affects the kernel constraints for both equivariant MLPs and SCNNs.

Equivariant MLPs As illustrated in Eq. 3.15, the integral over the uniform likelihood distribution in Eq. 3.13 correlates with the Fourier coefficient $\widehat{\mu}(\psi_i)$:

$$\operatorname{vec}(W) = \operatorname{vec}\left(\xi_H\left(\widehat{W}\right)\right) = Q\left(\bigoplus_i \bigoplus_r^{[i(Jl)]} \frac{\mathcal{R}_{\psi_i}\left(\widehat{\mu}(\psi_i)\right)}{\sqrt{d_i}}\right) Q^\top \operatorname{vec}\left(\widehat{W}\right).$$

Here $W \in \mathbb{R}^{d_l \times d_J}$ are the weights, ψ_i are irreps appearing in the decomposition of the tensor product $\psi_l \otimes \psi_J$ with multiplicity $[i(Jl)]$, Q is the change of basis associated with the decomposition, $\widehat{\mu}(\psi_i)$ is the Fourier coefficient of irrep ψ_i, $\mathcal{R}_{\psi_i}$ applies the endomorphism basis on the Fourier coefficient (containing only the non-redundant columns), and $\widehat{W} \in \mathbb{R}^{d_l \times d_J}$ are the unconstrained weights. In a setting that is fully equivariant, and thus uniform, the Fourier coefficient $\widehat{\mu}(\psi_i)$ is equal to $\mathbf{0}_{d_i \times n_i}$ for all irreps ψ_i, except for the trivial representation ψ_0 which only appears if $\psi_l = \psi_J$. Here, n_i represents the number of non-redundant columns. We show in Section 3.4.2 that this uniformity allows certain simplifications, leading to the following projection:

$$W = \sum_r^{[0(Jl)]} c_r^J \widehat{W}_r.$$

Where $\widehat{W}_r = Q_r^\top \operatorname{vec}(\widehat{W})$, with Q_r the r-th column of Q, c_r^J is an element of the endomorphism basis of irrep ψ_J, and the multiplicity $[0(Jl)]$ is only larger than zero if $\psi_l = \psi_J$.

However, unlike a uniform likelihood, a non-uniform likelihood $\lambda(h)$ contains non-zero frequencies and therefore also has non-zero Fourier coefficients $\widehat{\lambda}(\psi_i)$ for other non-trivial irreps $\psi_i \neq \psi_0$. For this reason, rather than separately parameterising the projection matrices c^{lJ} and $c^{jj'}$ for each pair (ψ_l, ψ_J) and $(\psi_j, \psi_{j'})$ respectively, we propose to directly parameterise the individual Fourier coefficients $\widehat{\lambda}(\psi_i)$ of the likelihood $\lambda(h)$, thereby parameterising the likelihood function λ itself.

As a result, the previously performed simplifications are no longer valid, and the projection thus becomes:

$$\operatorname{vec}(W) = \xi_\lambda\left(\widehat{W}\right) = c^{lJ} \cdot \underbrace{\widehat{W}}_{Q^\top \operatorname{vec}\left(\widehat{W}\right)}. \tag{5.3}$$

Where $\widehat{W}$ is not a single value, but a vector of size $d_l d_J$, and c^{lJ} is defined using learnable Fourier coefficients $\widehat{\lambda}(\psi_i)$:

$$c^{lJ} = Q\left(\bigoplus_i \bigoplus_r^{[i(Jl)]} \frac{\mathcal{R}_{\psi_i}\left(\widehat{\lambda}(\psi_i)\right)}{\sqrt{d_i}}\right). \tag{5.4}$$

It is important to note here that one specific irrep ψ_i can appear in the tensor product between multiple different pairs of irrep (ψ_l, ψ_J). For that reason, the learnt parameters $\widehat{\lambda}(\psi_i)$ are shared between multiple (ψ_l, ψ_J) pairs.

SCNNs We use a comparable method for the projection of SCNNs. Recall the final projection derived in Section 3.4.3:

$$\operatorname{vec}(k(\boldsymbol{x})) = \Pi_H(\widehat{K})(\boldsymbol{x}) = \sum_{j,k} \sum_s^{[j'(Jl)]} \left[\mathrm{CG}_s^{j(Jl)}\right]^\top \left[\sum_r^{0[jj]} c_r^j W_{j,k,s,r}\right] Y_j^k(\boldsymbol{x}).$$

Where k is the resulting H-equivariant kernel, $\widehat{K}$ the unconstrained kernel, ψ_j the irreps of the steerable basis Y_j^k, $\psi_{j'}$ the irreps appearing (with multiplicity $[j'(Jl)]$) in the tensor product decomposition $\psi_l \otimes \psi_J$, $CG_s^{[j(Jl)]}$ the Clebsch-Gordan coefficients, and $W_{j,k,s,r}$ the unconstrained weights of the steerable basis. Here, the projection is only valid when irrep $\psi_j = \psi_{j'}$, and thus there is no summation and indexing over irrep $\psi_{j'}$ in the unconstrained weights. Additionally, the projection matrix c_r^j

is dependant on irrep ψ_j rather than ψ_J. As a result, incorporating learnable equivariance through parameterisation of the Fourier coefficients of irreps ψ_i in $\psi_j \otimes \psi_J$ yields the following projection:

$$\text{vec}(k(\boldsymbol{x})) = \Pi_\lambda(\widehat{K})(\boldsymbol{x}) \sum_{j,k} \sum_{j',s} \left[C_s^{j'(Jl)}\right]^\top \left[c^{jj'} W_{j,j',k,s}\right] Y_j^k(\boldsymbol{x}). \tag{5.5}$$

Where $W_{j,j',k,s}$ is a vector of size $d_j d_{j'}$ that is projected using $c^{jj'}$ rather than c^{lJ}, following the same parameterisation as in Eq. 5.4.

In both cases, the Fourier coefficients $\widehat{\lambda}(\psi_i) \quad \forall \psi_i$ are initialised such that the resulting projection is fully H-equivariant, as given by:

$$\widehat{\lambda}(\psi_i) = \begin{cases} 1 & i = 0 \\ \mathbf{0}_{d_i \times n_i} & i \neq 0 \end{cases}. \tag{5.6}$$

Subsequently, the Fourier coefficients are updated through backpropagation, permitting a partially H-equivariant projection. Note that in reality, only the Fourier coefficients of irreps appearing in the corresponding Clebsch-Gordan decomposition need to be considered and parameterised.

5.2.2 Normalising the Likelihood Distribution

In order to make the learnt likelihood distributions comparable, it is important to ensure that each likelihood distribution is standardised through normalisation. Since, at uniform initialisation, the mean average likelihood $\int_h \mu(h) dh = 1$, the most straightforward method is to ensure that the learnt likelihood λ is always normalised such that $\int_h \lambda(h) dh = 1$. Rather than enforcing this constraint directly on the learnt Fourier coefficients, the unconstrained distribution is normalised through a Softmax non-linearity before applying them in the projection operation. Since the Softmax non-linearity requires a sampled likelihood distribution, we can use the Fourier-based Softmax non-linearity (Eq. 3.6):

$$\widehat{\boldsymbol{\lambda}} = \mathcal{F}\left(\sigma\left(\mathcal{F}^{-1}\left(\widehat{\boldsymbol{\lambda'}}\right)\right)\right), \tag{5.7}$$

where $\widehat{\boldsymbol{\lambda}}$ is the stack of normalised Fourier coefficients, $\widehat{\boldsymbol{\lambda'}}$ the stack of learnable unnormalised Fourier coefficients, $\mathcal{F}$ and $\mathcal{F}^{-1}$ the respective Fourier and inverse Fourier transform operations, and $\sigma(\cdot)$ is the Softmax non-linearity. Whereas a Softmax non-linearity is often applied to normalise input on a discrete domain to a probability distribution, in our approach the Softmax normalises a discretised input on a continuous domain. Therefore, rather than dividing by a summation to ensure that the sum of the output equals 1, the Softmax uses a division by the mean such that the average likelihood in the likelihood distribution is equal to 1:

$$\lambda(x) = \sigma(\lambda'(x)) = \frac{e^{\lambda'(x) - \max(\lambda')}}{z_\lambda}.$$

Where λ and $\lambda' = \mathcal{F}^{-1}\left(\widehat{\boldsymbol{\lambda'}}\right)$ are the sampled normalised and unnormalised distributions, and $z_\lambda = \frac{1}{N}\sum_{n=1}^{N} e^{\lambda'(n) - \max(\lambda')}$. To prevent rapidly growing terms, the maximum value of λ' is subtracted in the exponents, improving the numerical stability. Note here that N is the number of samples for which the likelihood distribution has been sampled through the inverse Fourier transform.

5.2.3 Interpretability of the Learnt Distributions

The primary advantage of this proposed approach over the preliminary one lies in its interpretability. Through the use of the inverse Fourier transform, the learnt Fourier coefficients can be employed to directly characterise the likelihood distribution $\lambda(h)$ over the subgroup H, facilitating easy inspection of the learnt equivariance. This contrasts with the preliminary approach, where the Fourier coefficients are entangled within the projection matrices, making it non-trivial to reverse-engineer the learnt likelihood distribution. Specifically, a single Fourier coefficient $\lambda(\psi_i)$ may contribute to the projection matrices of multiple irrep pairs, and since these matrices are individually learnt for each irrep pair, there is no guarantee of consistency between different pairs. This problem is solved by using the

additional weight sharing of the probabilistic approach; while each mapping in Figure 5.1 is separately parameterised for each irrep pair with the preliminary approach, through the use of Fourier coefficients, the probabilistic approach shares weights between multiple connections. This results in consistent frequencies in the underlying likelihood distribution.

The direct parameterisation of $\lambda(h)$ also allows for real-time or post-training sampling of the Fourier coefficients $\widehat{\lambda}(\psi_i)$ of a layer to obtain insights into the degree of equivariance for the respective layer. This is particularly useful to understand the symmetries in the task at hand, which may be largely unknown.

For instance, if the learnt likelihood function $\lambda(h)$ is uniform for a proper subgroup $S \leq H$, and 0 elsewhere, then the resulting projection is S-equivariant. Conversely, if $S \subset H$ is not a proper subgroup, the resulting projection is only *approximately* S-equivariant. Alternatively, given $a, b \in H$ for which $\lambda(a) > \lambda(b) > 0$ holds, the projection is at least partially equivariant to both elements a and b, albeit more equivariant with respect to element a than element b. Thus, the equivariance error $\epsilon(b)$ is higher compared to $\epsilon(a)$. See Figure 5.2 for examples.

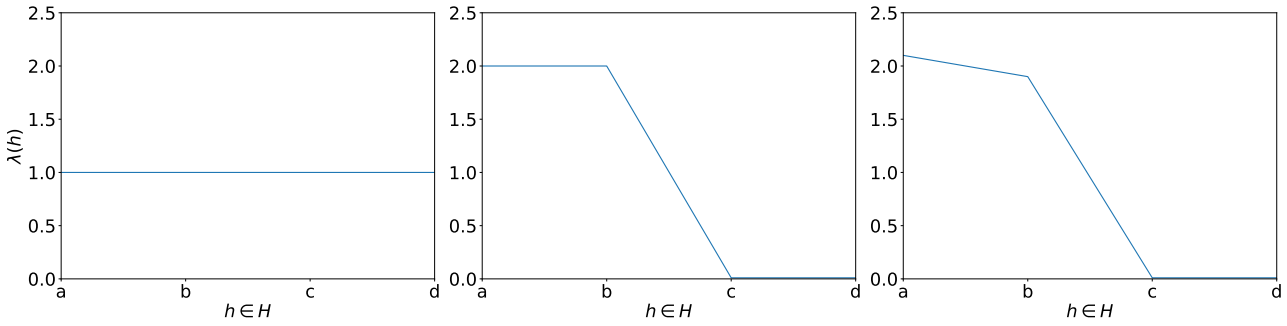


Figure 5.2: Examples of likelihood distributions over a group H with elements $a, b, c, d \in H$. The plot on the left shows a uniform distribution corresponding to full H-equivariance. The likelihood in the middle illustrates a partially equivariance projection, with full $S = \{a, b\}$-equivariance and no equivariance with respect to elements c and d. The plot on the right denotes full $S = \{a\}$-equivariance, with a lower partial equivariance with respect to element b and no equivariance with respect to elements c and d.

Although the proposed method offers a great degree of flexibility in its learnt equivariance, we identify two potential problems with the interpretability of the learnt distributions. Both of these potential problems are a consequence of too much flexibility in the model's ability to capture the required degree of equivariance. Consequently, we propose two regularisation losses to target these problems.

Improper Likelihood Alignment While the Fourier parameterisation enables the reconstruction of the likelihood distribution λ, there is no guarantee that the weights are properly aligned with the learnt likelihood distribution. Recall the learnable equivariant projection for the equivariant MLP projection from Eqs. 5.3 and 5.4:

$$\operatorname{vec}(W) = \operatorname{vec}\left(\xi_\lambda\left(\widehat{W}\right)\right) = Q\left(\bigoplus_i \bigoplus_r^{[i(Jl)]} \frac{\mathcal{R}_{\psi_i}\left(\widehat{\lambda}(\psi_i)\right)}{\sqrt{d_i}}\right) Q^\top \widehat{W}.$$

Note here that the change of basis $Q^\top$ is not absorbed in the weights $\widehat{W}$. To illustrate the problem, we perform the MLP projection given by $\xi_{\mathscr{R}_h\lambda}(\widehat{W}')$, where $\widehat{W}' = (\psi_l \otimes \psi_J)(h)\widehat{W}$. Here, the likelihood distribution undergoes a transformation through the right-action of an element $h \in H$ using the right-regular representation $\mathscr{R}_h$ (Def. 19).

$$\begin{aligned}\operatorname{vec}\left(\xi_{\mathscr{R}_h\lambda}\left(\widehat{W}'\right)\right) &= Q\left(\bigoplus_i \bigoplus_r^{[i(Jl)]} \frac{\mathcal{R}_{\psi_i}\left(\widehat{[\mathscr{R}_h\cdot\lambda]}(\psi_i)\right)}{\sqrt{d_i}}\right) Q^\top \widehat{W}' \\ &= Q\left(\bigoplus_i \bigoplus_r^{[i(Jl)]} \frac{\mathcal{R}_{\psi_i}\left(\widehat{\lambda}(\psi_i)\right)\psi_i(h)^\top}{\sqrt{d_i}}\right) Q^\top \widehat{W}'\end{aligned}$$

Owing to the block-diagonal structure, the equation can be formulated as:

$$\operatorname{vec}\left(\xi_{\mathscr{R}_h\lambda}\left(\widehat{W}'\right)\right) = Q\left(\bigoplus_i \bigoplus_r^{[i(Jl)]} \frac{\mathcal{R}_{\psi_i}\left(\widehat{\lambda}(\psi_i)\right)}{\sqrt{d_i}}\right) Q^\top Q\left(\bigoplus_i \bigoplus_r^{[i(Jl)]} \psi_i(h)^\top\right) Q^\top \widehat{W}'$$

Recall that the second direct-sum over ψ_i is simply equal to the Clebsch-Gordan decomposition of the tensor product $(\psi_l \otimes \psi_J)(h)$. Therefore:

$$\operatorname{vec}\left(\xi_{\mathscr{R}_h\lambda}\left(\widehat{W}'\right)\right) = Q\left(\bigoplus_i \bigoplus_r^{[i(Jl)]} \frac{\mathcal{R}_{\psi_i}\left(\widehat{\lambda}(\psi_i)\right)}{\sqrt{d_i}}\right) Q^\top (\psi_l \otimes \psi_J)(h)^\top \widehat{W}'$$

since $\widehat{W}' = (\psi_l \otimes \psi_J)(h)\widehat{W}$

$$\begin{aligned}\operatorname{vec}\left(\xi_{\mathscr{R}_h\lambda}\left(\widehat{W}'\right)\right) &= Q\left(\bigoplus_i \bigoplus_r^{[i(Jl)]} \frac{\mathcal{R}_{\psi_i}\left(\widehat{\lambda}(\psi_i)\right)}{\sqrt{d_i}}\right) Q^\top (\psi_l \otimes \psi_J)(h)^\top (\psi_l \otimes \psi_J)(h)\widehat{W} \\ &= Q\left(\bigoplus_i \bigoplus_r^{[i(Jl)]} \frac{\mathcal{R}_{\psi_i}\left(\widehat{\lambda}(\psi_i)\right)}{\sqrt{d_i}}\right) Q^\top \widehat{W} \\ &= \operatorname{vec}\left(\xi_\lambda\left(\widehat{W}\right)\right)\end{aligned}$$

As a result, performing these transformations on the likelihood λ and the unconstrained weights $\widehat{W}$ has no effect on the resulting constrained weights W. Consequently, the model is able to freely shift the learnt likelihood distribution by shifting the weights accordingly, thus diminishing the interpretability of the learnt likelihood distribution.

To address this concern, we propose to constrain the Fourier coefficients such that the maximum likelihood is specifically aligned with the identity element, denoted as $h = e$. The theoretical justification for this approach lies in the fact that, when the identity element is applied, it becomes intrinsically impossible to violate the principle of equivariance for that element. Therefore, to accurately represent this property within the learnt likelihood function, it is essential to enforce that the maximum likelihood is achieved when $h = e$. Thus, formally the following must hold:

$$\forall h \in H\colon \lambda(h) \leq \lambda(e) \tag{5.8}$$

Owing to the Fourier parameterisation, directly implementing this constraint via a hard constraint is unfeasible. As a viable alternative, we opt for the application of a soft constraint to achieve the desired outcome. Specifically, we define an alignment error; an error that can be added as a loss term to the task-dependent loss function – e.g., mean squared error – during training:

$$D_{\text{align}}(\lambda) = \max(\lambda) - \lambda(e). \tag{5.9}$$

Provided that the likelihood λ has been sampled using a sufficient number of samples N, this error term is equal to zero if the constraint 5.8 holds, and is greater than zero otherwise.

Regaining Equivariance In equivariant neural networks, global equivariance is obtained by ensuring that each individual operation is equivariant. Once a particular operation breaks the equivariance, this equivariance cannot be regained in subsequent operations. Since our method permits an independent parameterisation of the likelihood distributions of subsequent layers, the resulting likelihood distributions do not necessarily reflect this permanent loss of equivariance. For instance, when the equivariance with respect to element $h \in H$ is reduced in layer n, layer $m > n$ is able to model a likelihood distribution exhibiting an increased level of equivariance for element h, which is not representative of the actual state of equivariance in layer m.

To achieve representative likelihood distributions, we employ Kullback-Leibler divergence (KL divergence) [17]. KL divergence is a non-negative statistical distance measure between two *probability* distributions. It is commonly used as a regularisation term in deep learning, most notably for variational autoencoders [14]. Given a distribution P along with a reference distribution Q, KL-divergence is defined as:

$$D_{KL}(P \mid\mid Q) = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx.$$

Or, in a discretised setting:

$$D_{KL}(P \mid\mid Q) = \sum_{x \in X} p(x) \log\left(\frac{p(x)}{q(x)}\right). \tag{5.10}$$

Note that, due to the weighting of the logarithmic term by $p(x)$, the KL divergence is not a symmetrical measure, thus $D_{KL}(P \mid\mid Q) \neq D_{KL}(Q \mid\mid P)$; a larger value of $p(x)$ at a given point results in a stronger weight of the term $\log\left(\frac{p(x)}{q(x)}\right)$. Therefore, taking the equivariance of layer n as distribution Q and the equivariance from layer $n+1$ as distribution P aligns with our objectives. This approach assigns a higher weight when layer $n+1$ models a high degree of equivariance, particularly when it models the reacquisition of previously lost equivariance.

The definition of Eq. 5.10 requires a sampled likelihood distribution. As a result, the accuracy of the KL-divergence is dependant on the number and the uniformity of the samples used by the inverse Fourier transform. Therefore, we show that KL-divergence over a reference distribution λ_0 and a distribution λ_1 can be computed directly on the Fourier coefficients by first re-writing Eq. 5.10:

$$\begin{aligned} D_{KL}(\lambda_1 \mid\mid \lambda_0) &= \sum_{x \in X} \lambda_1(x) \log \lambda_1(x) - \sum_{x \in X} \lambda_1(x) \log \lambda_0(x) \\ &= \boldsymbol{\lambda_1}^\top \log \boldsymbol{\lambda_1} - \boldsymbol{\lambda_1}^\top \log \boldsymbol{\lambda_0}, \end{aligned}$$

Where $\boldsymbol{\lambda_1}$ and $\boldsymbol{\lambda_0}$ are the vectors stacking $\lambda_1(x)$ and $\lambda_0(x)$. Since the Fourier transform $\mathcal{F}(\boldsymbol{\lambda_1}) = \widehat{\boldsymbol{\lambda_1}}$ is an orthogonal operation:

$$D_{KL}(\lambda_1 \mid\mid \lambda_0) = \widehat{\boldsymbol{\lambda_1}}^\top \widehat{\log \boldsymbol{\lambda_1}} - \widehat{\boldsymbol{\lambda_1}}^\top \widehat{\log \boldsymbol{\lambda_0}} \tag{5.11}$$

Here $\widehat{\log \boldsymbol{\lambda_1}}$ and $\widehat{\log \boldsymbol{\lambda_1}}$ are the Fourier coefficients of the logarithms of the respective λ_1 and λ_0 distributions. Although the Fourier coefficients $\widehat{\boldsymbol{\lambda_1}}$ and $\widehat{\boldsymbol{\lambda_0}}$ are directly available, since these are the normalised Fourier coefficients obtained in Section 5.2.2, their logarithms are not directly available without first computing them. Fortunately, through the normalisation approach in Section 5.2.2, the unnormalised likelihood distribution relates to the logarithm of the normalised distribution:

$$\begin{aligned} \log \lambda(x) &= \log\left(\frac{e^{\lambda'(x) - \max(\lambda')}}{z}\right) \\ &= \lambda'(x) - \max(\lambda') - \log z. \end{aligned}$$

Again, using the orthogonality of the Fourier transform, we can simply add these additional terms to Eq. 5.11 and replace $\widehat{\log \boldsymbol{\lambda_1}}$ and $\widehat{\log \boldsymbol{\lambda_0}}$ with $\widehat{\boldsymbol{\lambda_1'}}$ and $\widehat{\boldsymbol{\lambda_0'}}$. As a result:

$$D_{KL}(\lambda_1 \mid\mid \lambda_0) = \widehat{\boldsymbol{\lambda_1}}^\top \widehat{\boldsymbol{\lambda_1'}} - \max(\lambda_1') - \log z_1 - \widehat{\boldsymbol{\lambda_1}}^\top \widehat{\boldsymbol{\lambda_0'}} + \max(\lambda_0') + \log z_0 \tag{5.12}$$

Since we now use the Fourier coefficients directly, the entire distribution is used to compute the KL-divergence, rather than only a finite number of sampled points. This should result in a more stable computation. However, this computation is still dependent on the normalisation process, which does involve sampling of the distribution. Consequently, computing the KL-divergence as described here can still result in minor inaccuracies.

5.2.4 Bandlimiting the Likelihood Distribution

While the approach outlined in this section allows for a flexible and interpretable parameterisation of the degree of equivariance, it also results in a significant increase in the number of learnable parameters. Consider, for instance, the fully equivariant and partially equivariant MLP projection operations from Eqs. 3.17 and 5.3:

$$W = \sum_r^{[0(Jl)]} c_r^J \widehat{W}_r$$

and

$$\text{vec}(W) = Q \left(\bigoplus_i \bigoplus_r^{[i(Jl)]} \frac{\mathcal{R}_{\psi_i}\left(\widehat{\lambda}(\psi_i)\right)}{\sqrt{d_i}} \right) \cdot \underbrace{\widehat{W}}_{Q^\top \text{vec}\left(\widehat{W}\right)} .$$

The former is parameterised by only $[0(Jl)]$ single weights $\widehat{W}_r$ for each irrep ψ_J, where $[0(Jl)] \in \{0, 1, 2, 4\}$. On the other hand, the partially equivariant projection parameterises the projection with a vector $\widehat{W}$ of size $d_l d_J = \sum_i [i(jj')] \cdot d_i$ for each unique pair of irreps ψ_l and ψ_J, which in most cases is significantly larger than $[0(Jl)]$.

Recall the composition of the projection matrix $c^{lJ} \in \mathbb{R}^{d_l d_J \times d_l d_J}$ that is used to project $\widehat{W}$ through a matrix multiplication:

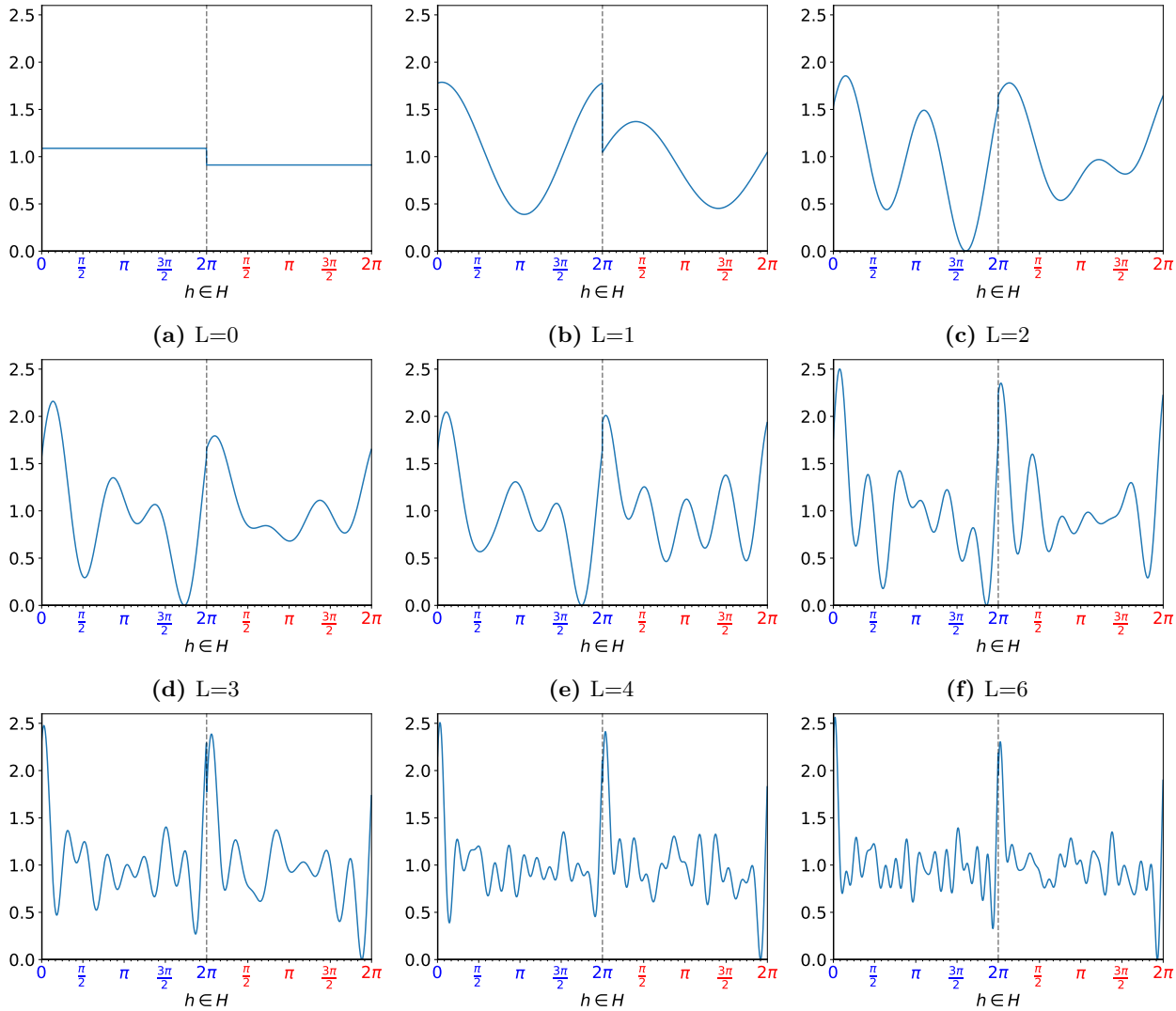
$$c^{lJ} = Q \left(\bigoplus_i \bigoplus_r^{[i(Jl)]} \frac{\mathcal{R}_{\psi_i}\left(\widehat{\lambda}(\psi_i)\right)}{\sqrt{d_i}} \right).$$

The size of this matrix is determined by the irreps ψ_i appearing in the tensor product $\psi_J \otimes \psi_l$, where the Fourier coefficients $\widehat{\lambda}(\psi_i)$ describe the frequencies appearing in the learnt likelihood distribution λ. To reduce the number of parameters, we can impose a limit on the maximum frequency L of irreps ψ_i. Accordingly, all Fourier coefficients for irreps with a frequency larger than L are set to zeros. Similarly to the fully equivariant setting in Eq. 3.15, the direct-sum results in a partially sparse block-diagonal matrix, effectively eliminating certain columns of Q. Since the dimensionality of $\widehat{W}$ is determined by the size of $Q^\top$, this also reduces the dimensionality of the vector $\widehat{W}$.

Another advantage of bandlimiting the irreps ψ_i is that this provides a means of regularising the complexity of the learnt degree of equivariance. The tensor product between irreps of rotational frequency k and f decomposes into irreps with a maximum frequency of $k + f$ and a minimum frequency of $|k - f|$. For example, consider input and output irreps ψ_l and ψ_J, both ranging from frequency 0 through 4. Consequently, in the decomposition of the tensor product $\psi_l \otimes \psi_J$, the irreps ψ_i range from frequencies 0 through 8. By setting $L = 4$, only the irreps ψ_i from frequency 0 to 4 are taken into account. As a result, the learnt likelihood distribution is limited to frequencies up to 4, reducing the flexibility of the equivariance and the risk of overfitting.

Besides the computational advantage, there are several situations where such a regularisation through bandlimiting might prove to be useful. Consider, for example, a task where the user is certain that the

task exhibits continuous rotational symmetries, but not certain whether the task exhibits reflection symmetries. By training a partially equivariant steerable model with $H = O(2)$ and a bandlimit of $L = 0$ on the rotational frequencies the likelihood distribution will only be modelled in terms of the irreps $\psi_{0,0}^{O(2)}$ and $\psi_{1,0}^{O(2)}$. Subsequently, the resulting likelihood distribution is only able to differentiate between reflecting and not reflecting. In Figure 5.3 various levels of bandlimiting are shown.



(a) L=0 **(b)** L=1 **(c)** L=2

(d) L=3 **(e)** L=4 **(f)** L=6

(g) L=8 **(h)** L=12 **(i)** L=16

Figure 5.3: Examples of random signal over $O(2)$ with various levels of bandlimiting. The dotted line indicates the border between the unreflected (left) and reflected domain (right).

Chapter 6

Implementation Details

This chapter covers the details of the implementations of the theoretical procedures described in the previous chapter. All code implementations are performed as an extension of the PyTorch `escnn` library [5], the successor of the `e2cnn` library [36]. These libraries provide an intuitive environment for the setup of steerable neural networks with respective $G \leq SE(n)$ and $G \leq SE(2)$ symmetries.

First, Section 6.1 provides an overview of the algorithms used by [5] and in our work to build the (partially) H-steerable kernels. Next, in Section 6.2, we highlight the notion of sharing equivariance between multiple layers. Finally, Section 6.3 describes the design choices regarding the regularisation losses defined in Section 5.2.3.

6.1 Building the H-Steerable Basis

In this section we discuss and compare the algorithms to construct H-steerable bases on a space X for the three discussed settings: fully equivariant, as implemented in [5], our preliminary partially equivariant (Sec. 5.1), and probabilistically partially equivariant (Sec. 5.2). In all these settings, we generalise the algorithms to group restrictions through the use of an H^*-steerable basis $\mathcal{B}^*$ to generate an H-steerable basis through the restricted representation as described in Section 3.5. Although the derivation for E-MLPs is distinct from SCNNs in that it does not require a steerable basis, we generalise the approach. Specifically, for E-MLPs, we take a steerable basis $\mathcal{B}^* = \{Y_{j^*}^{k^*} : \mathbb{R} \to 1\}_{j^*=0}^{k=0}$ and consider a space $X = \{0\}$, which contains a single element.

Full Equivariance For the fully equivariant setting, as implemented in the `escnn` library, we adapt the pseudocode from [5, Algorithm 1]. This algorithm operates on several inputs: A large compact group $H^* \leq O(n)$ endowed with irreducible representations $\widehat{H^*} = \{\psi_{j^*}\}_{j^*}$ and an H^*-steerable basis $\mathcal{B}^* = \left\{Y_{j^*}^{k^*}\right\}_{j^*}^{k^*}$. Furthermore, it requires a compact subgroup $H \leq H^*$ with irreducible representations $\widehat{H} = \{\psi_j\}_j$ and corresponding endomorphism basis $\{c_r^j\}$ for all $\psi_j \in \widehat{H}$. In addition, the algorithm takes as input the initial and final representations ψ_l and ψ_J respectively, along with a space X. We present the resulting pseudocode in Algorithm 1.

Here, $\boldsymbol{x}$ is one particular coordinate in the X-space where the filters are applied. It is important to note that when considering a static space X for each input, such as when performing convolutions on pixel or voxel grids, none of the intermediate results in this algorithm change. As a result, in this case, Algorithm 1 only needs to be performed at initialisation. Subsequently, each sampled non-zero basis vector in the resulting basis $K_{j(j^*i^*t)sr}(\boldsymbol{x})$ is normalised to a unit vector for numerical stability. The resulting $K_{j(j^*i^*t)sr}(\boldsymbol{x})$ basis can then be steered through the learnable weights. On the contrary, performing convolutions in a non-static space X, such as pointclouds, requires rebuilding the steerable basis at every forward pass and does not allow normalisation.

Algorithm 1 Generate a fully equivariant H-Steerable basis on space X (adapted from [5, Algorithm 1 Section 3])

Prerequisites: $\rho_{\text{in}} = \psi_l$ and $\rho_{\text{out}} = \psi_J$, H^*-steerable basis $\mathcal{B}^* = \left\{Y_{j^*}^{k^*}\right\}_{j^*}^{k^*}$, $\widehat{H} = \{\psi_j\}_j$ and $\widehat{H^*} = \{\psi_{j^*}\}_{j^*}$, $\boldsymbol{x} \in X$

1: $\left\{c_r^j\right\}_r \leftarrow$ basis for the endomorphism space of ψ_j, for all $\psi_j \in \widehat{H}$ ▷ Def. 26
2: CG^{lJ}, $\{[j(lJ)]\}_j \leftarrow$ decompose$(\psi_l \otimes \psi_J)$ ▷ Thm. 2
3: **for all** $Y_{j^*}^{k^*} \in \mathcal{B}^*$ **do**
4: $\quad \text{ID}^{\text{j}^*}, \{[jj^*]\}_j \leftarrow \text{decompose}\left(\text{Res}_H^{H^*}\, \psi_{j^*}\right)$ ▷ Thm. 2 & Def. 20
5: $\quad$ **for all** $j \in \widehat{H} : [jj^*] > 0, s \leq [j(lJ)], t \leq [jj^*], c_r^j \in \left\{c_r^j\right\}_r$ **do**
6: $\quad\quad Y_j^{(j^*k^*t)}(\boldsymbol{x}) \leftarrow \text{ID}_t^{jj^*} \cdot Y_{j^*}^{i^*}(\boldsymbol{x})$ ▷ Eq. 3.26
7: $\quad\quad K_{j(j^*i^*t)sr}(\boldsymbol{x}) \leftarrow \text{unvec}\left(\left[\text{CG}_{\text{s}}^{\text{j(lJ)}}\right]^\top \cdot c_r^j \cdot Y_j^{(j^*i^*t)}(\boldsymbol{x})\right)$ ▷ Eq. 3.25
8: $\quad\quad$ **yield** $K_{j(j^*i^*t)sr}(\boldsymbol{x})$

Learnable Equivariance: Preliminary Approach The preliminary approach (Sec. 5.1) enables us to build partially equivariant projections, and therefore kernels. Unlike the fully equivariant projection in 1, this algorithm does not require a basis for the endomorphism space $\left\{c_r^j\right\} \forall \psi_j \in \widehat{H}$ for each $r \leq [0(jj)]$, but a set of learnable matrices $\left\{c^{jj'}\right\}^{jj'}$. These matrices are initialised such that the projection is uniform, as defined in Eq. 5.1, with optional added standard Gaussian noise to the zero matrices with a standard deviation of 1×10^{-10}. In Section 8.1 we show why this noise might be necessary. Otherwise, the inputs remain the same. The resulting pseudocode can be found in Algorithm 2.

Algorithm 2 Preliminary Approach: Generate a learnable equivariant H-steerable basis on space X

Prerequisites: $\rho_{\text{in}} = \psi_l$ and $\rho_{\text{out}} = \psi_J$, H^*-steerable basis $\mathcal{B}^* = \left\{Y_{j^*}^{k^*}\right\}_{j^*}^{k^*}$, $\widehat{H} = \{\psi_j\}_j$ $\widehat{H^*} = \{\psi_{j^*}\}_{j^*}$, $\boldsymbol{x} \in X$ and a set of learnable matrices $\left\{c^{jj'}\right\}^{jj'}$

1: CG^{lJ}, $\{[j'(lJ)]\}_j \leftarrow$ decompose$(\psi_l \otimes \psi_J)$ ▷ Thm. 2
2: **for all** $Y_{j^*}^{k^*} \in \mathcal{B}^*$ **do**
3: $\quad \text{ID}^{\text{j}^*}, \{[jj^*]\}_j \leftarrow \text{decompose}\left(\text{Res}_H^{H^*}\, \psi_{j^*}\right)$ ▷ Thm. 2 & Def. 20
4: $\quad$ **for all** $j \in \widehat{H} : [jj^*] > 0, t \leq [jj^*]$ **do**
5: $\quad\quad Y_j^{(j^*k^*t)}(\boldsymbol{x}) \leftarrow \text{ID}_t^{jj^*} \cdot Y_{j^*}^{i^*}(\boldsymbol{x})$ ▷ Eq. 3.26
6: $\quad\quad$ **for all** $j' \in \widehat{H} : [j'(lJ)] > 0, s \leq [j'(lJ)]$ **do**
7: $\quad\quad\quad K_{j(j^*i^*t)j's}(\boldsymbol{x}) \leftarrow \text{unvec}\left(\left[\text{CG}_{\text{s}}^{\text{j'(lJ)}}\right]^\top \cdot c^{jj'} \cdot Y_j^{(j^*i^*t)}(\boldsymbol{x})\right)$ ▷ Eq. 5.2
8: $\quad\quad\quad$ **yield** $K_{j(j^*i^*t)j's}(\boldsymbol{x})$

There are two main changes compared to Algorithm 1. First, an additional loop in line 6 over the irreps $\psi_{j'}$, since $\psi_j = \psi_{j'}$ is no longer enforced to require equivariance. Second, the final projection is obtained in line 7 $K_{j(j^*i^*t)j's(\boldsymbol{x})}$, where learnable projection matrices are used in favour of fixed endomorphisms. While Algorithm 1 is executed only once during the initialisation of a layer when the space X is static, it is important to note that $K_{j(j^*i^*t)j's(\boldsymbol{x})}$ is now dependent on learnable and non-static parameterised projection matrices. Consequently, the final projection calculated in line 7 must be recomputed whenever these parameters are updated. However, in order to comply with current implementations in the `escnn` library, this projection is recomputed at each forward call in practice. Similarly to the fully equivariant setting, each non-zero basis vectors in the resulting basis is normalised to a unit vector if the convolutions are performed on discretised grids. To reduce computational overhead, the vector norms used to perform the normalisation are computed and stored at initialisation.

Subsequent forward calls then use the same norms to compute the normalisation.

Learnable Equivariance: Probabilistic Approach The implementation of the probabilistic approach of Section 5.2 is presented in Algorithm 3. Similarly to the preliminary approach, it learns to build partially equivariant projections by introducing an additional loop over the $\psi_{j'}$ (line 6) and producing an (indirectly) learnable partially equivariant projection for each $(\psi_j, \psi_{j'})$ pair (line 8). The main difference between these approaches is how the projection matrices $c^{jj'}$ are constructed. Instead of directly parameterising these matrices, the probabilistic approach models the breaking of equivariance through the learnable irrep Fourier coefficients representing a likelihood distribution over the group H. These Fourier coefficients, and therefore the corresponding likelihood distribution, are used to construct the corresponding projection matrices in line 7. This approach enables inspection of the underlying learnt likelihood distribution, improving interpretability. Like the projection matrices in the preliminary approach, the non-stationary projection matrices require the entire algorithm to be recomputed at each forward call to comply with the current implementations in the `escnn` library. We perform the same normalisation steps for the normalisation of the resulting basis as for the preliminary approach.

Algorithm 3 Probabilistic Approach: Generate a learnable equivariant H-steerable basis on space X

Prerequisites: $\rho_{\text{in}} = \psi_l$ and $\rho_{\text{out}} = \psi_J$, H^*-steerable basis $\mathcal{B}^* = \left\{Y_{j^*}^{k^*}\right\}_{j^*}^{k^*}$, $\widehat{H} = \{\psi_j\}_j$

$\widehat{H^*} = \{\psi_{j^*}\}_{j^*}$, $\boldsymbol{x} \in X$

1: CG^{lJ}, $\{[j'(lJ)]\}_{j'} \leftarrow \text{decompose}(\psi_l \otimes \psi_J)$ ▷ Thm. 2

2: **for all** $Y_{j^*}^{k^*} \in \mathcal{B}^*$ **do**

3: $\quad \text{ID}^{j^*}, \{[jj^*]\}_j \leftarrow \text{decompose}\left(\text{Res}_H^{H^*} \psi_{j^*}\right)$ ▷ Thm. 1 & Def. 20

4: $\quad$ **for all** $j \in \widehat{H} : [jj^*] > 0, t \leq [jj^*]$ **do**

5: $\quad\quad Y_j^{(j^*k^*t)}(\boldsymbol{x}) \leftarrow \text{ID}_t^{jj^*} \cdot Y_{j^*}^{i^*}(\boldsymbol{x})$ ▷ Eq. 3.26

6: $\quad\quad$ **for all** $j' \in \widehat{H} : [j'(lJ)] > 0, s \leq [j'(lJ)]$ **do**

7: $\quad\quad\quad c^{jj'} \leftarrow \text{construct\_projection}(j', j)$ ▷ Algorithm 4

8: $\quad\quad\quad K_{j(j^*i^*t)j's}(\boldsymbol{x}) \leftarrow \text{unvec}\left(\left[\text{CG}_{\text{s}}^{\text{j'(lJ)}}\right]^\top \cdot c^{jj'} \cdot Y_j^{(j^*i^*t)}(\boldsymbol{x})\right)$ ▷ Eq. 5.5

9: $\quad\quad\quad$ **yield** $K_{j(j^*i^*t)j's}(\boldsymbol{x})$

The implementation of the construction of the resulting projection matrices $c^{jj'}$ is described in more detail in Algorithm 4. As inputs, this algorithm requires a set of learnable matrices $\left\{\widehat{\lambda'(\psi_i)}\right\}_i$, initialised such that the projection is uniform (Eq. 5.6). Here, the normalisation in line 4 is performed as described in Eq. 5.7. In line 4, we utilise the inverse Fourier transform $\mathcal{F}^{-1}$. For supported groups, we adopt a regular sampling to ensure an evenly spaced grid. The number of samples, denoted by N, is set to equal the combined size of the bandlimited irreps, ensuring equivariance. For unsupported groups such as $SO(3)$ and $O(3)$, we sample based on the Thomson problem provided by the `escnn` library. To compensate for the non-exact sampling method, we set N to 1.5 times the combined size of the bandlimited irreps.

In addition, some caching methods are included to improve computational performance. Most notably, the Fourier coefficients are only re-normalised after the unnormalised Fourier coefficients have updated through backpropagation (line 1). Furthermore, the $c^{jj'}$ matrices are cached for each $(\psi_j, \psi_{j'})$ pair (line 16).

Algorithm 4 Probabilistic Approach: Constructing partially H-equivariant projection matrices through Fourier parameterisation

Prerequisites: $\widehat{H} = \{\psi_i\}_i$, $j' \in \widehat{H}$, $j \in \widehat{H}, L \in \mathbb{N}^+$, set of learnable unnormalised Fourier coefficients $\left\{\widehat{\lambda'}(\psi_i) \mid i \in \widehat{H},\ \text{freq}(\psi_i) \leq L\right\}_i$

if Fourier coefficients $\left\{\widehat{\lambda'}(\psi_i)\right\}_i$ were recently updated **then**
  remove all cached c^{mn} matrices for all $m, n \in \widehat{H}$
  $\widehat{\boldsymbol{\lambda'}} \leftarrow \text{stack}\left(\left\{\widehat{\lambda'}(\psi_i)\right\}_i\right)$
  $\widehat{\boldsymbol{\lambda}} \leftarrow \mathcal{F}\left(\sigma\left(\mathcal{F}^{-1}\left(\widehat{\boldsymbol{\lambda'}}\right)\right)\right)$ ▷ Eq. 5.7
  $\left\{\widehat{\lambda}(\psi_i)\right\}_i \leftarrow \text{unstack}\left(\widehat{\boldsymbol{\lambda}}\right)$
if $c^{jj'}$ is cached **then**
  $c^{jj'} \leftarrow$ obtain cached $c^{jj'}$
else
  $Q, \{[i(jj')]\}_i \leftarrow \text{decompose}(\psi_j \otimes \psi_{j'})$ ▷ Thm. 2
  $Q \leftarrow$ columns of Q corresponding to ψ_i with frequency $\leq L$ ▷ Sec. 5.2.4
  diagonal $\leftarrow$ empty list
  for all $\widehat{\lambda}(\psi_i) \in \left\{\widehat{\lambda}(\psi_i)\right\}_i : [i(jj')] > 0$ **do**
    for $r \in [i(jj')]$ **do**
      diagonal.add$\left(\mathcal{R}_{\psi_i}\left(\widehat{\lambda}(\psi_i)\right)\right)$ ▷ Def. 26
  $D \leftarrow \text{diag}(\text{diagonal})$ ▷ Using direct-sum
  $c^{jj'} \leftarrow Q \cdot D$
  cache $c^{jj'}$ for irrep pair $(\psi_j, \psi_{j'})$
yield $c^{jj'}$

6.2 Sharing Equivariance

Algorithms 3 and 4 enable the learning of partially equivariant mappings in SCNNs or E-MLPs, which can be applied to obtain corresponding weight projections that are also partially equivariant. Traditional CNNs consist of a hierarchical sequence of convolution layers, each specialised in modelling features at a unique scale. Therefore, it is straightforward for such CNNs to learn the degree of equivariance individually, resulting in different degrees of equivariance for each scale. However, such a layered specialisation in scale does not translate well to MLPs. As such, it is perhaps more prudent to have a single, shared degree of equivariance across all layers in an E-MLP.

To implement this shared degree of equivariance, each layer receive a unique `layer_id` by default, which can be overridden by the user at initialisation of the layer. If two, or more, layers receive the same id, then the parameters parameterising the degree of equivariance are shared between these layers, resulting in the sharing of the degree of equivariance between these layers.

This notion of shared equivariance also finds potential utility in other architectures or architectural elements, such as residual connections and auto-encoders. For instance, in a residual network, a skip-connection could share its degree of equivariance with the layers it bypasses. Similarly, in auto-encoders, it could be advantageous for the first layer of the encoder to share its degree of equivariance with the last layer of the decoder, and so forth.

6.3 Interpretability losses

This section discusses the implementation details of the losses introduced in Section 5.2.3, which are aimed at improving the interpretability of the likelihoods for the probabilistic approach. The following two paragraphs cover KL-divergence and *alignment loss* respectively.

KL-divergence The KL-divergence from Eq. 5.12 is defined over the likelihood distribution of two layers. The intended use is that the two layers are consecutive, with the likelihood of the first layer acting as reference. To account for varying architectures, for example a layer which is preceded by two different layers, the user can provide a set τ, containing pairs of `layer_id`'s. Subsequently, the total KL-divergence loss is computed as the mean average KL-divergence between the pairs:

$$\mathcal{L}_{\text{KL}} = \frac{1}{|\tau|} \sum_{(m,n)\in\tau} D_{\text{KL}}(\lambda_m \,||\, \lambda_n)$$

from Eq. 5.12

$$= \frac{1}{|\tau|} \sum_{(m,n)\in\tau} \widehat{\boldsymbol{\lambda_m}}^{\top} \widehat{\boldsymbol{\lambda'_m}} - \max(\lambda'_m) - \log z_m - \widehat{\boldsymbol{\lambda_m}}^{\top} \widehat{\boldsymbol{\lambda'_n}} + \max(\lambda'_n) + \log z_n.$$

Where λ'_m is the unnormalised likelihood distribution of the layer with `layer_id` m, with corresponding Fourier coefficients $\widehat{\boldsymbol{\lambda'_m}}$, Fourier coefficients of the normalised distribution $\widehat{\boldsymbol{\lambda_m}}$ and the logarithmic term $\log z_m$. Since the aim of the KL-divergence is to incentivise the likelihood of layer m to be a subset of the reference likelihood of layer n, the terms corresponding to layer n are treated as constants and therefore do not contribute to the KL-divergence loss. As a result, the model can only alter the likelihood of layer m to be closer to the likelihood of layer n, and not the other way around. Finally, we allow the user to pair a `layer_id` m with $n = \text{None}$, rather than an existing `layer_id` n. In this case λ'_n (and by extension λ_n) is taken to be the uniform likelihood distribution and subsequently results in the likelihood of layer m being pushed towards a uniform and therefore equivariant projection.

Alignment Loss The alignment error D_{align} from Eq. 5.9 prevents the misalignment by ascribing a nonzero error when the likelihood $\lambda(e)$ for the identity element $e \in H$ is not the maximum likelihood appearing in the entire likelihood distribution. To compute this error, we use the points sampled by the inverse Fourier transform used for the normalisation in line 4 of Algorithm 4. As the sampling is a (near) uniform sampling over the group H, using these points should ensure that the likelihood is shifted in approximately the correct direction. However, for continuous rotational groups, particularly continuous rotational 3D groups, more precision may be required. Therefore, for such continuous groups, 100 additional points are randomly sampled around the identity element e.

For 2D rotation groups, we sample the rotation angles of these points from a normal distribution $\mathcal{N}(0, 0.2)$, where the standard deviation is 0.2 radians. In contrast, for 3D rotation groups, the sampling procedure is more involved. We start with the quaternion representation of the identity element, represented as $\mathbf{q} = [1, 0, 0, 0]$. Random perturbations are then added to each component of this quaternion, drawn from a normal distribution $\mathcal{N}(0, 0.1)$. Formally, if $\mathbf{d} = [d_1, d_2, d_3, d_4]$ are the random deviations, the perturbed quaternion is given by:

$$\mathbf{q}_{\text{perturbed}} = \mathbf{q} + \mathbf{d} = [1 + d_1, 0 + d_2, 0 + d_3, 0 + d_4]$$

Subsequently, these randomly sampled quaternions are normalised to unit quaternions. In this manner, the uniformly sampled points serve to shift the likelihood in case the likelihood is strongly misaligned, and the additional points aid in shifting the likelihood distribution when it is nearly aligned.

Similarly to KL-divergence, the total alignment loss $\mathcal{L}_{\text{align}}$ is defined as the mean average alignment error of all likelihood distributions $\lambda_i \in \Lambda$, where Λ is the set of learnt likelihood distributions.

$$\mathcal{L}_{\text{align}} = \frac{1}{|\Lambda|} \sum_{\lambda_i\in\Lambda} D_{\text{align}}(\lambda_i)$$

Chapter 7

Evaluation Methodology

This chapter contains an outline of our evaluation methodology used to obtain the results in Chapter 8. First, in Section 7.1 we outline the three main classes of datasets, along with their variations. Section 7.2 contains descriptions of the model architectures used for the varying experimental setups.

7.1 Datasets

In this thesis we evaluate our learnable partially equivariant SCNNs and E-MLPs on various datasets to assess the behaviour of our approaches, both in terms of task-specific performance and in terms of the learnt degrees of equivariance.

7.1.1 Vectors

To assess the performance of learnable partial equivariance in E-MLPs, we employ a simple `Vectors` dataset. This dataset consists of 1000 two-dimensional vectors. Each vector is generated with an independently and uniformly sampled angle between 0 and 2π radians that represents the angle between the vector and the positive y-axis. Moreover, the ℓ^2 norms of the vectors are between 0 and $\sqrt{2}$. Importantly, the dataset is designed to be flexible in its training targets: a model can be trained to predict either the cosine of the angle or the norm of the vectors as binary regression targets, or it can be trained to predict both the cosine of the angle and the norm simultaneously as two regression targets.

While the prediction of the norm of a vector is a clear $O(2)$-invariant task, predicting the angle is inherently non-invariant. Specifically, when the cosine of the angle is considered, vectors that are opposite to each other produce the most significant difference in the regression target. This implies that an appropriate equivariance likelihood distribution would predominantly exhibit a frequency signal of one, reaching its minimum likelihood at rotations of π. Given the contrasting nature of these two tasks, one invariant and one non-invariant, this dataset serves as a valuable resource to juxtapose the performance of various model architectures.

7.1.2 Double MNIST

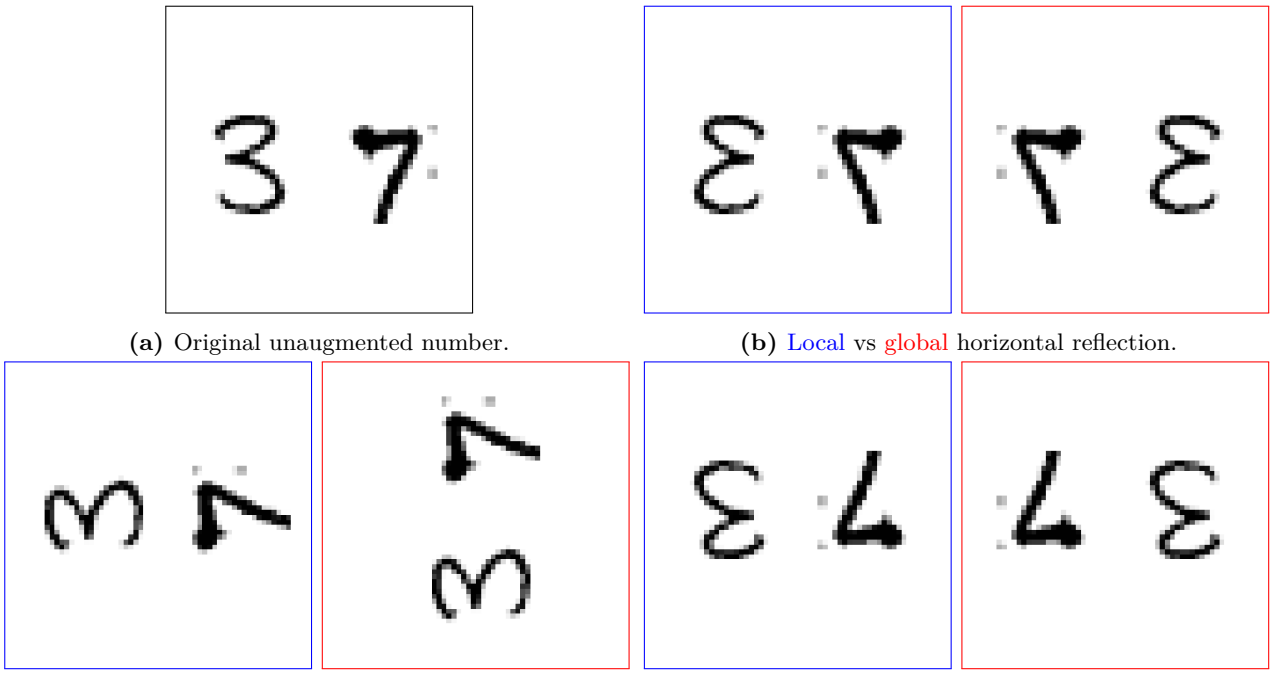
While the `Vectors` dataset focusses on E-MLPs, the main motivation for incorporating a learnable degree of equivariance is the varying degrees of scale in CNNs acting on planar or volumetric images. For planar images, we build the `Double MNIST` dataset, which is an adaptation of the popular MNIST dataset [21]. Whereas the original MNIST dataset contains $60,000$ 28×28 training images of single handwritten digits numbered 0 through 9, our `Double MNIST` dataset contains $10,000$ 56×56 images containing double-digit numbers ranging from 0 through 99. We created these images by sampling 100 images of the two required digits from the original MNIST dataset for each of the 100 double-digit numbers. Before combining the images of the two digits, they are both independently and randomly transformed by a random element of one of the pre-determined groups in Table 7.1 without altering

the label. To keep interpolation artefacts consistent, before augmentation each digit is rotated by a random angle $\theta \in [0, 2\pi)$ and consequently rotated by $-\theta$. This introduces interpolation artefacts in each image, even if they are not transformed at all. Afterwards, the double-digit images are concatenated horizontally and padded so that the resulting image has a size of 56×56 pixels.

Group	Group Type	Symmetries	# of elements
C_1	Cyclic/None	None	1
C_4	Cyclic	90 Degree rotations	4
D_1	Dihedral	Horizontal reflection	2
D_4	Dihedral	Horizontal reflection + 90 degree rotations	8
$SO(2)$	Special Orthogonal	Continuous rotations in 2D	∞
$O(2)$	Orthogonal	Horizontal reflection + continuous rotations in 2D	∞

Table 7.1: Overview of available symmetries for our `Double MNIST` dataset.

Although the resulting datasets contain clear symmetries in the individual digits, these symmetries do not always occur in the entire double-digit number. For instance, consider `Double MNIST` with $O(2)$ augmentations. The digits can be rotated and reflected individually without changing the label. However, applying a reflection on the whole number changes the corresponding label; for instance, 37 becomes 73. See Figure 7.1 for an example.



(a) Original unaugmented number.

(b) Local vs global horizontal reflection.

(c) Local vs global 90 degree rotation.

(d) Local vs global 180 degree rotation.

Figure 7.1: Examples of the effect of local vs global augmentations on the double-digit number 37. Figure a shows the original image. Figures b, c, and d show the effect of applying a local (per-digit) compared to a global augmentation using a horizontal reflection, 90 and 180 degree rotation, respectively. In each case, it becomes apparent that while a local augmentation acting purely on the digit does not change the corresponding number, but applying the augmentation globally changes the number to 73 in the case of figures b and d, resulting in incorrect equivariance. Conversely, c shows that a global rotation by 90 degrees results in a non-existing double-digit number and is hence extrinsic equivariance.

7.1.3 Medical MNIST

Finally, to evaluate our approach on 3D groups, we use several volumetric datasets from the MedMNIST dataset collection [44]. This collection contains challenging, diverse, and standardised biomedical datasets using real-world data. The different datasets seem to exhibit different types and degrees of

symmetries [16], making these datasets suitable for evaluating our approach.

MedMNIST provides several planar and volumetric biomedical datasets, where each sample is labelled and of size 28×28 or $28 \times 28 \times 28$. For the evaluation of 3D groups, we focus on the following three volumetric datasets.

OrganMNIST3D The `OrganMNIST3D` dataset [3, 43] consists of volumetric CT scans of the following 11 human organs that are also the classification targets: liver, right kidney, left kidney, right femur, left femur, bladder, heart, right lung, left lung, spleen, and pancreas. All samples are aligned to the abdominal view such that the sagittal, coronal and axial planes are aligned with the x, y, and z axes. This dataset is intriguing due to its wide variety of symmetric patterns. While some organs and structures display evident symmetries that an equivariant model might leverage, distinguishing between the left and right versions of certain organs can pose challenges for models that are rotationally or reflectionally equivariant.

SynapseMNIST3D `SynapseMNIST3D` [44] focusses on 3D microscopy scans of neural synapses in adult rats to classify synapses as excitatory or inhabitory. These 3D scans are obtained by a multi-beam scanning electron microscope. Historically, using microscopy, one of the main features used to classify a synapse as inhibitory or excitatory has been its structural symmetry; excitatory synapses are generally symmetric, while inhibitory synapses tend to be asymmetric[11]. However, there are other potential discriminating features [15, 31, 30]. Although biomedically informed decisions are outside the scope of this thesis, a model with a learnable and interpretable degree of equivariance can give some insight into which type of features the model prefers. This can in turn yield potentially valuable theoretical information for related fields.

NoduleMNIST3D The `NoduleMNIST3D` dataset [1] contains thoracic CT scans of lung nodules that are categorised as low or high malignancy. International guidelines cite nodule size (and its growth rate) as the predominant indicator for malignancy levels [4, 24, 10, 19]. Given that size-based features remain invariant to any compact group, we employ this dataset primarily to assess how our approach performs on tasks that exhibit strong symmetry and, as a result, presumably demand minimal equivariance loss.

7.2 Model Architectures

For each of the three types of datasets, we use a separate base structure for the neural networks. While the base structure for each of the datasets remains mostly the same throughout our experiments, certain aspects, such as the type of convolution/linear layer, are determined by the individual experiment and method. By default, these base architectures are used for all approaches in our experiments. This includes the RPP approach [8], which we include as one of the baselines, as RPPs are comparable to some degree to our approach and are also trivially applicable to SCNNs. By using the same base architecture, we ensure that each model has the same number of output features at each intermediate layer.

MLP For `Vectors` we use a simple three-layer MLP, where the first two layers are followed by a 1D batch normalisation and a non-linearity. The complete structure can be found in Figure 7.2. This structure has four main configurations, see Table 7.2 for an overview. For each of the equivariant configurations, we use an irrep-field containing irreps up to frequency 4 as intermediate features, and a trivial representation as output feature to ensure an invariant mapping. For the configurations with a learnt degree of equivariance, each layer learns an individual degree of equivariance by default. Furthermore, for the probabilistic approach we use a bandlimit of $L = 4$ for the learnt likelihood distribution.

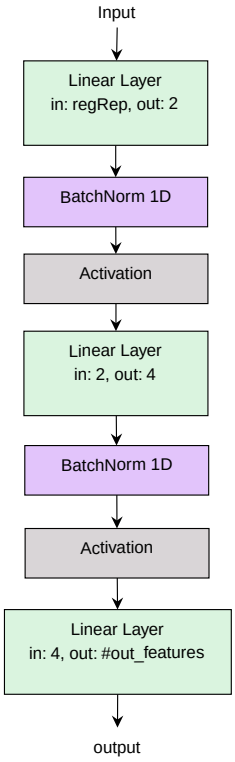


Figure 7.2: The base structure of the MLP networks. In and out refer to the number of input and output features.

Configuration	Equivariance	Non-linearity
MLP	Non-equivariant	ELU
E-MLP	$O(2)$ [5]	Gated/FourierELU
PE-MLP (Preliminary) (ours)	Partial $O(2)$ (Sec. 5.1)	Gated/FourierELU
PE-MLP (Probabilistic) (ours)	Partial $O(2)$ (Sec. 5.2)	Gated/FourierELU

Table 7.2: Overview of the four main MLP architectures. FourierELU refers to a Fourier-based non-linearity (Sec. 3.6) using a pointwise ELU non-linearity acting on the sampled signal.

2D CNN For the `Double MNIST` and its variations, we employ a CNN with five blocks as its primary structure. This is followed by a 1×1 convolution and a final classification layer. Notably, the 1×1 convolution is executed on a 1×1 spatial domain, resulting in translation *invariance*.

Each block in this architecture comprises a convolution, 2D batch normalisation, and a non-linearity. See Figure 7.3a for a complete overview of the base architecture. It is worth noting that the 56×56 images are initially upsampled to 57×57. Subsequently, the input is masked with a circular mask with a radius of 28 pixels. As a result, information is lost at the corners of the input, thereby enhancing rotational equivariance.

Configuration	Description	Partial Equivariance	Non-linearity
CNN	CNN	N/A	ELU
SCNN	Steerable CNN	None [5]	Gated/FourierELU
RPP	SCNN + Residual Pathway Prior	RPP [8]	Gated
R-SCNN	Restricted SCNN	Restriction [5]	Gated
P-SCNN (ours)	Partial/Probabilistic Steerable CNN	Ours	Gated

Table 7.3: Overview of the main configurations of our base 2D CNN used for our `Double MNIST` experiments.

For our experiments, we use five primary configurations of the base structure. These configurations and

their specifics, especially concerning the attainment of partial equivariance (if present), are detailed in Table 7.3. Here, the RPP configuration [8] is derived by incorporating the SCNN configuration and appending residual conventional (non-equivariant) CNN connections. These connections span each individual block and conclude at the final pooling convolution, cumulatively resulting in six residual pathways for partial equivariance. Additionally, for the R-SCNN configuration. This decision aligns with the theoretical absence of symmetries on a broader scale in `Double MNIST`. Through empirical analysis, this setup was deemed optimal when applied to the final two convolution layers.

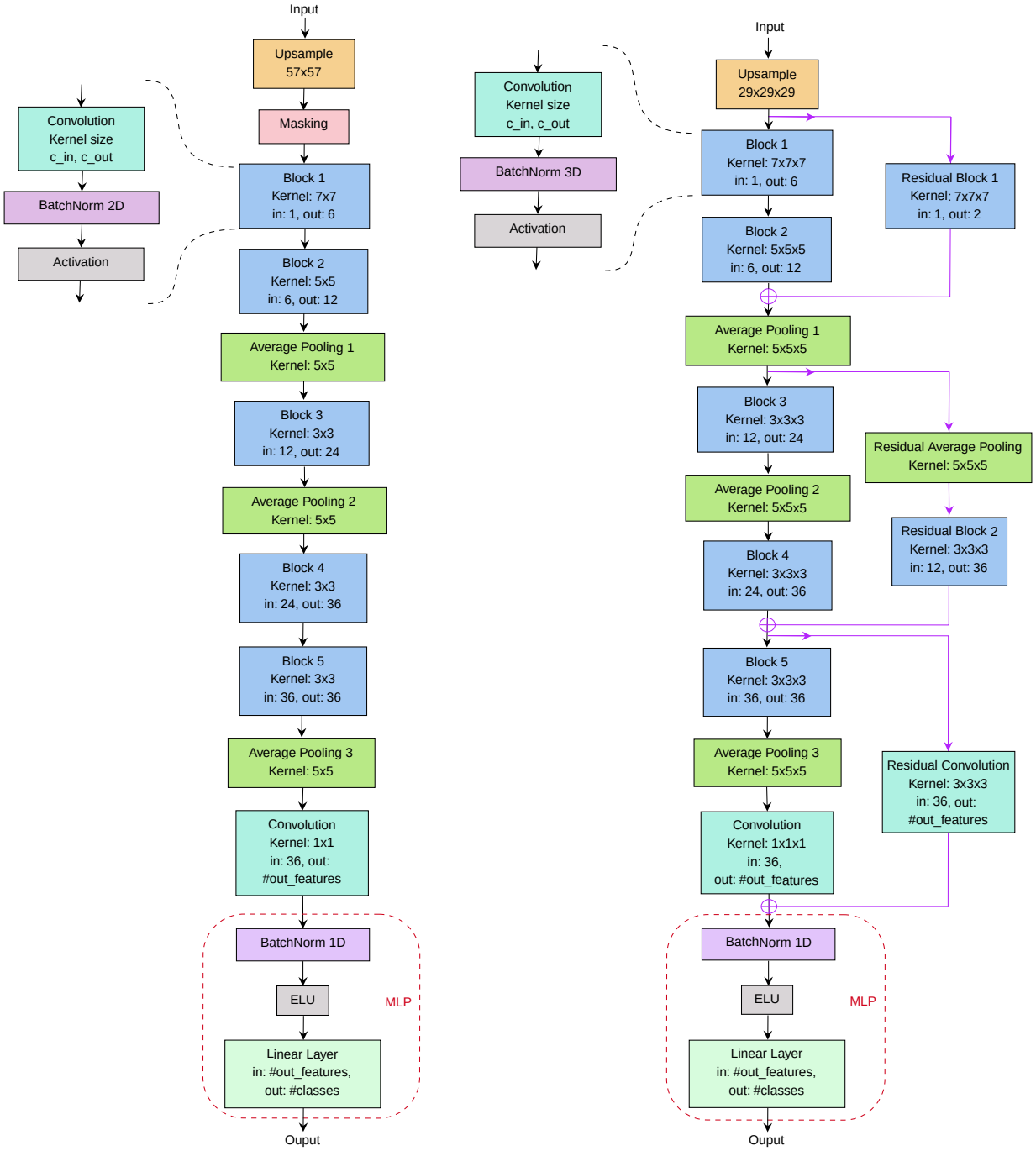
For all configurations, whether fully or partially equivariant, we conduct experiments with various compact groups H. Each group has its unique maximum level of bandlimiting for the intermediate irrep feature fields, as detailed in Table 7.4. In line with the MLP structure, we map to the trivial representation during the final 1×1 convolution to guarantee invariance by default. We refer to Appendix A for more details on the networks. This includes information on the bandlimiting of features for every layer in the equivariant network architectures. Lastly, in our approach, we allow each of the six convolutional layers to learn its distinct likelihood distribution, with a default bandlimit of $L = 2$.

Group	**Feature Field Bandlimiting**
$O(2)$	4
$SO(2)$	4
D_4	2
C_4	2
D_1	0
C_1	0
$SO(3)$	3
$O(3)$	2

Table 7.4: Maximum level of bandlimiting of the intermediate features for each group.

3D CNNs Finally, for `MedMNIST` we use a modified version of our base structure used for `Double MNIST` acting on $\mathbb{R}^3$ rather than $\mathbb{R}^2$. As the exact symmetries in these datasets are not known a-priori, we do not perform experiments using group restrictions. Otherwise, the base configurations are the same as in Table 7.3. A diagram of our modified base-structure can be found in Figure 7.3b. Here, we add three residual connections, spanning two blocks at a time. Due to these residual connections, the residual non-equivariant connections for the RPP configuration now also span two blocks at a time instead of one. Additionally, we set the width of each layer to half of the other models for the RPP configuration. This is required to prevent overfitting, since the regular CNN layers of the RPP configuration become too large in order to commute with the steerable layers.

Additionally, for our approach, rather than learning a separate likelihood distribution for each individual layer, the first block of each residual block –consisting of two sequential blocks and one residual block– has a fixed non-learnable degree of equivariance, whereas the second block and the residual block share a degree of equivariance. As a result, we end up with three likelihood distributions rather than six. For each (partially) equivariant configuration, we perform experiments using $H = SO(3)$ and $H = O(3)$, using the maximum feature field bandlimiting from Table 7.4. Similarly to the experiments on `Double MNIST`, we set the default banlimiting of the likelihood distributions to $L = 2$.



(a) Base 2D CNN configuration

(b) Base 3D CNN configuration

Figure 7.3: Diagrams of our 2D and 3D CNN base configurations.

Chapter 8

Results and Evaluation

This chapter covers our experiments and the evaluation of the results. First, in Section 8.1 we compare the quantitative performance of our probabilistic approach with the preliminary approach on the `Vectors` dataset. In Sections 8.2 and 8.3 we study the effect of various approaches of partially breaking equivariance in terms of quantitative results using the `Double MNIST` and `Medical MNIST` datasets respectively. To gain insight into the interpretability of our probabilistic approach, we inspect the learnt likelihood distributions across the datasets in Section 8.4. Subsequently, we study the impact of learning a shared equivariance in favour of an individual degree of equivariance in Section 8.5. Section 8.6 provides an analysis on the influence of the regularisation methods described in Sections 5.2.3 and 5.2.4 in terms of task-specific performance and interpretability. Since our main results are obtained using structurally invariant architectures and the Gated non-linearity, we experiment with non-invariant mapping and Fourier-based non-linearities in Section 8.7. Subsequently, to investigate the effect of our approach on data efficiency, which is an advantage of equivariant networks, Section 8.8 contains a data ablation study. Subsequently, Section 8.9 compares the generalisation capabilities of the various models. In Section 8.10 we recap and discuss our results. Finally, Section 8.11 discusses the considerations and future directions of this work.

All models undergo training five times, each with a different, pre-determined seed. Training is performed on an NVIDIA A100 40GB GPU using the Adam optimiser [13]. For our *probabilistic* PE-MLP and our P-SCNN we use the following objective function:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \alpha_{\text{align}}\mathcal{L}_{\text{align}} + \alpha_{\text{KL}}\mathcal{L}_{\text{KL}}$$

Where $\mathcal{L}_{\text{task}}$ is the task-specific loss function that is used for all other models, and α_{align} and α_{KL} are tunable weighting factors for the alignment loss and KL-divergence. We refer to Table 8.1 for more specific training details on the individual datasets, including these weighting factors.

Dataset	`Vectors`	`DoubleMNIST`	`OrganMNIST3D`	`SynapseMNIST3D`	`NoduleMNIST3D`
Learning rate	5×10^{-5}	5×10^{-5}	5×10^{-5}	1×10^{-5}	1×10^{-5}
batch size	1024	256	32	32	32
# of epochs	100	50	100	100	100
Loss function	Mean squared error (MSE)	Cross-entropy	Cross-entropy	Cross-entropy	Cross-entropy
Evaluated Epoch	Last	Best on validate	Last	Last	Last
$\boldsymbol{\alpha}_{\text{align}}$	5	5	5	5	5
$\boldsymbol{\alpha}_{\text{KL}}$	25	3	1	1	1

Table 8.1: Training details for the datasets employed in our experiments.

Furthermore, all RPP models are trained with a prior variance of $\sigma^2 = 10^5$ on the equivariant weights and $\sigma^2 = 10^3$ on the non-equivariant weights. Empirically, we found these settings to be the most optimal. These priors are comparable to those used for some experiments in [8].

8.1 Preliminary Approach compared to Probabilistic Approach

First, as proof of concept we compare the performance of the baselines to our *preliminary* and *probabilistic* approach on the simple `Vectors` dataset. For the *preliminary* approach, we include two configurations; one where the coefficients are initialised according to Eq. 5.1, and one with added noise to the matrices that are initialised as zeros, as detailed in Section 6.1. Table 8.2 provides an overview of the results. In the following two paragraphs, we discuss the performance on the angle and norm regression tasks separately.

Model	Non-linearity	Angle	Norm
MLP	ELU	0.169 (0.188)	0.058 (0.032)
E-MLP [5]	Gated	1.486 (0.044)	0.080 (0.002)
	FourierELU	0.643 (0.176)	0.004 (0.006)
Ours (Preliminary)	Gated	1.496 (0.010)	0.035 (0.018)
	FourierELU	0.264 (0.345)	0.019 (0.032)
Ours (Preliminary + Noise)	Gated	0.064 (0.022)	<u>0.033</u> (0.016)
	FourierELU	0.258 (0.316)	0.020 (0.040)
Ours (Probabilistic)	Gated	**<u>0.046</u>** (0.004)	0.052 (0.008)
	FourierELU	<u>0.128</u> (0.103)	**<u>0.001</u>** (0.001)

Table 8.2: Test MSE scores of our approaches and the baseline models on `Vectors` norm and angle regression tasks. **Bold** indicates the lowest MSE error for the specific task. For each non-linearity, <u>underline</u> indicates the lowest MSE error for this specific non-linearity on for the given task. Standard deviations over 5 runs are denoted in parentheses.

Angle We observe that, in terms of the non-$O(2)$-invariant angle regression task, the fully $O(2)$ invariant E-MLP is significantly outperformed by most of the other models, regardless of non-linearity. It is notable that the performance improves considerably when using the FourierELU non-linearity. This could partially be from the approximate equivariance of this non-linearity. Furthermore, we note that omitting the addition of noise to our *preliminary* approach results in similar performance compared to the E-MLP under the Gated non-linearity. However, the addition of noise results in a significant increase in performance. Otherwise, our *probabilistic* configuration and our *preliminary* configuration with added noise both achieve significantly lower errors compared to the E-MLP, while also outperforming the non-equivariant MLP. Finally, using the Gated non-linearity results in a lower error for these two configurations.

Upon further inspection, we find that, under the Gated-nonlinearity, employing the *preliminary* approach without added noise ensures that the c^{lJ} matrices for non-matching irreps ψ_l and ψ_J remain zero matrices throughout training. Consequently, the model exhibits invariance for the duration of the training process. To unravel the cause, it is vital to consider the difference between FourierELU and Gated non-linearities. The Gated non-linearity, due to its perfect equivariance, lacks any initial mappings between non-matching input and output irreps ψ_l and ψ_J. As a result, with the c^{lJ} matrices initialised to zero, the associated weights remain inactive and are, in effect, zero. Therefore, these weights do not influence the output, and since there are multiple sequential layers that are initialised in this manner, these weights cannot be updated through backpropagation.

This behaviour does not manifest with the *approximately* equivariant FourierELU non-linearity, which induces a degree of feature mixing across the irreps through the inverse Fourier transform to generate the sampled signal. Thus, weights that define mappings between non-matching irreps are utilised at initialisation, allowing backpropagation to update these weights.

The *probabilistic* approach, intriguingly, does not encounter this issue. This is a result of the additional weight sharing; within this framework, the Fourier coefficient of the trivial representation contributes

to mappings for both matching and non-matching irreps. This allows for the possibility of non-zero mappings between non-matching irreps after updates to this Fourier coefficient. Then, through normalisation, the other Fourier coefficients become slightly non-zero, freeing them up for backpropagation. Moreover, our method of normalising the likelihood distribution inadvertently introduces minor rounding errors, giving rise to slight, non-zero weights for non-trivial Fourier coefficients at initialisation.

Norm In the $O(2)$ invariant norm regression task, the performance gaps between the models are much smaller in general, suggesting that breaking equivariance is not as beneficial in this task. In fact, under the FourierELU non-linearity –which is comparable to the MLP's regular ELU– the E-MLP outperforms the regular MLP. Although we do note that using our preliminary approach with Gated non-linearity or our probabilistic approach with either non-linearity does result in an increase in terms of performance.

We provide additional results where the models are trained in both predicting the angles and the norms of the vectors simultaneously in Appendix B.

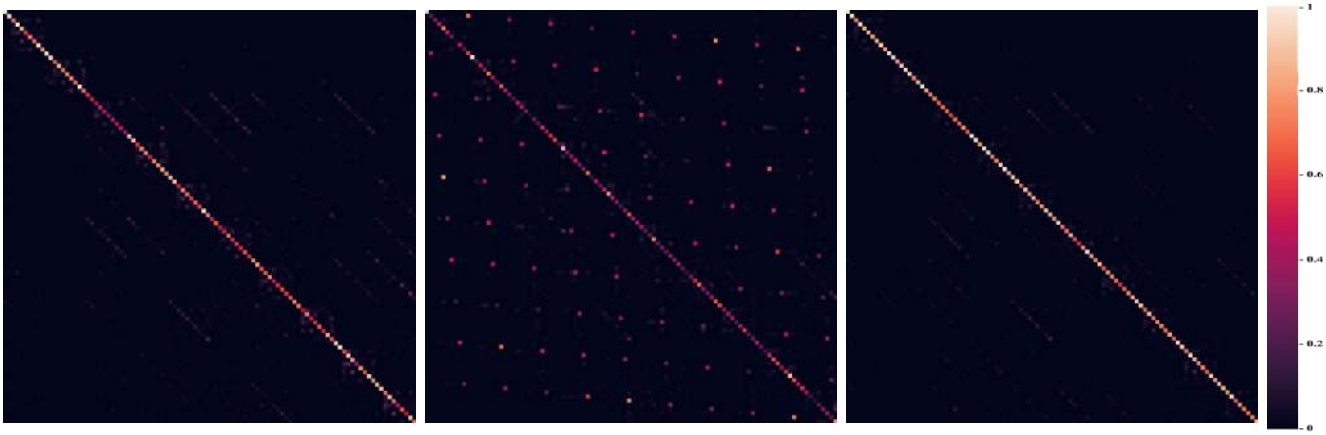
8.2 Benchmarks on Double-Digit Image Classification

The classification accuracies obtained in the test set of various augmentations of `Double MNIST` are shown in Table 8.3. Figure 8.1 shows samples of the confusion matrices obtained on `Double MNIST` with $O(2)$ symmetries achieved under the CNN, $O(2)$ SCNN, and $O(2)$ P-SCNN (ours) models.

For all symmetries, the non-equivariant CNN outperforms the fully H-invariant SCNNs, regardless of the choice of H. While the performance gap is relatively small for C_1 symmetries (i.e., no additional symmetries), this gap is considerably larger for the other symmetries. On the contrary, the use of any of the partial equivariant methods on top of SCNNs shows a performance improvement compared to the regular CNN for all but C_1 symmetries.

Network Group	Partial Equivariance	Symmetries C_1	C_4	$SO(2)$	D_1	D_4	$O(2)$
CNN	N/A	**0.962** (0.002)	0.868 (0.011)	0.807 (0.007)	0.919 (0.006)	0.711 (0.009)	0.649 (0.019)
C_4	None [5]	0.933 (0.002)	0.484 (0.008)	0.459 (0.007)	0.893 (0.005)	0.427 (0.008)	0.405 (0.008)
	Restriction [5]	0.954 (0.003)	0.911 (0.006)	0.877 (0.009)	0.928 (0.006)	0.827 (0.013)	0.776 (0.018)
	RPP [8]	0.937 (0.006)	0.901 (0.012)	0.867 (0.025)	0.899 (0.014)	0.821 (0.023)	0.772 (0.009)
	Ours	0.947 (0.006)	0.916 (0.005)	0.891 (0.006)	0.923 (0.007)	0.848 (0.007)	0.795 (0.011)
D_4	None [5]	0.895 (0.005)	0.439 (0.010)	0.396 (0.009)	0.473 (0.010)	0.431 (0.010)	0.394 (0.008)
	Restriction [5]	0.953 (0.004)	0.912 (0.007)	0.887 (0.003)	0.930 (0.007)	0.827 (0.007)	0.773 (0.009)
	RPP [8]	0.934 (0.007)	0.888 (0.014)	0.867 (0.014)	0.895 (0.007)	0.821 (0.020)	0.775 (0.013)
	Ours	0.949 (0.005)	**0.922** (0.007)	0.885 (0.012)	0.921 (0.008)	0.848 (0.011)	0.801 (0.009)
SO(2)	None [5]	0.936 (0.005)	0.485 (0.010)	0.474 (0.016)	0.890 (0.006)	0.430 (0.010)	0.403 (0.021)
	Restriction [5]	0.949 (0.002)	0.911 (0.010)	0.893 (0.009)	0.928 (0.003)	0.841 (0.012)	0.796 (0.011)
	RPP [8]	0.935 (0.008)	0.890 (0.005)	0.870 (0.011)	0.899 (0.009)	0.821 (0.022)	0.779 (0.021)
	Ours	0.953 (0.004)	**0.922** (0.005)	**0.901** (0.005)	**0.932** (0.005)	**0.863** (0.009)	**0.823** (0.005)
O(2)	None [5]	0.881 (0.005)	0.415 (0.008)	0.391 (0.012)	0.461 (0.012)	0.424 (0.009)	0.399 (0.014)
	Restriction [5]	0.953 (0.005)	0.914 (0.005)	0.894 (0.005)	0.928 (0.005)	0.845 (0.011)	0.799 (0.008)
	RPP [8]	0.931 (0.005)	0.891 (0.003)	0.861 (0.013)	0.891 (0.004)	0.824 (0.009)	0.772 (0.019)
	Ours	0.958 (0.003)	0.919 (0.006)	0.894 (0.004)	0.927 (0.004)	0.859 (0.011)	0.819 (0.010)

Table 8.3: Test accuracies on various `Double MNIST` data symmetries. The first column indicates the equivariance group, where CNN is the standard non-equivariant CNN. The second column indicates the method used to break the equivariance, along with a citation if applicable. For each symmetry, the highest accuracy for each network group is underlined. The highest overall accuracy for each symmetry is indicated in **bold**. Standard deviations over 5 runs are denoted in parentheses.



(a) CNN **(b)** O(2) SCNN **(c)** O(2) P-SCNN (ours)

Figure 8.1: Confusion matrices obtained on `Double MNIST` with $O(2)$ symmetries using a regular CNN, an $O(2)$ equivariant SCNN and our $O(2)$ P-SCNN.

The confusion matrices in Figure 8.1 provide more insight into the cause of this behaviour. Here, Figure 8.1a shows that the CNN manages to correctly classify most samples. The lines parallel to the diagonal and the clumps around the diagonal indicate errors by missclassifications of one of the two digits, e.g. confusing 34 for 84 or 34 for 37. On the other hand, the pattern in Figure 8.1b shows that the $O(2)$ SCNN mainly confuses digit reversals, e.g. confusing 37 for 73. In fact, for each double-digit number the amount of digit-reversals is comparable to the number of accurate classifications, illustrating that the model is not able to distinguish between digit reversals. Finally, Figure 8.1c reveals that our $O(2)$ P-SCNN manages to eliminate these confusions, while also clearing up the missclassifications of one of the two digits compared to the CNN.

Our method, especially when partially equivariant to $SO(2)$, consistently achieves superior performance across each data symmetry compared to the restricted SCNN and RPP method. Zooming in on specific groups, we note that for the smaller C_4 and D_4 groups, a manual restriction yields slightly higher performance compared to our method for C_1 and D_1 symmetries.

Furthermore, it is interesting to note that choosing a group that is too large results in a higher drop in performance for the fully equivariant SCNN than for the partially equivariant approaches. For instance, taking $H = O(2)$ rather than $SO(2)$ results in a 48.2% drop in performance for D_1 symmetries and 17.5% for $SO(2)$ symmetries, whereas our approach reduces these differences to 0.5% and 0.8%, respectively.

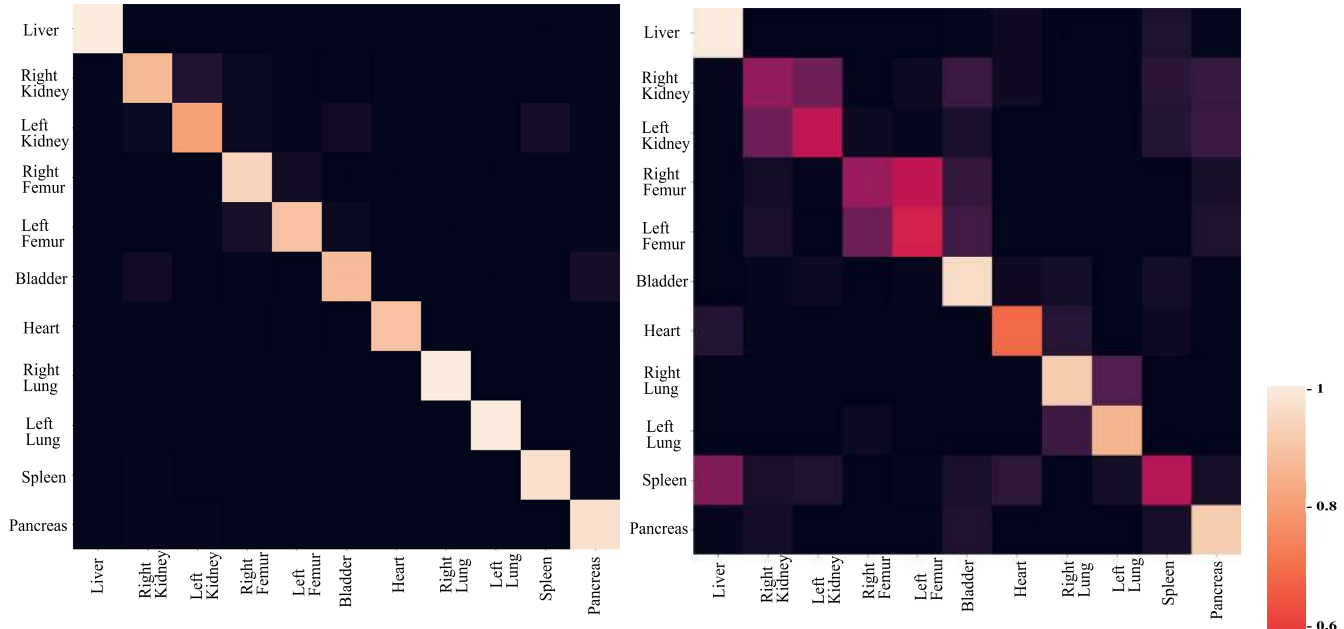
8.3 Benchmarks on Biomedical Image Classification

We present the test accuracies for the `MedMNIST` datasets in Table 8.4 using structurally invariant models when applicable. The confusion matrices for `OrganMNIST3D` using the $O(3)$ models –when applicable– are shown in Figure 8.2.

Here, we observe that the three datasets behave differently under full invariance compared to a regular CNN. For `NoduleMNIST3D`, both CNN and fully invariant SCNN models provide comparable performance. In fact, CNNs, SCNNs, and our P-SCNN models all achieve nearly identical performance, regardless of the chosen group. However, the $SO(3)$ RPP has a notable drop in performance compared to the $O(3)$ RPP, and is slightly outperformed overall.

Network Group	Partial Equivariance	Nodule	Synapse	Organ
CNN	N/A	**0.873** (0.005)	0.716 (0.008)	0.920 (0.003)
$SO(3)$	None [5]	**0.873** (0.002)	0.738 (0.009)	0.607 (0.006)
	RPP [8]	0.801 (0.003)	0.695 (0.037)	0.936 (0.002)
	Ours	0.871 (0.001)	**0.770** (0.030)	0.902 (0.006)
$O(3)$	None [5]	0.868 (0.009)	0.743 (0.004)	0.592 (0.008)
	RPP [8]	0.810 (0.013)	0.722 (0.023)	**0.940** (0.006)
	Ours	**0.873** (0.008)	0.769 (0.005)	0.905 (0.004)

Table 8.4: Test accuracies on `MedMNIST` datasets. For each subset, the highest accuracy for each network group is underlined. The highest overall accuracy for each type of symmetry is indicated in **bold**. Standard deviations over 5 runs are denoted in parentheses.



(a) CNN

(b) $O(3)$ SCNN

(c) $O(3)$ P-SCNN (ours)

(d) $O(3)$ RPP

Figure 8.2: Confusion matrices obtained on OrganMNIST3D symmetries using a regular CNN, an $O(3)$ SCNN, $O(3)$ RPP and our $O(3)$ P-SCNN.

`SynapseMNIST3D` sees an improvement of around 3 percentage points for the SCNNs compared to the CNN, and an additional 2 percentage points for our P-SCNNs. Similarly to `NoduleMNIST3D`, the RPP models are outperformed by all other models. These results indicate that `SynapseMNIST3D` is fairly symmetric, while `NoduleMNIST3D` is a highly symmetric dataset, showing minimal benefit from breaking equivariance.

Conversely, the results on `OrganMNIST3D` show that the non-equivariant CNN strongly outperforms the fully equivariant SCNNs by around 30 percentage points. Our P-SCNN models improve upon the SCNN models, performing only 2 percentage points worse than the CNN. Finally, the RPP models manage to achieve the highest performance, outperforming the CNN by 2 percentage points. As a result, it seems that unlike the other datasets, `OrganMNIST3D` contains important non-symmetric features, resulting in a significant benefit from partial invariance compared to full invariance.

The confusion matrices in Figure 8.2 provide some insight into this behaviour on `OrganMNIST3D`. Here, the CNN's confusion matrix in Figure 8.2a shows some, albeit minor, increased confusion for the left and right versions of the kidney and femur. These confusions are significantly worsened for the $O(3)$ SCNN in Figure 8.2b, showing poor performance for any of the organs with left and right versions, in addition to the confusion of other organs such as the liver and the spleen. Our $O(3)$ P-SCNN model shows a significant improvement over the SCNN in Figure 8.2c, eliminating most missclassifications, although still showing some additional density for the left and right versions for the femur and kidney compared to the CNN. Finally, the $O(3)$ RPP from Figure 8.2d manages to show an additional slight improvement for these specific organs.

8.4 Inspecting the Learnt Likelihood Distributions

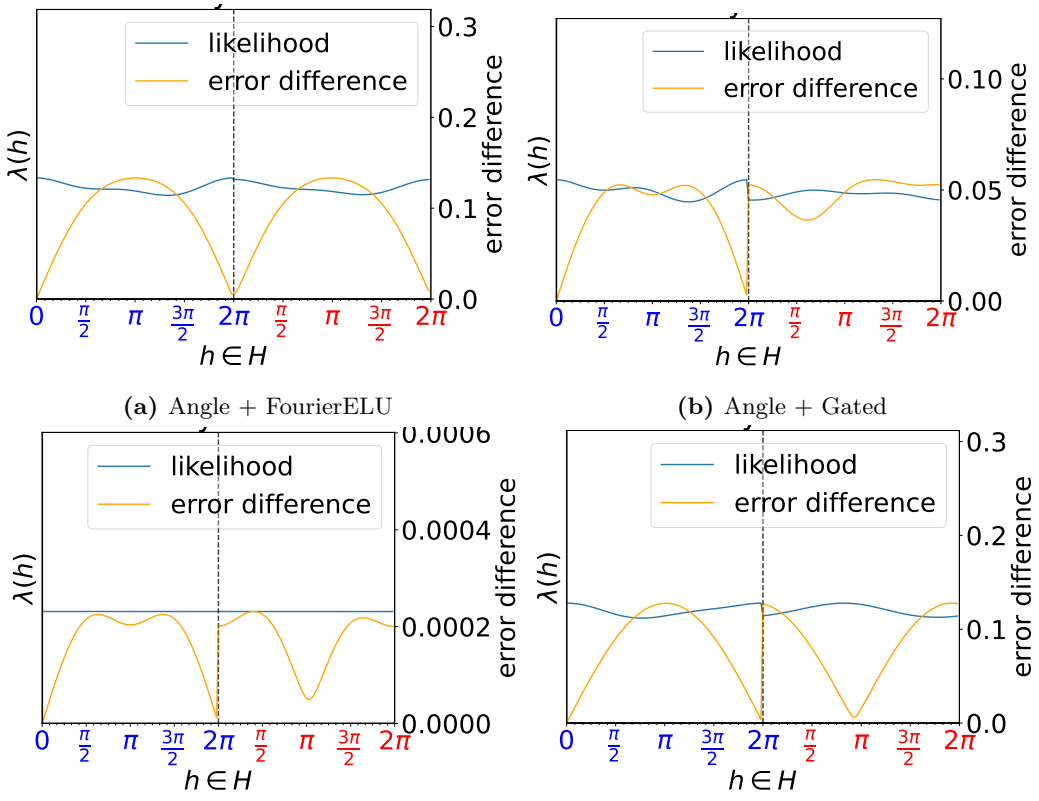
In the previous sections, we discussed the task-specific performance of the various models under the various dataset. This section focusses on the interpretability of our models by presenting and evaluating the learnt likelihood distributions for our datasets. To verify the validity of the learnt likelihood distributions, we include measured equivariance errors across the group. However, instead of simply computing the error from Eq. 3.27, we compensate for the differences in feature magnitude by normalising the relative error vectors based on the magnitude of the original output feature. Additionally, to obtain scalar values we calculate the ℓ^2 norm over the feature vectors. Thus, for each layer l and a batch of samples B, we compute the following:

$$\epsilon_l(h) = \frac{1}{|B|} \sum_{b \in B} \frac{||h \cdot l(b) - l(h \cdot b)||_2}{||l(b)||_2}, \quad \forall h \in H.$$

Since the errors can be non-zero for the fully invariant models due to interpolation artefacts or other sources of inherent partial equivariance, rather than reporting the equivariance errors directly we report the difference in equivariance errors between the fully invariant SCNNs/E-MLPs and our partially equivariant P-SCNNs/PE-MLPs. As a result, the reported errors indicate how our approach differs from the invariant setting. Furthermore, all reported errors are calculated using one batch of data.

In the ensuing discussion, we examine the learnt likelihoods across three datasets. We often refer to specific elements of $O(2)$ to distinguish between different types of transformations. To simplify the notation, we use colour-coding: blue signifies a basic rotation, such as a rotation by π denoted as π, while red indicates a composite transformation involving a horizontal reflection followed by a rotation, exemplified by a horizontal reflection and a $\frac{\pi}{2}$ rotation denoted as $\frac{\pi}{2}$. Therefore, this section is best viewed in colour.

`Vectors` Figure 8.3 displays the learnt likelihood distributions and equivariance errors for the second layer of our *probabilistic* $O(2)$ PE-MLP utilising Gated and FourierELU non-linearities, the quantitative results of which are discussed in Section 8.1. These metrics are defined over the group $O(2)$, which is divided into two domains with a dotted line in our visualisations: the first for rotation elements



(a) Angle + FourierELU

(b) Angle + Gated

(c) Norm + FourierELU

(d) Norm + Gated

Figure 8.3: Learnt likelihood λ and error difference for the second layer in our *probabilistic* PE-MLP trained on angle or norm regression, with Gated and FourierELU non-linearities. The error difference is calculated against an E-MLP. The dotted line marks the transition between the non-reflective and $O(2)$ reflection domain. Note that the scale of the equivariance error varies between the plots.

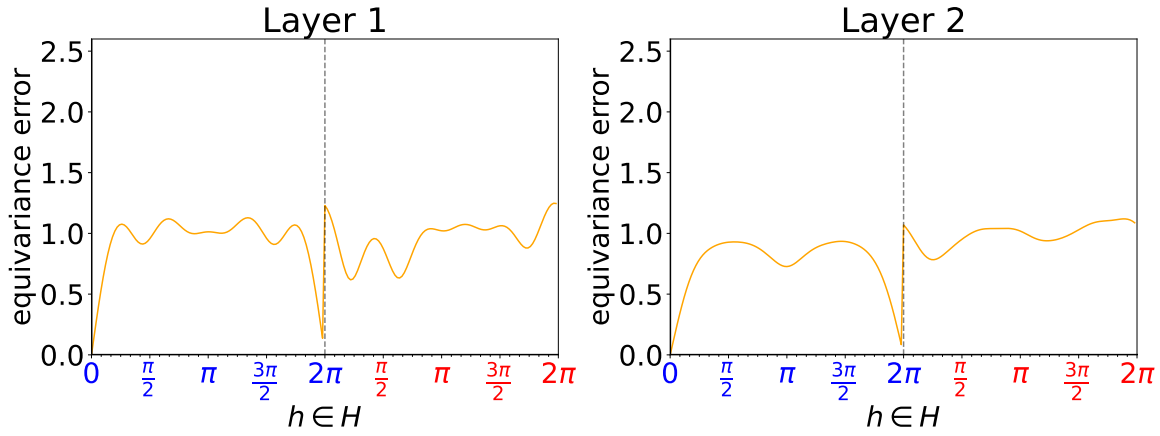
and the second for elements comprising a horizontal reflection followed by a rotation. It should be noted that the scales differ between subfigures due to substantial variations in equivariance errors. In Appendix B, we present the likelihood distributions for all three layers.

An analysis of Figure 8.3 reveals that a decline in likelihood is typically correlated with an increase in the equivariance error. Additionally, the equivariance error exhibits frequencies comparable to those observed in the likelihood distribution. Likewise, a more uniform likelihood distribution leads to reduced equivariance errors, as highlighted in Figure 8.3c.

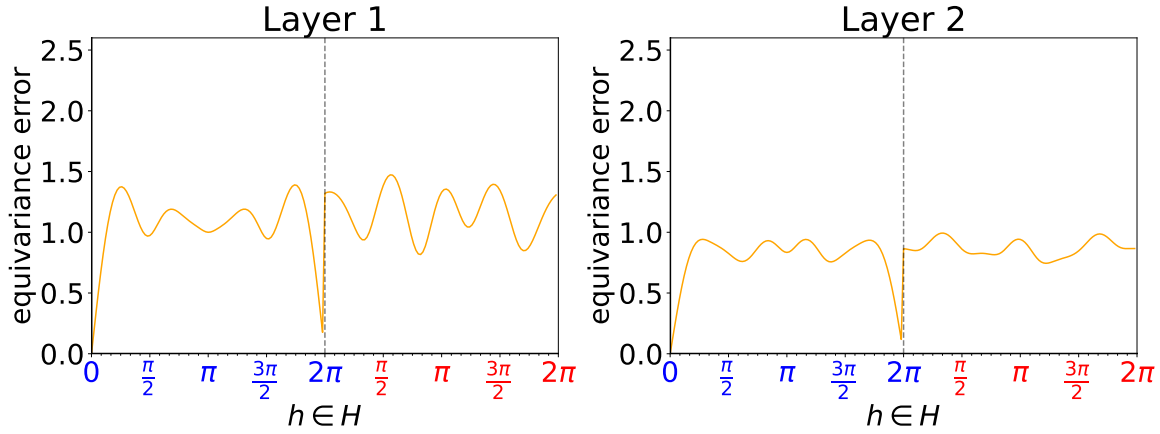
In the task of non-invariant angle prediction, both Gated and FourierELU non-linearities maintain a similar degree of partial equivariance in the rotational domain. Both break equivariance for all but the identity elements, with the lowest likelihood and the highest error occurring for rotations close to an angle of π, which is to be expected, as rotating a vector by π yields a vector in the opposite direction. However, the Gated non-linearity exhibits reduced equivariance in the reflective domain when compared to its FourierELU counterpart.

For the invariant norm prediction task, the FourierELU non-linearity features an almost uniform likelihood distribution, with low equivariance errors as a result. Conversely, the Gated non-linearity has a less uniform distribution, displaying a signal with a frequency of one and consequently higher equivariance errors. Further results relevant to this behaviour are presented in Section 8.5.

Finally, the equivariance errors in the *probabilistic* approach manifest low-frequency signals, whereas those in the *preliminary* approach, as seen in Figure 8.4, contain high-frequency signals across both regression tasks in the first layer and in the second layer for the norm regression task. Furthermore, ignoring high-frequency fluctuations, the equivariance errors also seem to remain relatively consistent throughout the group H.



(a) Preliminary approach on angle regression task.



(b) Preliminary approach on norm regression task.

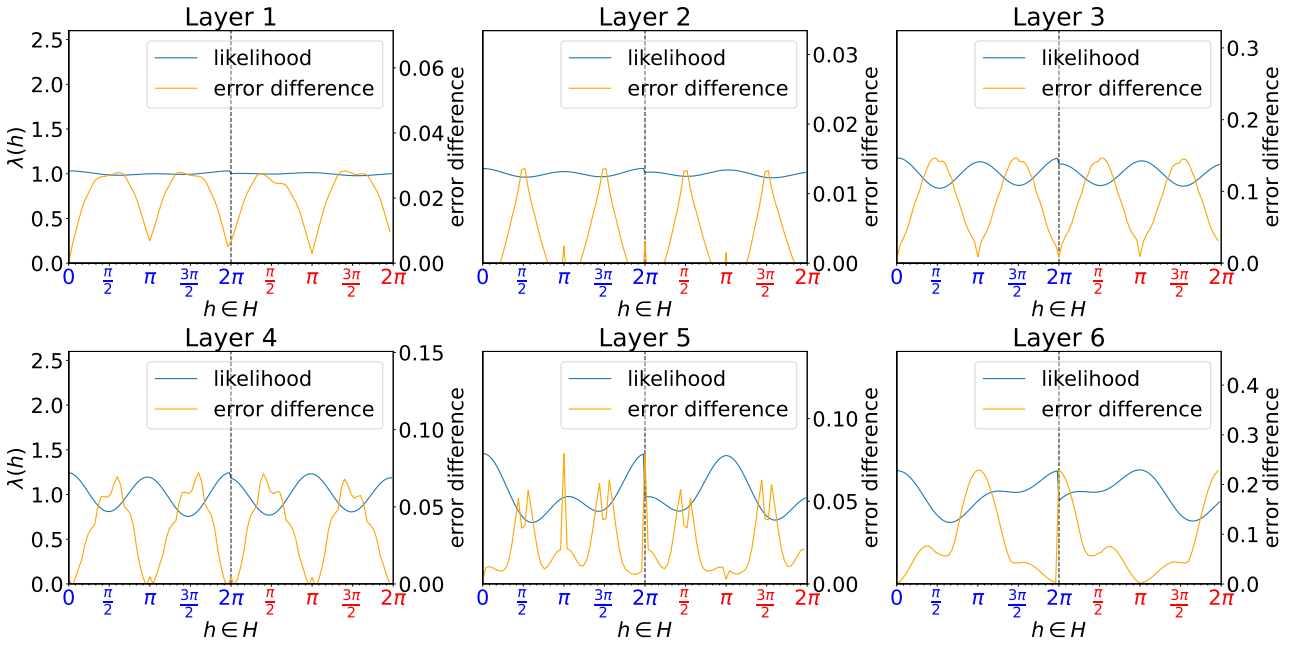
Figure 8.4: Measured equivariance error of the first two layers in our *preliminary* PE-MLP with added noise trained on angle or norm regression with Gated non-linearity. The error difference is calculated against an E-MLP. The dotted line marks the transition between the non-reflective and $O(2)$ reflection domain. Note that the scale of the equivariance error varies between the plots.

These results suggest that the probabilistic approach results in a more consistent and representative breaking of equivariance in comparison to the preliminary approach, with easy interpretability as an additional advantage. Due to these results, we only consider the probabilistic approach in future sections.

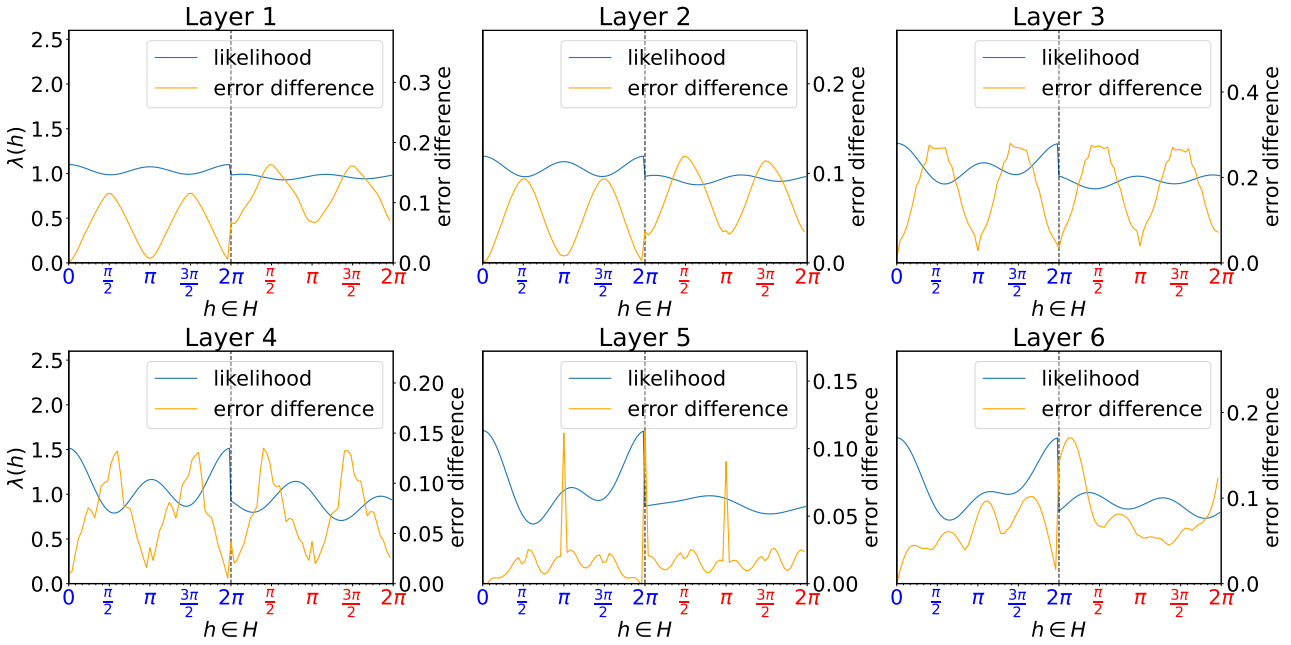
`Double MNIST` For our `Double MNIST` dataset we compare the learnt likelihood distributions and resulting equivariance errors for each layer of our $O(2)$ P-SCNN, trained on `Double MNIST` with $O(2)$ or C_1 symmetries. We present the results in Figure 8.5. Note that earlier layers correspond to smaller-scale features and later layers correspond to larger-scale features.

Like the results from `Vectors`, we observe that a lower likelihood results in a higher equivariance error. However, while the frequencies of the errors seem to match the frequencies of the likelihoods in the first three layers, the last three layers exhibit high frequencies in their equivariance errors. This becomes especially apparent in the fifth layer $h = \pi$, indicating that this might be the result of interpolation artefacts.

In Figure 8.5a, the model trained on $O(2)$ symmetries remains relatively equivariant in its first two layers. However, in layers three and four, it subsequently loses equivariance for $\frac{\pi}{2}$ and $\frac{3\pi}{2}$ rotations in both the reflective and non-reflective domains, corresponding to instances of extrinsic equivariance. Subsequently, layers 5 and 6 show a decrease in equivariance with respect to $h = \pi$ and $h = 0$. At a large scale, these two symmetries both result in a digit-reversal, and are thus instances of incorrect equivariance, with the former also turning the individual digits upside down. *Interestingly, the model*



(a) $O(2)$ symmetries

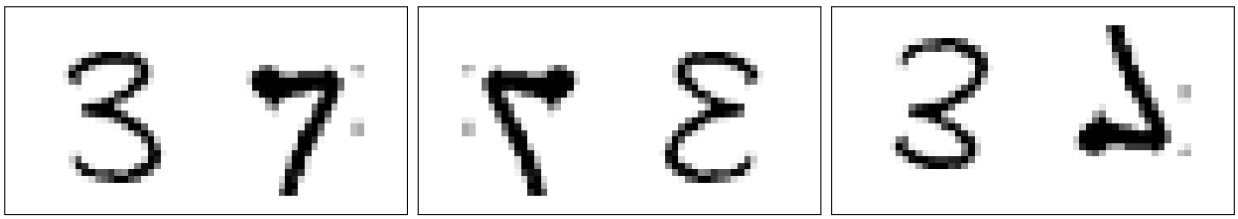


(b) C_1 symmetries (i.e. no symmetries)

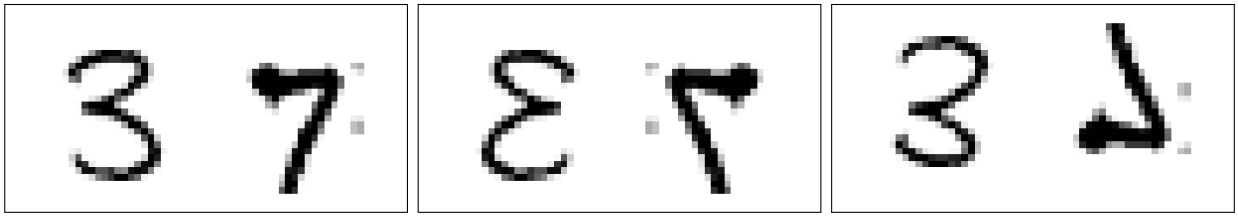
Figure 8.5: Learnt equivariance likelihood λ and error difference for layers 1 through 6 in an $O(2)$ P-SCNN trained on Double MNIST, with and without $O(2)$ symmetries. Error difference is calculated against an $O(2)$ SCNN. Dotted line marks the $O(2)$ reflection domain transition. Note that the scale of the equivariance error varies between the plots.

maintains a high degree of equivariance with respect to $h = \pi$, while applying this symmetry at a large scale corresponds to a rotation by π on the individual digits, as the digits are swapped twice. Therefore, this is indeed a case of correct equivariance, even though we did not initially consider this symmetry. See Figure 8.6 for a visualisation.

Comparatively, the model trained on C_1 symmetries in Figure 8.5b breaks equivariance much more rapidly throughout the network compared to the previous model, with higher equivariance errors as a result. Most notable, reflection equivariance is significantly reduced in layer two rather than layer five, and the final two layers do not maintain equivariance with respect to $h = \pi$. This is to be expected, as there are no rotation or reflection augmentations under C_1 symmetries, and therefore



(a) Global augmentations.



(b) Local augmentations.

Figure 8.6: Illustration comparing global and local (digit-wise) augmentations on images of double-digit numbers. Each row presents a case, showing, from left to right: the original image, a horizontal reflection denoted by $h = 0$, and a rotation by $h = \pi$. It should be noted that although the images in the middle column differ – with the image in the bottom row denoting a different number—- the images in the last column are identical.

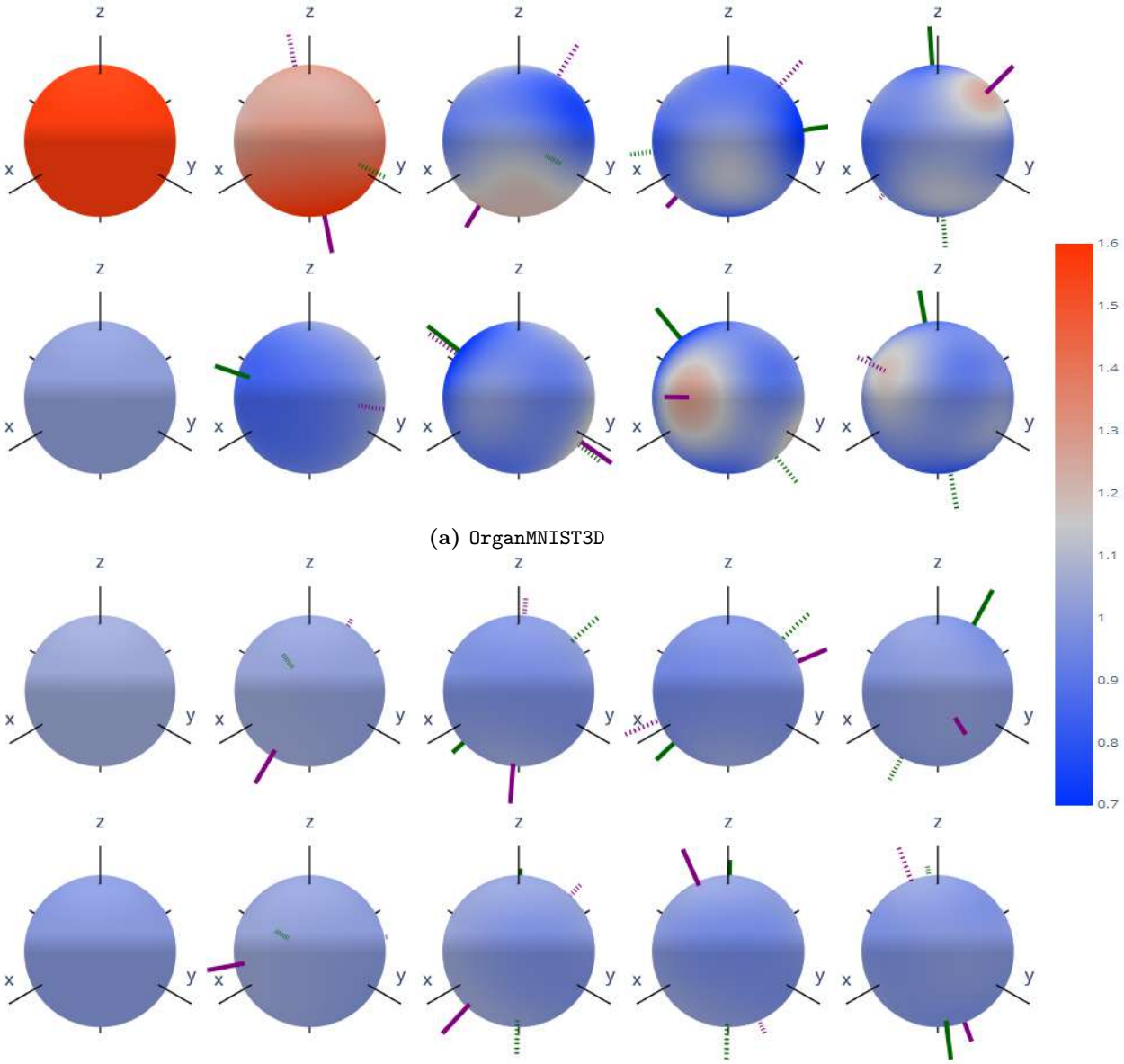
these transformations are cases of extrinsic equivariance.

`MedMNIST` While 2D rotations can be described using a single angle, making them easy to visualise, 3D rotations require at least three parameters and can be represented in multiple ways —such as Euler angles, quaternions, or rotation matrices— complicating their intuitive understanding and visualisation. However, similar to how elements of 2D rotations can be interpreted as elements on a circle, or a 1-sphere, the quaternion representation of 3D rotations can be interpreted as elements on a 3-sphere, which is a sphere living in 4D space. As this is still difficult to visualise, we aim for a different approach.

For our parameterisation, we choose an axis-angle representation, where each rotation is described by a 3D unit vector, or Euler vector, and a single rotation between 0 and π around this vector. Since the collection of all 3D unit vectors naturally forms a 2-sphere, the addition of a rotation vector makes this representation similar to a 3-ball. We generate our 2-sphere slices by taking increments of $\frac{\pi}{2}$ over the rotation angle. As a rotation over θ over an Euler vector is equal to a rotation of $-\theta$ over the opposite vector, we only need to parameterise the rotation angle up to π. Figure 8.7 shows the resulting likelihood distributions of the first residual block of our $O(3)$ P-SCNN trained on `OrganMNIST3D` and `SynapseMNIST3D`.

Here, it becomes apparent that the model trained on `OrganMNIST3D` shows a non-uniform degree of equivariance in the non-reflective and reflective domains. Specifically, the model breaks equivariance considerably for rotations of $\frac{\pi}{2}$ or higher for most Euler vectors in the non-reflective domain, and for any rotation in the reflective domain. However, there are exceptions where certain rotation axes result in significantly higher likelihoods for the same degree of rotation. For example, a rotation by π along the vector located between the positive y and z axes yields a higher likelihood. Likewise, a reflection followed by a $\frac{3\pi}{4}$ rotation along an axis between the positive x and z axes generates a markedly elevated likelihood compared to a standalone reflection.

On the contrary, the likelihood for `SynapseMNIST3D` is almost entirely uniform over the entire group, containing only a small decrease for non-identity elements. *These results suggest that the task defined in `OrganMNIST3D` is indeed non-symmetric, whereas the task in `SynapseMNIST3D` is mostly symmetric.*



(a) `OrganMNIST3D`

(b) `SynapseMNIST3D`

Figure 8.7: Heatmap of the learnt $O(3)$ likelihood distribution from the second residual block for `OrganMNIST3D` and `SynapseMNIST3D` visualised as 2-sphere slices of the 3-ball using the axis-angle representation. A higher value corresponds to a higher degree of equivariance. For each dataset, the first row contains the non-reflective domain and the second row the reflective domain. The columns correspond to rotations over the Euler vectors by increments of $\frac{\pi}{4}$ up to π. The first column indicates a rotation of 0 radians and is thus uniform. The solid green and purple lines point in the directions of the respective lowest and highest likelihood for each sphere, with dotted lines denoting their opposite directions to improve the visualisation in case of occlusions. Each 2-sphere is sampled at 10,000 points.

8.5 Shared Equivariance compared to Individual Equivariance

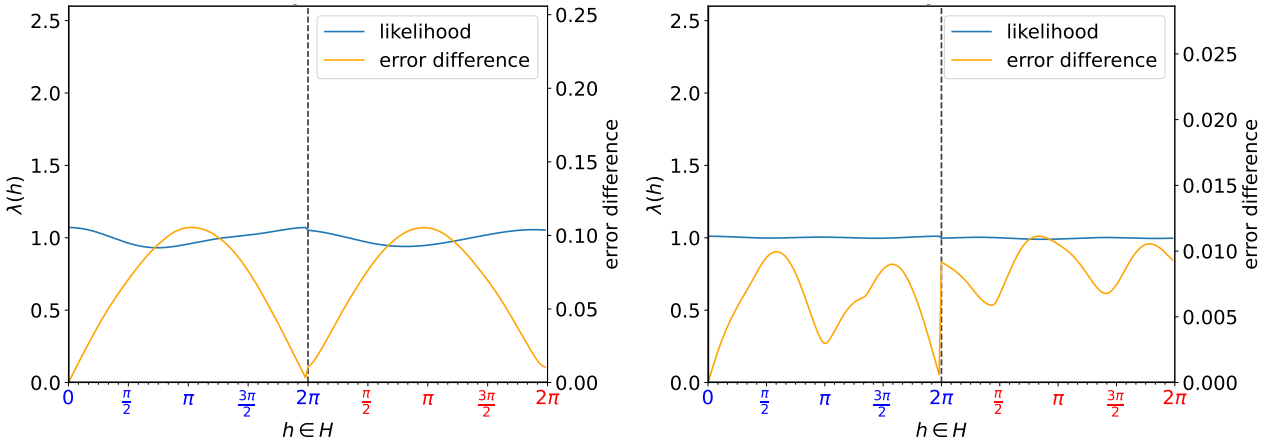
In the previous experiments, each layer, or block of layers, has learnt its own likelihood distribution. However, as discussed in Section 6.2, since MLPs lack the concept that each layer models features of its own scale, it might be beneficial to share the degree of equivariance between MLP layers. To explore this, we compare PE-MLPs parameterised by a single shared equivariance with PE-MLPs with layer-wise likelihoods from Section 8.1, both using the Gated non-linearity. Table 8.5 contains the MSE regression errors for both configurations. Figure 8.8 shows the likelihoods along with the equivariance errors for the models parameterised by a single likelihood distribution.

In terms of regression loss, both parameterisations achieve comparable performance across both tasks, with an individual parameterisation outperforming the shared parameterisation slightly on the norm regression task. However, comparing Figure 8.8 to Figure 8.3b and 8.3d reveals that the likelihoods are more uniform under the layer-wise parameterisation, particularly for the norm regression task, which more closely represents the invariant nature of this task. Furthermore, for the angle regression task, the likelihood distribution only appears to exhibit a frequency one signal, whereas the individual parameterisation in Figure 8.3b also contains higher frequencies. This more closely follows our expectations regarding the symmetries of the tasks in this dataset described in Section 7.1.1.

Equivariance	Angle	Norm
Shared	**0.044** (0.003)	0.064 (0.016)
Individual	0.0046 (0.005)	**0.052** (0.008)

Table 8.5: Comparison of MSE test scores between models parameterised by shared and individual degrees of equivariance on `Vectors`. For both regression tasks, the lowest error is **bold**. Standard deviations over 5 runs are denoted in parentheses.

These findings suggest that sharing a single degree of equivariance between subsequent linear layers (MLP) is preferable to learning distinct degrees for each. This approach yields more representative likelihood distributions.



(a) Angle **(b)** Norm

Figure 8.8: Resulting likelihood distributions and equivariance errors of our PE-MLP using a shared degree of equivariance across the layers. Note that the scale of the equivariance error varies between the plots.

8.6 Investigating the Effect of Regularisation Methods

In Section 5.2.3, we proposed two error terms that serve as regularisation components for the training objective function. One is the alignment loss, responsible for aligning the likelihood properly with the identity element, while the other is the KL-divergence, aimed at discouraging the model from inaccurately recovering lost equivariance in layers that follow. In the previous experiments, both regularisation terms were enabled. Therefore, we analyse the effect of separately disabling both these regularisation methods on the quantitative and qualitative results in Sections 8.6.1 and 8.6.2.

Additionally, in Section 5.2.4, we proposed to bandlimit the likelihood distribution as further method of regularisation. As described in Section 7.2, all previous SCNN experiments used a bandlimit of $L = 2$. Section 8.6.3 explores the effect of higher and lower levels of bandlimiting on the performance and degrees of equivariance.

8.6.1 Alignment Loss

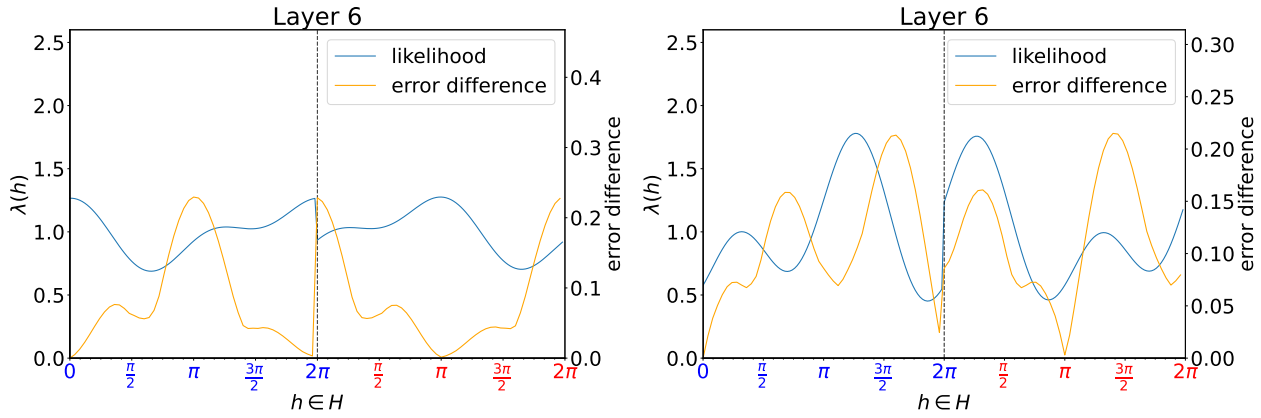
Quantitative results for both `Vectors` and `Double MNIST` with $O(2)$ symmetries, comparing scenarios with alignment loss either enabled or disabled, are presented in Tables 8.6 and 8.7, respectively. Furthermore, Figure 8.9 provides a comparison of the resulting likelihoods and equivariance errors for the sixth layer of our $O(2)$ P-SCNN.

Alignment Loss	Angle	Norm
✓	**0.044** (0.002)	**0.052** (0.008)
✗	0.064 (0.028)	0.062 (0.018)

Table 8.6: Comparing the effect of alignment loss on our PE-MLP in terms of MSE regression loss on `Vectors`. For both regression tasks, the lowest error is **bold**. Standard deviations over 5 runs are denoted in parentheses.

Alignment Loss	Accuracy
✓	**0.819** (0.010)
✗	0.805 (0.015)

Table 8.7: Comparing the effect of alignment loss on our $O(2)$ P-SCNN in terms of classification accuracy on `Double MNIST` with $O(2)$ symmetries. The higest accuracy is **bold**. Standard deviations over 5 runs are denoted in parentheses.



(a) With alignment loss

(b) Without alignment loss

Figure 8.9: Comparison of the likelihood learnt in layer 6 of an $O(2)$ P-SCNN on `Double MNIST` with $O(2)$ symmetries. Note that the scale of the equivariance error varies between the plots.

For both datasets, Tables 8.6 and 8.7 show a moderate increase in performance from the addition of alignment loss in the training objective. More importantly, Figure 8.9 reveals that the likelihood distribution is not properly aligned when the alignment loss from the loss function is omitted. Here, the group element of around $h = \frac{5\pi}{4}$ yields the maximum likelihood rather than the required identity element $h = 0 = e$. As a result, the measured equivariance error does not match the likelihood distribution, with some high equivariance errors occurring for group elements ascribed with high likelihoods, such as $h = \frac{\pi}{4}$, and vice-versa. Upon closer inspection, it appears that the entire likelihood distribution is indeed shifted by $\frac{5\pi}{4}$ across the entire distribution, since performing a shift of $-\frac{5\pi}{4} = \frac{3\pi}{4}$ aligns the peak at $h = \frac{5\pi}{4}$ to the identity $h = 0 = e$, in addition to aligning the peak at $h = \frac{1\pi}{4}$ to $h = \pi$.

As such, excluding alignment loss from the objective leads to misaligned and thus uninterpretable likelihood distributions, indicating that is critical for interpretability of the learnt degree of equivariance.

8.6.2 KL-Divergence

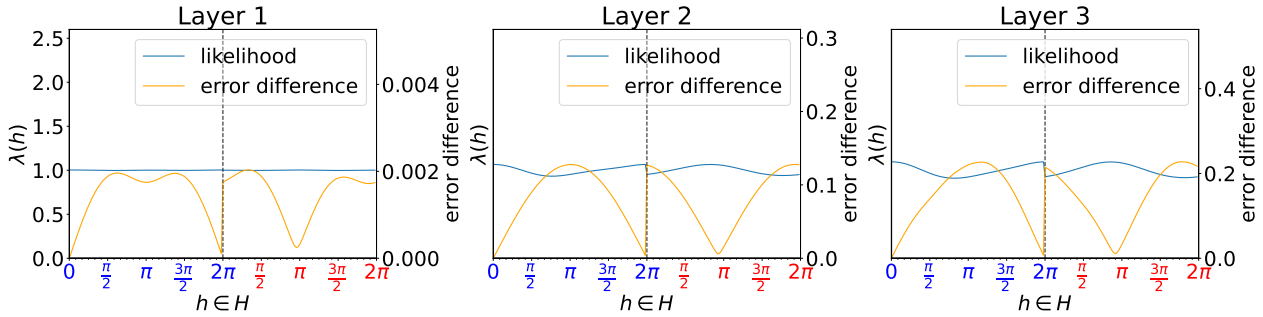
Similarly to the previous section, we present our quantitative results for `Vectors` and `Double MNIST` with $O(2)$ symmetries in Tables 8.8 and 8.9 respectively. To inspect the effect of KL-divergence on subsequent layers, we present the resulting likelihood distributions for various layers in our PE-MLP trained on norm regression and $O(2)$ PS-SCNN in Figures 8.10 and 8.11 respectively.

KL-divergence	Angle	Norm
✓	**0.044** (0.002)	0.055 (0.011)
✗	0.061 (0.001)	**0.003** (0.002)

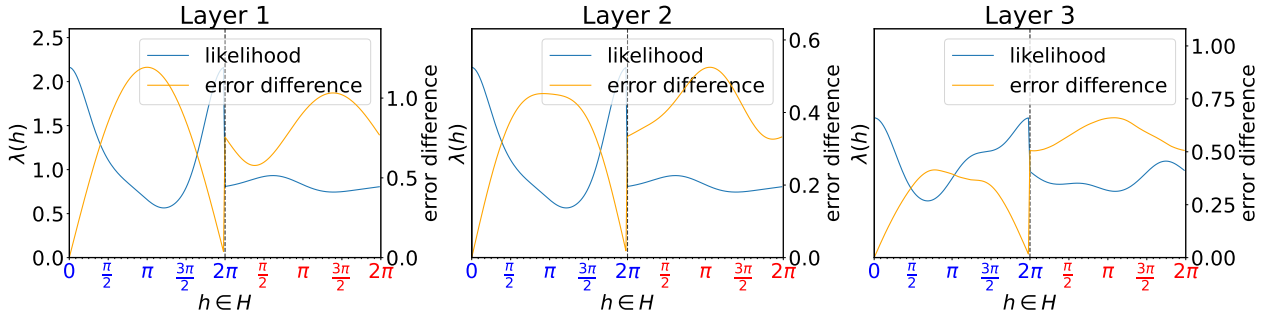
Table 8.8: Comparing the effect of KL-divergence on our PE-MLP in terms of MSE regression loss on `Vectors`.

KL-divergence	Accuracy
✓	**0.819** (0.010)
✗	0.797 (0.015)

Table 8.9: Comparing the effect of KL-divergence on our $O(2)$ P-SCNN in terms of classification accuracy on `Double MNIST` with $O(2)$ symmetries.



(a) With KL regularisation



(b) Without KL regularisation

Figure 8.10: PE-MLP layer-wise likelihoods with and without KL-divergence on `Vector`'s norm regression task. Note that the scale of the equivariance error varies between the plots.

From Tables 8.8 and 8.9, we observe that both `Vector`'s angle regression and `Double MNIST` number classification tasks benefit from incorporating KL-divergence in the objective function. However, `Vector`'s norm regression task sees a significant performance reduction from the KL-divergence regularisation. Figure 8.10 shows that removing the KL-divergence from the loss function results in significantly less uniform likelihood distributions. In fact, each layer essentially entirely loses equivariance with respect to all non-identity group elements. However, as the regression loss has improved and the norm

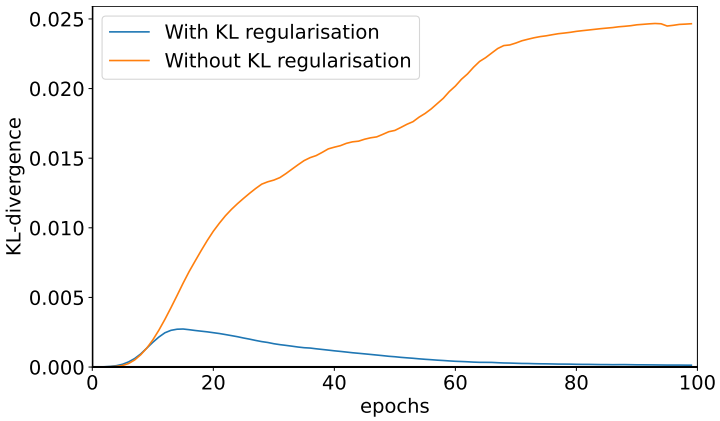
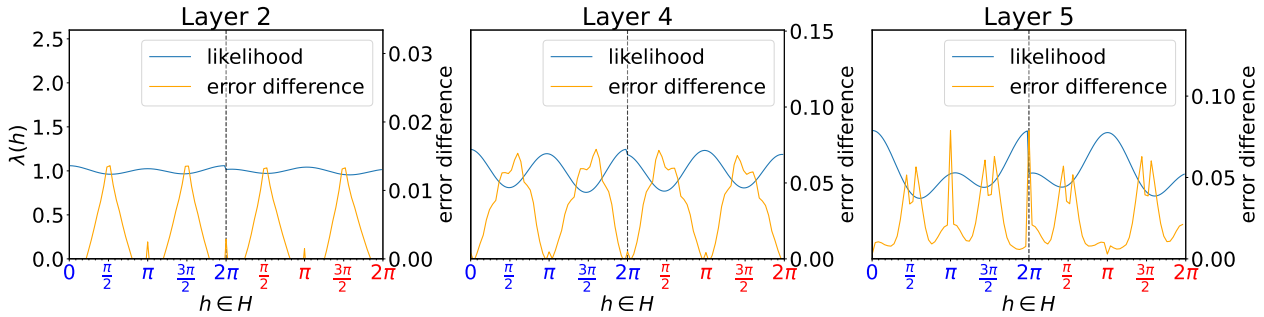
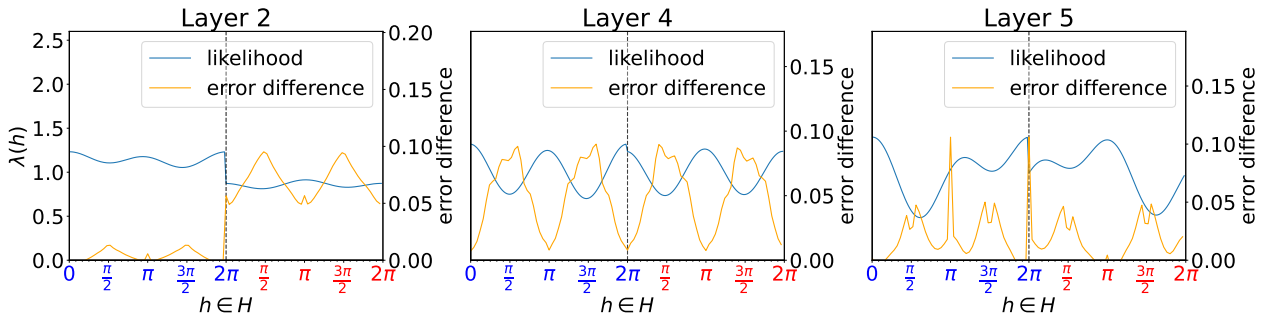


Figure 8.12: Development of KL-divergence during training with and without KL-regularisation.



(a) With KL regularisation.



(b) Without KL regularisation.

Figure 8.11: P-SCNN likelihoods for layer 2, 4 and 5 with and without KL-divergence on `Double MNIST` with $O(2)$ symmetries. Note that the scale of the equivariance error varies between the plots.

prediction is an invariant feature, the results suggest that the entire model is still invariant, despite the loss of equivariance. Furthermore, the unregularised setting shows that previously lost equivariance is incorrectly regained in the third layer at around $h = \frac{3\pi}{2}$, which does not occur in the regularised setting.

To further investigate this behaviour, we visualise the development of the KL-divergence for each epoch during training on `Vectors`' norm regression task in Figure 8.12. Here, we observe that, in both settings, the KL-divergence rises rapidly at the start of training. However, in the regularised setting, the KL-divergence starts to decrease around epoch 15, while in the unregularised setting the KL-divergence continues to increase steadily. This might be a result of the additional parameters introduced from the breaking of equivariance. To learn the task quickly at the beginning of training, it might be most beneficial to reduce equivariance to gain access to more parameters. However, with the help of the regularisation term, the model finds a way to use fewer parameters and still accurately compute the norm at a later stage. Another potential reason is that while computing the norm is an invariant task, given the available non-linearity and network structure, the most efficient way to compute the norm might be through non-invariant intermediate features that result in an invariant feature when combined. Both of these potential causes are supported by the fact that the error is lower when KL-regularisation is disabled.

For `Double MNIST`, Figure 8.11 shows that without KL regularisation our $O(2)$ P-SCNN similarly suffers from incorrectly regaining previously lost equivariance. In the second layer, equivariance is lost for the entire reflective domain. However, in the fourth layer equivariance for $h = 0$ and $h = \pi$ has been regained. Subsequently, the fifth layer models a relatively high likelihood for $h = \frac{3\pi}{2}$ and $h = \frac{\pi}{2}$, whereas the fourth layer models a significantly lower likelihood for these group elements.

Overall, these results indicate that while KL-regularisation might result in a performance reduction on certain datasets, it makes the learnt likelihood distributions more representative of the task.

8.6.3 Bandlimiting

In previous experiments, we used a bandlimit of $L = 2$ for the likelihood distributions of our partially equivariant models. By choosing a higher or lower level of bandlimiting, the flexibility of the likelihood distributions can be altered. In this section, we experiment with multiple levels of bandlimiting on `Double MNIST` and the subsets of `MedMNIST`. We present the results in Tables 8.10 and 8.11. We also show the learnt likelihood distributions for the fifth layer of our $O(2)$ P-SCNN trained on $O(2)$ `Double MNIST` for a bandlimit of $L = 0$ through $L = 4$ in Figure 8.13. Furthermore, due to the rapid increase in complexity, the highest level of bandlimiting for $O(3)$ is set at $L = 2$.

`Double MNIST` Table 8.10 shows that `Double MNIST` with $SO(2)$ symmetries requires a bandlimit of $L \geq 1$, with generally minimal differences between higher levels of bandlimiting, and a moderate improvement from $L = 1$ to $L = 2$ for $H = SO(2)$. A bandlimit of $L = 0$, which only allows reducing reflective equivariance, yields a performance improvement compared to the fully $O(2)$-invariant model, resulting in effectively the same performance as using a fully $SO(2)$-invariant model. However, it is significantly outperformed by higher levels of bandlimiting.

Network Group	L	Symmetries $SO(2)$	Symmetries $O(2)$
$SO(2)$	None	0.474 (0.016)	0.403 (0.021)
	1	0.883 (0.007)	0.794 (0.011)
	2	0.901 (0.005)	**0.823** (0.005)
	3	**0.908** (0.006)	0.821 (0.002)
	4	0.904 (0.004)	0.820 (0.013)
$O(2)$	None	0.391 (0.012)	0.399 (0.014)
	0	0.469 (0.010)	0.402 (0.003)
	1	0.894 (0.011)	0.780 (0.009)
	2	0.894 (0.004)	0.819 (0.010)
	3	0.889 (0.013)	0.817 (0.007)
	4	0.891 (0.006)	0.819 (0.018)

Table 8.10: `Double MNIST` test accuracies using various levels of bandlimiting for our $SO(2)$ and $O(2)$ P-SCNNs. For each symmetry, the highest accuracy is **bold**, and the highest for each network group within this type of symmetry is underlined. Standard deviations over 5 runs are denoted in parentheses.

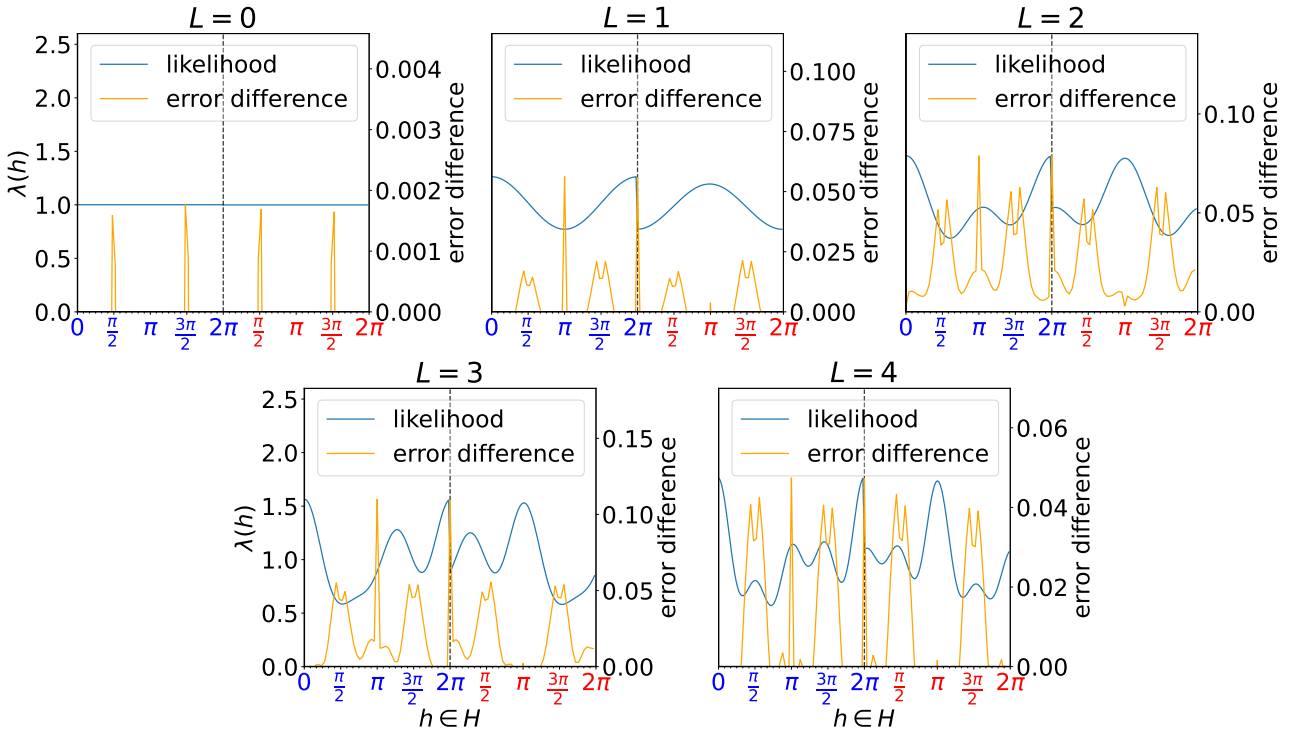


Figure 8.13: Likelihoods of the fifth layer of our $O(2)$ P-SCNN trained on $O(2)$ `Double MNIST` using a bandlimit of $L = 0$ through $L = 4$. Note that the scale of the equivariance error varies between the plots.

Under $O(2)$ symmetries, there is a more substantial performance benefit going from $L = 1$ to $L = 2$. Figure 8.13 shows that, using a bandlimit of $L = 1$, the model seems to struggle to describe low likelihoods for $h = \frac{\pi}{2}$ and $h = \frac{3\pi}{2}$, while maintaining a high likelihood for $h = \frac{\pi}{2}$. This problem does not occur under $L = 2$. Furthermore, a bandlimit of $L = 0$ no longer yields a substantial improvement over a fully $O(2)$-invariant model, which is supported by the uniform likelihood distribution in Figure 8.13.

For both symmetries, we observe that a bandlimit of $L = 4$ does not offer a substantial advantage over a bandlimit of $L = 2$ in terms of performance. Furthermore, the corresponding learnt likelihood distribution in Figure 8.13 shows a significantly more complex and, therefore, less easily interpretable likelihood distribution.

`MedMNIST` The results in Table 8.11 reveal that for `OrganMNIST3D` and `SynapseMNIST3D` a bandlimit of $L = 2$ (or $L = 3$ when available) obtains the highest performance. Using a bandlimit of $L = 4$ yields a performance reduction of around 2 percentage points, indicating minor overfitting. Moreover, `NoduleMNIST3D` sees a minimal performance difference for the various levels of bandlimiting, with only a bandlimit of $L = 0$ using $H = O(3)$ yielding a small performance improvement compared to the other models. This further indicates that this dataset is highly symmetric.

Network Group	L	Synapse	Nodule	Organ
$SO(3)$	None	0.738 (0.009)	0.873 (0.005)	0.607 (0.006)
	1	0.753 (0.018)	0.873 (0.007)	0.862 (0.012)
	2	0.770 (0.030)	0.871 (0.001)	0.902 (0.006)
	3	**0.771** (0.030)	0.872 (0.010)	0.902 (0.002)
	4	0.750 (0.006)	0.867 (0.009)	0.896 (0.003)
$O(3)$	None	0.743 (0.004)	0.868 (0.009)	0.592 (0.008)
	0	0.737 (0.008)	**0.879** (0.009)	0.636 (0.003)
	1	0.756 (0.022)	0.872 (0.006)	0.875 (0.005)
	2	0.769 (0.013)	0.873 (0.002)	**0.905** (0.004)

Table 8.11: `MedMNIST` test accuracies using various levels of bandlimiting for our $SO(2)$ and $O(2)$ P-SCNNs. For each type of symmetry, the highest accuracy is **bold**, and the highest for each network group within this type of symmetry is underlined. Standard deviations over 5 runs are denoted in parentheses.

Overall, these results show that in addition to the computational benefits, performing a higher degree of bandlimiting can act as a regularisation method, potentially improving performance and improving the interpretability of the likelihood distributions.

8.7 Competitive Results on Image Classification Tasks

In previous sections, all (partially) equivariant CNNs used a trivial output representation at the last convolution layer to ensure a structurally invariant architecture. This equivariance could only be broken by breaking the equivariance of the individual layers. Furthermore, while Fourier-based non-linearities allow for more complexity, we opted for Gated non-linearities to prevent models from breaking equivariance through the non-linearity.

In this section, we aim to obtain higher performance by removing these restrictions, in addition to providing insight into the behaviour of our approach when there are other potential sources of partial equivariance. To achieve this, rather than limiting the output of the final convolution layer to a trivial field, we use an irrep field consisting of irreps up to frequency 2. This allows the subsequent MLP unrestricted freedom in mixing the resulting irrep features. Furthermore, we replace the Gated non-linearity in all applicable models with a Fourier-based non-linearity. For the 2D groups we use the

FourierELU non-linearity. As discussed in Section 3.6, applying a Fourier-based activation function to 3D groups such as $SO(3)$ and $O(3)$ is computationally expensive. Therefore, we use quotient FourierELU non-linearities, with features over the quotient space $Q = SO(3)/SO(2)$ and $Q = O(3)/O(2)$, i.e. the 2-sphere. Here, both subgroups $K \leq H$ are defined as rotations (and reflections) around the z-axis. Since these spaces are smaller than the respective $SO(3)$ and $O(3)$ groups, these features require fewer channels and allow for a more efficient architecture. As a result, this also enables us to experiment with higher levels of bandlimiting of the likelihood distribution for $O(3)$.

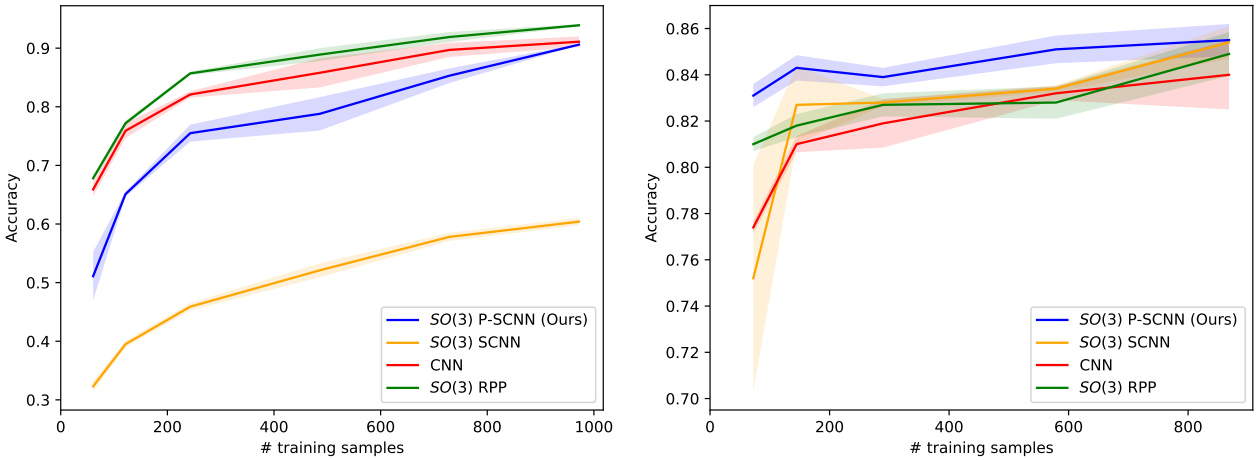
The results are summarised in Table 8.12, covering both `Double MNIST` with $O(2)$ symmetries and `OrganMNIST3D`. We observe noteworthy improvements in fully equivariant SCNNs across both datasets as a result of our modifications. Specifically, our approach yields minor improvements for $O(3)$ on `OrganMNIST3D` and $SO(2)$ on `Double MNIST`. Moreover, more substantial performance gains are observed for $SO(3)$ and $O(2)$ on `OrganMNIST3D`, as well as for $SO(2)$ on `Double MNIST`. Interestingly, all RPP models exhibit diminished performance after these modifications, particularly in the case of `Double MNIST`. This could potentially be attributed to the inherently unconstrained nature of RPP's residual CNN connections that already allow a high degree of flexibility in the base configuration. Consequently, our $SO(3)$ P-SCNN with $L = 4$ now matches the performance of the baseline RPP featuring a structurally invariant mapping and Gated non-linearity, while our $O(2)$ P-SCNN with $L = 3$ extends its lead over RPP models.

Network Group	**Partial Equivariance**	**L**	`OrganMNIST3D`		`Double MNIST` with $O(2)$ Symmetries	
			FourierELU	**Gated**	**FourierELU**	**Gated**
CNN	N/A	N/A	<u>0.921</u> (0.003)		<u>0.649</u> (0.019)	
$SO(n)$	None	N/A	0.879 (0.007)	0.607 (0.006)	0.842 (0.007)	0.403 (0.021)
	RPP	N/A	0.930 (0.011)	<u>0.936</u> (0.002)	0.617 (0.043)	0.779 (0.021)
	Ours	2	0.935 (0.003)	0.902 (0.006)	0.852 (0.009)	**<u>0.823</u>** (0.005)
		3	0.932 (0.003)	0.902 (0.002)	0.853 (0.016)	0.821 (0.002)
		4	**<u>0.941</u>** (0.007)	0.896 (0.003)	<u>0.855</u> (0.004)	0.820 (0.013)
$O(n)$	None	N/A	0.821 (0.005)	0.592 (0.008)	0.860 (0.005)	0.399 (0.014)
	RPP	N/A	<u>0.936</u> (0.004)	**<u>0.940</u>** (0.006)	0.677 (0.037)	0.772 (0.019)
	Ours	2	0.911 (0.007)	0.905 (0.004)	0.869 (0.005)	<u>0.819</u> (0.010)
		3	0.920 (0.008)	-	**<u>0.885</u>** (0.003)	0.817 (0.007)
		4	0.911 (0.003)	-	0.876 (0.006)	<u>0.819</u> (0.018)

Table 8.12: Test accuracies on `OrganMNIST3D` and `DoubleMNIST` comparing the performance of our baseline configurations (Gated) with the structurally non-invariant configurations using a Fourier based non-linearity. For each column, **bold** indicates the highest accuracy and <u>underline</u> denotes the highest accuracy for the given network group. Standard deviations over 5 runs are denoted in parentheses.

8.8 Data Ablation Study

One of the advantages of equivariant neural networks is their increased data-efficiency. To investigate the data-efficiency of the various approaches, we perform a data ablation study using the non-symmetric `OrganMNIST3D` and highly symmetric `NoduleMNIST3D` datasets. To prevent overfitting, we perform validation testing to choose the most optimal model, rather than taking the model in the last epoch as in previous experiments. See Figure 8.14 for the results.



(a) `OrganMNIST3D` **(b)** `NoduleMNIST3D`

Figure 8.14: Data ablation study on `OrganMNIST3D` and `NoduleMNIST3D`.

Figure 8.14a reveals that under the non-symmetric `OrganMNIST3D`, RPP maintains the highest performance, regardless of the number of training samples, followed by the basic CNN and subsequently our P-SCNN. However, when examining the performance drop, we note that the CNN maintains the flattest curve, followed by the RPP. As a result, our P-SCNN shows the highest performance reduction from data ablation, suggesting a worse data efficiency compared to the other methods. In contrast, Figure 8.14b shows that for the highly symmetric `NoduleMNIST3D` dataset, our approach maintains the highest performance, while also suffering from the smallest performance regression when using fewer data samples.

While the results may initially appear contradictory, it is important to consider the different initialisation and parameterisation approaches of the various methods. Our models start as fully equivariant at initialisation, requiring a gradual adjustment to handle non-symmetric tasks. Consequently, more training time or a larger dataset may potentially be required for effective learning. On the other hand, both RPP and the basic CNN start either partially equivariant or non-equivariant, respectively. Therefore, these methods may need less training time and fewer samples for non-symmetric tasks. However, for symmetric tasks, our method has an advantage as it initialises as fully equivariant, effectively reversing the roles.

8.9 Generalisability

Network Group	Partial Equivariance	Test Symmetries C_4	Test Symmetries $SO(2)$	Δ
CNN	N/A	0.868 (0.011)	0.602 (0.024)	−0.266
$SO(2)$	Restriction	0.915 (0.003)	0.718 (0.008)	−0.197
	RPP	0.890 (0.005)	0.665 (0.009)	−0.225
	Ours	**0.921** (0.005)	0.699 (0.013)	−0.222
$O(2)$	Restriction	0.920 (0.007)	**0.732** (0.009)	**−0.188**
	RPP	0.891 (0.003)	0.648 (0.014)	−0.243
	Ours	0.919 (0.006)	0.706 (0.022)	−0.213

(a) Trained on C_4 symmetries.

Network Group	Partial Equivariance	Test Symmetries D_4	Test Symmetries $O(2)$	Δ
CNN	N/A	0.711 (0.009)	0.474 (0.037)	-0.237
$SO(2)$	Restriction	0.842 (0.015)	0.624 (0.012)	-0.218
	RPP	0.821 (0.022)	0.569 (0.030)	-0.252
	Ours	**0.863** (0.009)	**0.637** (0.012)	-0.226
$O(2)$	Restriction	0.848 (0.007)	0.632 (0.007)	**−0.216**
	RPP	0.824 (0.009)	0.566 (0.029)	-0.258
	Ours	0.857 (0.011)	0.615 (0.023)	-0.242

(b) Trained on D_4 symmetries.

Table 8.13: Generalisability results for models trained on discrete C_4 and D_4 symmetries. Each of the models is evaluated on the discrete symmetries and the corresponding continuous $SO(2)/O(2)$ symmetries. The final column shows the difference in performance between the two cases. In each column, the best value (highest accuracy or lowest difference) is denoted in **bold**, with the best value for each network group underlined. Standard deviations over 5 runs are denoted in parentheses.

Another major appeal of equivariant neural networks is their improvement in terms of generalisability to data outside the training data. To investigate the influence of our approach on the ability to generalise, we train our baseline configurations on `Double MNIST` with discrete C_4 and D_4 symmetries and analyse their performance on `Double MNIST` with $SO(2)$ and $O(2)$ symmetries. It is important to note here that interpolation artefacts are introduced in all versions of `Double MNIST`, and as such models trained in discrete groups should be inherently adapted to these artefacts. We present our results on C_4 and D_4 symmetries in Tables 8.13a and 8.13b respectively.

We observe that our method consistently exceeds both RPP and conventional CNN in classification accuracy across various training and evaluation datasets. Moreover, it exhibits a less pronounced reduction in performance when tested on continuous groups. Although our $SO(2)$ P-SCNN achieves higher classification accuracy on D_4 symmetries, evaluated on both D_4 and $O(2)$ symmetries, it suffers from a higher performance regression compared to the baseline SCNNs that is restricted to C_1 equivariance (i.e., no equivariance) in the last two layers. Additionally, our $O(2)$ P-SCNN is outperformed by the restricted $O(2)$ SCNN. Moreover, on the C_4 symmetries, both restricted SCNNs achieve a higher accuracy on the regularisation to $SO(2)$ symmetries compared to our P-SCNNs.

These results reinforce the notion that equivariance enhances generalisation performance, as models with greater flexibility in breaking equivariance tend to exhibit a higher loss in performance. This trend is expected because partially equivariant models can adapt their equivariance for elements not included in the discrete groups, resulting in performance degradation when evaluated on continuous groups. In essence, the more flexible the model, the more easily it can adjust its equivariance for these group elements, which in turn leads to a decrease in performance in such generalisations.

8.10 Discussion of the Results

The extensive experimentation conducted in this study has provided significant insights into the performance and behaviour of our *preliminary* and mainly our *probabilistic* approach in learning the degree of equivariance in SCNNs. In the following paragraphs, we briefly recap and discuss the results and intermediate evaluations.

Probabilistic approach compared to preliminary approach. From the results in Section 8.1, it becomes apparent that both our *probabilistic* and *preliminary* approaches offer improved performance on a non-equivariant task compared to a fully equivariant E-MLP. Notably, the *probabilistic* approach both slightly outperforms the *preliminary* approach and achieves a more consistent performance. Section 8.4 illustrates that the equivariance error of the *preliminary* approach contains significantly higher frequencies compared to the equivariance errors and likelihood distributions of the *probabilistic* approach, which conform more closely to our expectations. This is attributed to the additional weight sharing of the *probabilistic* approach, which ensures that the modelling of partial equivariance is consistent throughout the group. As such, future paragraphs only consider the *probabilistic* approach.

Performance benefits of the probabilistic approach. In Sections 8.1, 8.2, and 8.3 we evaluated the quantitative performance of our probabilistic approach compared to CNNs, RPPs, regular SCNNs, and restricted SCNNs. Here we showed that our probabilistic approach yields a considerable performance gain in datasets with non-symmetric features compared to the regular SCNNs, with similar performance on more symmetric datasets. Furthermore, on `Double MNIST`, our probabilistic approach consistently outperformed RPPs and restricted SCNN baselines. Furthermore, the results in `MedMNIST` indicate that our probabilistic approach outperforms RPP on the fairly symmetric `SynapeMNIST3D` and highly symmetric `NoduleMNIST3D` datasets. However, our probabilistic approach is outperformed by RPPs on the non-symmetric `OrganMNIST3D` dataset in the structurally invariant setting, owing to the additional flexibility of RPPs in breaking equivariance. Nevertheless, as demonstrated in section 8.7, utilising the FourierELU non-linearity and a non-invariant final mapping elevates the performance of our probabilistic approach to the same level as RPPs when using an invariant final mapping with Gated non-linearity. Notably, applying the same non-invariant mapping with FourierELU non-linearity results in a performance reduction for RPPs. Furthermore, these results also demonstrate that even under a non-invariant structure, our probabilistic approach still outperforms the fully *equivariant* SCNNs. Finally, we observed that while it can be detrimental under a fully invariant SCNN to choose an equivariance group that is too large for given symmetries in the data, under our *probabilistic* approach, this effect is significantly less severe.

Interpretability of the likelihood distributions. In Section 8.4, we verify that, under our default configurations, the learnt likelihood distributions accurately describe the underlying layer-wise equivariance of the models as the likelihoods align with the measured equivariance errors. Furthermore, the resulting layer-wise likelihood distributions obtained on `Double MNIST` and `MedMNIST` align with our expectations regarding the existing symmetries in these datasets. In fact, the learnt likelihood distribution obtained on `Double MNIST` with $O(2)$ symmetries displayed a symmetry that we did not initially expect but is in fact an example of correct equivariance. Therefore, this is an interesting example of the benefits of the interpretability of the degree of equivariance.

Shared or layer-wise equivariance. The results in Section 8.4 demonstrate that the likelihood distributions learnt by our PE-MLPs, which are trained on norm regression with layer-wise equivariance, do not always accurately reflect the symmetries of the task—particularly under the Gated non-linearity. However, the results in Section 8.5 indicate that the use of a shared likelihood distribution leads to a representation that is more accurate and more reflective of the underlying symmetries. Therefore, these results suggest that the use of a shared degree of equivariance for subsequent linear MLP layers in favour of individual layer-wise degrees of equivariance.

Necessity of regularisation terms. The results in Sections 8.4 and 8.5 show that under the default configuration with both KL-regularisation and alignment loss enabled our models to produce accurate likelihood distributions. More precisely, the results in Section 8.6 show that adding KL-divergence and particularly our alignment loss term to the objective is essential to obtain aligned, accurate, and therefore interpretable likelihood distributions describing the symmetries of the given task. Here, disabling alignment loss results in unpredictably misaligned likelihood distributions, making it impossible to interpret the degree of equivariance without measuring the equivariance error. Furthermore, we observed that omitting KL-regularisation results in inaccurate regaining of previously lost equivariance on `Double MNIST`, and excessive breaking of equivariance on `Vector`'s $O(2)$-invariant norm regression task. Finally, for most tasks, disabling these terms results in a slight performance reduction, which suggests that they are effective in regularising the likelihood distributions to prevent overfitting.

Benefits of bandlimiting. Section 8.6 also covers our bandlimiting results. Here we show that in addition to the computational benefits of using fewer Fourier coefficients and therefore fewer weights, bandlimiting is an effective regularisation method. In most experiments, applying a bandlimit of $L = 4$ generally does not significantly improve performance over a bandlimit of $L = 2$, and sometimes even results in a performance regression. Furthermore, a higher level of bandlimiting results in more complex likelihood distributions that are harder to interpret. Additionally, bandlimiting offers a relatively interpretable method to control what kinds of symmetries can be broken and what kind of symmetries cannot be broken. For instance, setting the bandlimit to $L = 0$ only allows the model to distinguish between the reflective and non-reflective domains in $O(2)$. Likewise, a bandlimit of $L = 1$ does not allow network to separately model high equivariance for rotations of π and low equivariance for rotations of $\frac{\pi}{2}$ and $\frac{3\pi}{2}$, or vice versa.

Influence on data-efficiency. Section 8.8 showed that compared to RPPs and regular CNNs, our probabilistic approach struggles to maintain a high performance on highly non-symmetric datasets such as `OrganMNIST3D` when limited training data is available. In contrast, on highly symmetric datasets, such as `NoduleMNIST3D`, our probabilistic approach shows superior data-efficiency compared to RPPs and CNNs. This phenomenon is caused by differences in the initialisation of these methods. As our probabilistic approach initialises as fully equivariant, it requires a sufficient amount of data to learn to break that equivariance. This could perhaps be partially remedied by using a weaker KL-regularisation on the first layer. Conversely, CNNs and RPPs –due to their residual CNN layers– initialise as non-equivariant or partially equivariant. Therefore, these methods instead require a sufficient amount of data to learn to be fully equivariant.

Influence on generalisability. FInally, Section 8.9 covered our results on the generalisability of the various approaches by training our models on discrete groups and evaluating them on continuous groups. Although manually restricted SCNNs (without learnable equivariance) showed superior performance in terms of test accuracy on C_4 to $SO(2)$ generalisation, our probabilistic approach outperformed it in the generalisation D_4 to $O(2)$. However, in both cases, the manually restricted SCNN showed a smaller decrease in performance when evaluating the continuous group compared to the discrete group. This can be explained due to the fact that the degree of equivariance is static in the restricted SCNN. As a result, unlike RPPs and our P-SCNNs, it does not have the ability to adapt its degree of equivariance to discrete groups in a way that does not generalise well to continuous groups. Finally, due to the higher degree of flexibility in breaking equivariance through the lack of weight sharing, the RPP approach is significantly outperformed by both approaches, both in terms of test accuracies and in terms of performance regression.

8.11 Future Work

Despite the extensive experimental study performed in this thesis, additional research is still required to explore learnable equivariance in SCNNs, and equivariant neural networks in general. In this section, we discuss considerations and suggestions for future research.

Computational complexity and pruning. While our approach does not influence the computational complexity of SCNNs during inference, it introduces additional parameters to parameterise the constrained kernels, particularly for high levels of bandlimiting. Furthermore, our implementation as discussed in Chapter 6 requires us to recompute the steerable basis for each forward pass rather than only during initialisation; this can lead to increased computational overhead. Furthermore, as discussed in Chapter 6, our implementation, built on the `escnn` library, does not employ the most optimal method to build the steerable basis. This is due to the need to comply with current implementations of the library, which impose additional constraints on computational performance. As a result, further development and research is required to optimise the implementation of our approach, in addition to a more in-depth computational analysis.

More specifically, throughout the training of our current implementation, the bandlimiting is set as a static hyper-parameter. However, as we observed in various tasks in Section 8.4, it might become apparent during training that certain Fourier coefficients do not contribute to the likelihood distribution. To alleviate some of the computational strain, it could be beneficial to prune such non-contributing Fourier coefficients and their corresponding kernel weights.

Additionally, it may not be necessary for our approach to continuously adapt the degree of equivariance throughout the entire training process. It is possible that likelihood distributions only need to be updated during an initial phase of training. Subsequently, once these distributions have stabilised, they could be fixed, eliminating the need for further reconstruction of the steerable basis.

Pointclouds to prevent interpolation artefacts. All our P-SCNN results have been obtained on data using a discretised spatial domain, introducing interpolation artefacts. Although we took steps to account for such artefacts in our experiments by purposefully introducing these artefacts in every `Double MNIST` sample and by computing the error in equivariance errors, these artefacts potentially affect our results and the learnt likelihood distributions. As interpolation artefacts are more severe for rotations that are not multiples of $\frac{\pi}{2}$, our models could potentially reduce equivariance for such rotations, even though these rotations are within the symmetries of the task. Although this might be necessary for optimal performance, it might further complicate the likelihood distributions while also making them less indicative of the actual symmetries in the task at hand under a non-discretised setting. Therefore, we deem that it would be interesting to apply our approach on pointclouds rather than pixel or voxel data, as pointclouds allow for a continuous spatial domain, effectively eliminating interpolation artefacts.

Multi-disciplinary research. The interpretability of our approach offers a significant advantage: it can produce new insights into particular types of data and validate anticipated patterns. However, even though the `MedMNIST` datasets were selected as examples where interpretability could be particularly beneficial, due to a lack of medical expertise, this thesis does not perform such medically informed analyses. For future work, it could be effective to conduct a collaborative study with experts in relevant fields to undertake a detailed analysis of such datasets, using the interpretability aspects of our approach to validate hypotheses or discover novel patterns in the data.

Sharing equivariance. In Section 8.5, our PE-MLP experiments demonstrated the effectiveness of employing a shared degree of equivariance across multiple layers, yielding likelihood distributions that more accurately reflect the task at hand. Furthermore, we employed a shared degree of equivariance between the residual connections and the final convolution layer spanned by the residual connection in our 3D P-SCNNs to ensure that the equivariance of the resulting features is consistent. This concept

could be further extended to other architectures; for example, in equivariant auto-encoders, it might be beneficial for the encoder layer to mirror the degree of equivariance observed in the decoder layers, in reverse order.

Instead of increasing the degree with which equivariance is shared, future research could explore ways to reduce it. While our approach models a consistent degree of equivariance for all channels within a layer, this might not be optimal for detecting features with varying symmetries at a specific scale.

Our approach can already be used to design a network with a separate degree of equivariance for every output feature in the last layer by initialising a single channel layer for each of the output features. This not only facilitates the previously described concept, but also permits a separate examination of global symmetries for each of the output features in tasks such as image classification. However, more work is required to accurately model such a channel-wise equivariance for intermediate layers, as our approach does not offer a solution for the channel mixing of the subsequent layer. Future research could investigate methods to modulate the sharing of equivariance, aiming for more granular control that accommodates both varying symmetries across different scales and within the same scale.

Non-uniform initialisation of the likelihood. In our current implementation, we set the initialisation of the likelihood distribution as uniform over a pre-determined group, therefore only allowing the network to maintain or reduce the degree of equivariance. However, in our theoretical approach, we are not constrained to such an initialisation and could instead initialise a non-uniform likelihood distribution, thus allowing the model to maintain, reduce, or increase the degree of equivariance with respect to a specific group. For instance, given $H = O(2)$ the likelihood could be initialised such that it is uniform over the non-reflective domain, and 0 elsewhere, effectively yielding $SO(2)$ equivariance. Subsequently, the model might gradually learn to become uniform over the entire group $O(2)$. This could prove to be useful as a method for incorporating prior beliefs about the dataset into the model, similarly to RPPs.

Chapter 9

Conclusion

Various recent works have been proposed to allow equivariant models to learn partial equivariance. While only RPPs are directly applicable in the flexible framework of SCNNs, to our knowledge, no other works have been proposed that are specifically designed within this framework. Furthermore, while some approaches offer an interpretable and consistent parameterisation of the degree of equivariance, RPPs do not.

Building upon recent work in the SCNN framework that generalised SCNNs to any compact group in nD, we proposed a probabilistic approach to learn the degree of equivariance. We achieved this by reinterpreting the SCNN kernel constraint as an averaging operator through a uniform likelihood distribution over the group in terms of its Fourier coefficients. These coefficients are subsequently parameterised and updated with backpropagation, resulting in a learnable likelihood distribution. Through this likelihood distribution, our method uniquely allows for the direct modelling of equivariance with respect to any subgroup–or even any subset–S of a compact group H. This represents a clear advantage over RPPs, which might even require an infinite number of residual connections to achieve equivariance modelling for an unknown subgroup within a continuous compact group H.

Furthermore, we proposed to employ normalisation, KL-regularisation and our alignment loss to ensure that the likelihood distribution is interpretable and an accurate representation of the underlying equivariance. Through our results, we verified that this is indeed the case and showed that the resulting likelihood distributions exhibit a high degree of equivariance with respect to the expected symmetries. Furthermore, our results on `Double MNIST` with $O(2)$ symmetries demonstrated that this interpretability can be used to gain new insights into the data.

Besides the benefits of interpretability, we showed that the additional weight sharing through the likelihood distribution results in an improvement in performance and consistency by comparing our approach to a non-probabilistic preliminary approach. Additionally, we have shown that our approach significantly enhances performance on datasets exhibiting mixed symmetries when compared to SCNNs that are either fully invariant or fully equivariant. Moreover, it achieves a slight edge over both RPPs and SCNNs with a manual restriction across most of the tested datasets.

We showed that our proposed KL-regularisation and particularly alignment loss are essential for interpretable and accurate likelihood distributions. Moreover, we observed that for subsequent partially equivariant linear MLP layers, employing a shared degree of equivariance is favourable compared to individual layer-wise degrees of equivariance due to the lack of a receptive field. Finally, we show that tuning the bandlimiting of the likelihood distribution can be a useful tool to regularise the complexity of the likelihood distribution, reducing the computational complexity, and potentially increasing the interpretability of the likelihood distributions.

From a computational perspective, our approach requires additional parameters for the construction of the constrained kernels compared to a regular SCNN, particularly with high levels of bandlimiting. However, unlike RPPs, our approach does not alter the number or size of the features. Therefore, the

resulting kernel size remains the same. As such, at evaluation, the computational complexity is equal to regular SCNNs.

In conclusion, our probabilistic approach to learning the degree of equivariance in the SCNN framework has proven to be a powerful tool, providing both performance benefits and an additional degree of interpretability in tasks with varying or unknown symmetries. Through the flexible framework of SCNNs, our approach is able to model an interpretable degree of equivariance for many types of equivariant networks.

Bibliography

[1] Samuel G Armato III, Geoffrey McLennan, Luc Bidaut, Michael F McNitt-Gray, Charles R Meyer, Anthony P Reeves, Binsheng Zhao, Denise R Aberle, Claudia I Henschke, Eric A Hoffman, et al. The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans. *Medical physics*, 38(2):915–931, 2011.

[2] Erik J Bekkers, Maxime W Lafarge, Mitko Veta, Koen AJ Eppenhof, Josien PW Pluim, and Remco Duits. Roto-translation covariant convolutional networks for medical image analysis. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part I*, pages 440–448. Springer, 2018.

[3] Patrick Bilic, Patrick Christ, Hongwei Bran Li, Eugene Vorontsov, Avi Ben-Cohen, Georgios Kaissis, Adi Szeskin, Colin Jacobs, Gabriel Efrain Humpire Mamani, Gabriel Chartrand, et al. The liver tumor segmentation benchmark (lits). *Medical Image Analysis*, 84:102680, 2023.

[4] MEJ Callister, DR Baldwin, AR Akram, S Barnard, P Cane, J Draffan, K Franks, F Gleeson, R Graham, P Malhotra, et al. British thoracic society guidelines for the investigation and management of pulmonary nodules: accredited by nice. *Thorax*, 70(Suppl 2):ii1–ii54, 2015.

[5] Gabriele Cesa, Leon Lang, and Maurice Weiler. A program to build E(n)-equivariant steerable CNNs. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=WE4qe9xlnQw.

[6] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016.

[7] Taco S Cohen and Max Welling. Steerable cnns. *arXiv preprint arXiv:1612.08498*, 2016.

[8] Marc Finzi, Gregory Benton, and Andrew G Wilson. Residual pathway priors for soft equivariance constraints. *Advances in Neural Information Processing Systems*, 34:30037–30049, 2021.

[9] Marc Finzi, Max Welling, and Andrew Gordon Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In *International conference on machine learning*, pages 3318–3328. PMLR, 2021.

[10] Michael K Gould, Jessica Donington, William R Lynch, Peter J Mazzone, David E Midthun, David P Naidich, and Renda Soylemez Wiener. Evaluation of individuals with pulmonary nodules: When is it lung cancer?: Diagnosis and management of lung cancer: American college of chest physicians evidence-based clinical practice guidelines. *Chest*, 143(5):e93S–e120S, 2013.

[11] Edward G Gray. Axo-somatic and axo-dendritic synapses of the cerebral cortex: an electron microscope study. *Journal of anatomy*, 93(Pt 4):420, 1959.

[12] Emiel Hoogeboom, Jorn WT Peters, Taco S Cohen, and Max Welling. Hexaconv. *arXiv preprint arXiv:1803.02108*, 2018.

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[15] Cornelius JHM Klemann and Eric W Roubos. The gray area between synapse structure and function—gray's synapse types i and ii revisited. *Synapse*, 65(11):1222–1230, 2011.

[16] Thijs P Kuipers and Erik J Bekkers. Regular SE(3) group convolutions for volumetric medical image analysis. *arXiv preprint arXiv:2306.13960*, 2023.

[17] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[18] Leon Lang and Maurice Weiler. A wigner-eckart theorem for group equivariant convolution kernels. *arXiv preprint arXiv:2010.10952*, 2020.

[19] Anna Rita Larici, Alessandra Farchione, Paola Franchi, Mario Ciliberto, Giuseppe Cicchetti, Lucio Calandriello, Annemilia Del Ciello, and Lorenzo Bonomo. Lung nodules: size still matters. *European respiratory review*, 26(146), 2017.

[20] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[21] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[22] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems*, 29, 2016.

[23] George W Mackey. Induced representations of locally compact groups I. *Annals of Mathematics*, pages 101–139, 1952.

[24] Heber MacMahon, John HM Austin, Gordon Gamsu, Christian J Herold, James R Jett, David P Naidich, Edward F Patz Jr, and Stephen J Swensen. Guidelines for management of small pulmonary nodules detected on ct scans: a statement from the fleischner society. *Radiology*, 237(2): 395–400, 2005.

[25] Diego Marcos, Michele Volpi, Nikos Komodakis, and Devis Tuia. Rotation equivariant vector field networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5048–5057, 2017.

[26] Diego Marcos, Michele Volpi, Nikos Komodakis, and Devis Tuia. Rotation equivariant vector field networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5048–5057, 2017.

[27] Mircea Petrache and Shubhendu Trivedi. Approximation-generalization trade-offs under (approximate) group equivariance. *arXiv preprint arXiv:2305.17592*, 2023.

[28] David W Romero and Suhas Lohit. Learning partial equivariances from data. *Advances in Neural Information Processing Systems*, 35:36466–36478, 2022.

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[30] Morgan Sheng and Eunjoon Kim. The postsynaptic organization of synapses. *Cold Spring Harbor perspectives in biology*, 3(12):a005678, 2011.

[31] Peter Somogyi, Gabor Tamas, Rafael Lujan, and Eberhard H Buhl. Salient features of synaptic organisation in the cerebral cortex. *Brain research reviews*, 26(2-3):113–135, 1998.

[32] Tycho van der Ouderaa, David W Romero, and Mark van der Wilk. Relaxing equivariance constraints with non-stationary continuous filters. *Advances in Neural Information Processing Systems*, 35:33818–33830, 2022.

[33] Tycho FA van der Ouderaa, Alexander Immer, and Mark van der Wilk. Learning layer-wise equivariances automatically using gradients. *arXiv preprint arXiv:2310.06131*, 2023.

[34] Dian Wang, Xupeng Zhu, Jung Yeon Park, Robert Platt, and Robin Walters. A general theory of correct, incorrect, and extrinsic equivariance. *arXiv preprint arXiv:2303.04745*, 2023.

[35] Rui Wang, Robin Walters, and Rose Yu. Approximately equivariant networks for imperfectly symmetric dynamics. In *International Conference on Machine Learning*, pages 23078–23091. PMLR, 2022.

[36] Maurice Weiler and Gabriele Cesa. General E(2)-equivariant steerable CNNs. *Advances in neural information processing systems*, 32, 2019.

[37] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco S Cohen. 3D steerable CNNs: Learning rotationally equivariant features in volumetric data. *Advances in Neural Information Processing Systems*, 31, 2018.

[38] Maurice Weiler, Fred A Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 849–858, 2018.

[39] Maurice Weiler, Patrick Forré, Erik Verlinde, and Max Welling. Coordinate independent convolutional networks–isometry and gauge equivariant convolutions on riemannian manifolds. *arXiv preprint arXiv:2106.06020*, 2021.

[40] Marysia Winkels and Taco S Cohen. 3d g-cnns for pulmonary nodule detection. *arXiv preprint arXiv:1804.04656*, 2018.

[41] Daniel Worrall and Gabriel Brostow. Cubenet: Equivariance to 3d rotation and translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 567–584, 2018.

[42] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5028–5037, 2017.

[43] Xuanang Xu, Fugen Zhou, Bo Liu, Dongshan Fu, and Xiangzhi Bai. Efficient multiple organ localization in ct image using 3d region proposal network. *IEEE transactions on medical imaging*, 38(8):1885–1898, 2019.

[44] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

Appendix A

Additional Training Details

Here we present some additional training details for reproducibility. Table A.1 contains the maximum level of bandlimiting of the intermediate features for each group. Table A.2 contains the level of bandlimiting for each of the three layers in the E-MLP-based configurations. In addition to the degrees of feature bandlimiting for each layer, Table A.3 presents the strides and padding for each of the layers for the SCNN-based configurations.

Group	C_1	C_4	$SO(2)$	D_1	D_4	$O(2)$	$SO(3)$	$O(3)$
Maximum	0	2	4	0	2	4	3	2

Table A.1: Maximum level of bandlimiting of the intermediate for each supported transformation group.

Layer	**Features Bandlimiting**
1	4
2	3
3	*Trivial*

Table A.2: Bandlimiting of the intermediate features of the $O(2)$ E-MLP configurations. '*Trivial*' means that the output is only a trivial representation.

Layer	Features Bandlimiting	Stride	Padding	
			2D SCNN	3D SCNN
Block 1	$\min(2, L_{\max})$	1	2	2
Block 2	$\min(3, L_{\max})$	1	2	2
Avg. Pooling 1	-	2	1	1
Block 3	$L_{\max}$	2	2	1
Avg. Pooling 2	-	2	1	1
Block 4	$L_{\max}$	2	0	2
Block 5	$\min(2, L_{\max})$	1	1	1
Avg. Pooling 3	-	1	1	1
Convolution	*Trivial* or $L_{\max}$	1	0	0
Res. Block 1	$\min(3, L_{\max})$	1	-	2
Res. Avg. Pooling	-	2	-	0
Res. Block 2	$L_{\max}$	1	-	0
Res. Convolution	*Trivial* or $L_{\max}$	1	-	0

Table A.3: Additional details for all layers in our SCNN-based configurations. Features bandlimiting denotes the degree of bandlimiting of the layer's output features. Here, $L_{\max}$ refers to the maximum level of bandlimiting for each specific group (Table A.1). The residual blocks in the last few rows are only applicable for our 3D SCNNs acting on `MedMNIST`, all other rows are applicable for all (2D and 3D) SCNN-based configurations. The output features of the final convolution and residual convolution layers are only trivial features or band-limited features up to $L_{\max}$ depending on whether it is the base configuration with a structurally invariant mapping or the non-invariant mapping from Section 8.7.

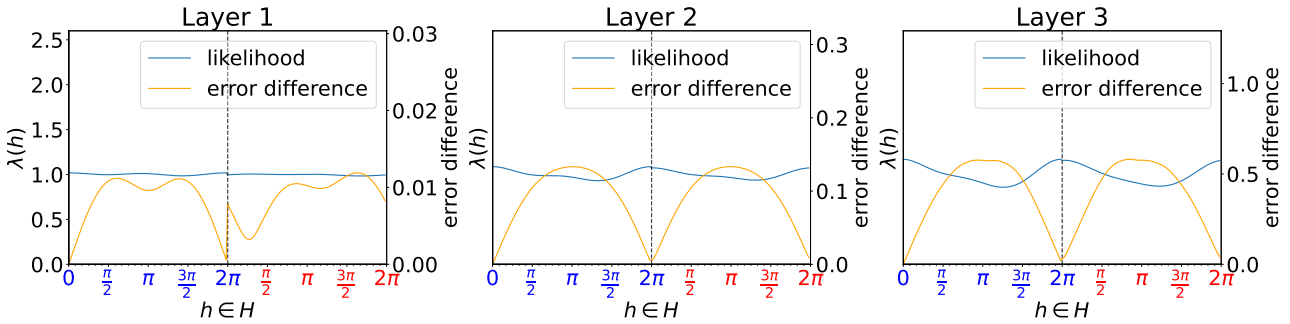
Appendix B

Additional Results

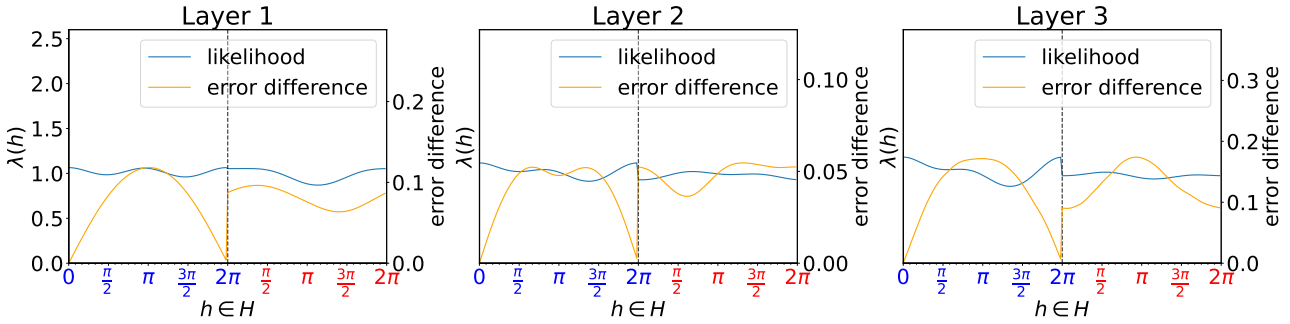
In this section, we present additional results on the `Vectors` dataset. Table B.1 compares MSE test errors between our baselines on a combined norm and angle regression task using two output features. Here, we separately report the errors for each task and the combined error. In Section 8.4, we presented the equivariance errors and likelihood distributions for the second layer of our *probabilistic* $O(2)$ PE-MLPs. Figure B.1 shows the corresponding results for all three layers. Furthermore, we present the equivariance errors for the three layers in our *preliminary* $O(2)$ PE-MLPs under the Gated non-linearity in Figure B.2.

Model	**Non-linearity**	**Angle**	**Norm**	**Combined**
MLP	ELU	0.192 (0.163)	0.092 (0.059)	0.284 (0.159)
E-MLP [5]	Gated	0.536 (0.061)	0.108 (0.037)	0.644 (0.077)
	FourierELU	0.658 (0.221)	0.005 (0.009)	0.663 (0.228)
Ours (Preliminary)	Gated	0.139 (0.043)	0.051 (0.005)	0.190 (0.042)
Ours (Preliminary)	FourierELU	0.079 (0.061)	0.002 (0.001)	0.081 (0.056)
Ours (Preliminary + noise)	Gated	0.131 (0.039)	0.040 (0.006)	0.171 (0.038)
	FourierELU	**0.074** (0.056)	0.002 (0.001)	**0.076** (0.056)
Ours (Probabilistic)	Gated	0.077 (0.033)	0.056 (0.010)	0.133 (0.028)
	FourierELU	0.101 (0.068)	**0.001** (0.001)	0.102 (0.068)

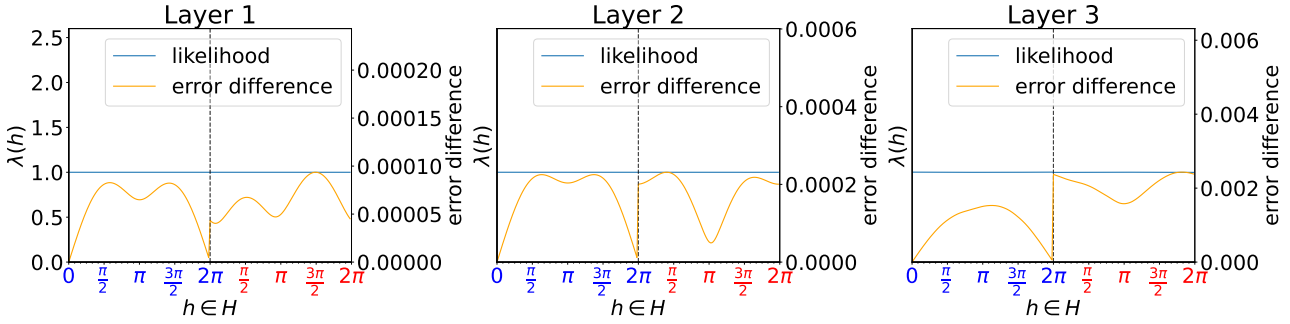
Table B.1: Test MSE scores of our approaches and the baseline models on `Vectors` trained on a combined norm/angle regression task. **Bold** indicates the lowest MSE error for the specific task. For each non-linearity, underline indicates the lowest MSE error for this specific non-linearity on for the given task. The standard deviations over 5 runs are denoted within parentheses.



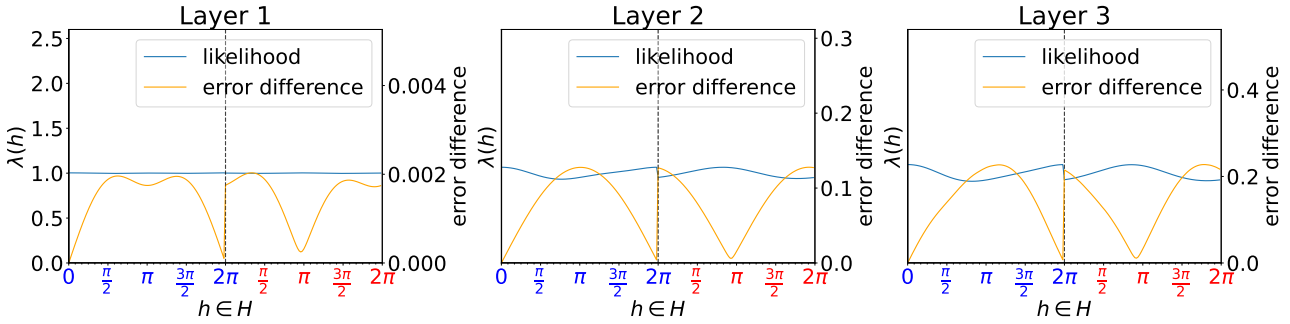
(a) Angle FourierELU



(b) Angle Gated

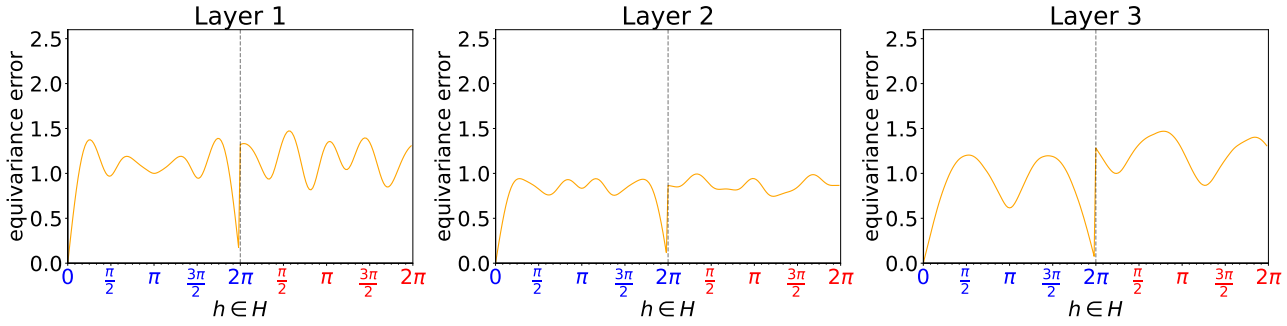


(c) Norm FourierELU

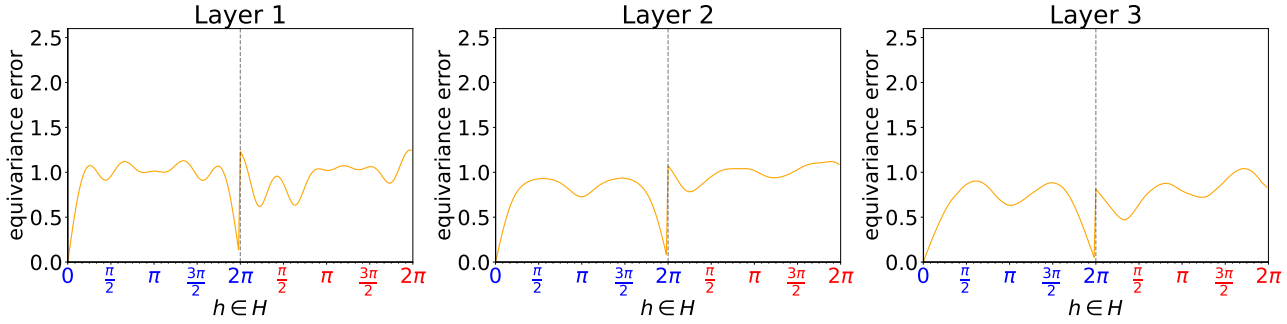


(d) Norm Gated

Figure B.1: Learnt equivariance likelihood λ and error difference for layers $1-3$ in an $O(2)$ PE-MLP trained on angle or norm regression, with Gated and FourierELU non-linearities. Error difference is calculated against an $O(2)$ E-MLP. The dotted line marks the $O(2)$ reflection domain transition.



(a) Preliminary on Norm



(b) Preliminary on Angle

Figure B.2: Measured equivariance error of layers 1 through 3 in an $O(2)$ PE-MLP (using preliminary approach) trained on angle or norm regression with gated non-linearity. Error difference is calculated against an $O(2)$ E-MLP. The dotted line marks the $O(2)$ reflection domain transition.

List of Figures

List of Tables

List of Algorithms

List of Definitions

List of Theorems

List of Examples