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

Lars Veefkind [1] [2]   **Gabriele Cesa** [1] [3]

Abstract

Steerable convolutional neural networks (SCNNs) enhance task performance by modelling geometric symmetries through equivariance constraints on weights. Yet, unknown or varying symmetries can lead to overconstrained weights and decreased performance. To address this, this paper introduces a probabilistic method to learn the degree of equivariance in SCNNs. We parameterise the degree of equivariance as a likelihood distribution over the transformation group using Fourier coefficients, offering the option to model layer-wise and shared equivariance. These likelihood distributions are regularised to ensure an interpretable degree of equivariance across the network. Advantages include the applicability to many types of equivariant networks through the flexible framework of SCNNs and the ability to learn equivariance with respect to any subgroup of any compact group without requiring additional layers. Our experiments reveal competitive performance on datasets with mixed symmetries, with learnt likelihood distributions that are representative of the underlying degree of equivariance.

1. Introduction

Natural images and structures often exhibit various geometrical symmetries, such as rotations, translations, and reflections. Since LeCun et al. (1995) introduced translation-equivariant Convolutional Neural Networks (CNNs), integrating *group* equivariance properties as a form of inductive bias into deep learning architectures has been an active area of research (Weiler et al., 2021; Bronstein et al., 2021).

While incorporating higher degrees of equivariance can serve as a powerful inductive bias, over-constraining a model can lead to diminished performance by preventing the use of potentially significant features (Wang et al., 2022b; Petrache & Trivedi, 2023). 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 scale (Fig. 1). In a CNN, with its hierarchical layers processing features of varying sizes (receptive fields), selecting the appropriate transformation group beforehand poses a challenge, demanding substantial knowledge about the dataset and the layers' receptive fields.
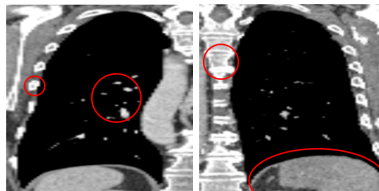

Figure 1. Coronal CT view of left and right lung (Yang et al., 2023). Smaller features ◯ (i.e., textures/components) are orientation invariant in determining whether the object is a lung, unlike the task of distinguishing between the two lungs as a whole, as they are (approximately) mirrored versions of each other.

To address these complexities, various strategies have been proposed to autonomously learn the relaxation of equivariance constraints on a layer-by-layer basis for Group CNNs (Cohen & Welling, 2016a). Among these are Residual Pathway Priors (RPPs) (Finzi et al., 2021), non-stationary GCNNs (van der Ouderaa et al., 2022), and Partial GCNNs (Romero & Lohit, 2022). We cover these methods in more detail in Sec. 4. Despite the flexibility of the framework of steerable CNNs (Cohen & Welling, 2016b) compared to GCNNs, only very few methods are compatible with this framework. To the best of our knowledge, RSteer (Wang et al., 2022b) is the only method directly developed for SCNNs, while RPP provides a generalised method for modulating equivariance regardless of the network type. However, with no separate parameterisation describing the degree of equivariance, such as a likelihood distribution over the transformation group as proposed by Romero & Lohit (2022), it is impossible to obtain an interpretable degree

of equivariance from these methods without performing computationally expensive analyses. Furthermore, RPPs achieve this modulation through residual connections that are non-equivariant, or equivariant with respect to a smaller subgroup of the large group. Still, accurately choosing the relevant equivariance subgroups can be challenging: for instance, there are infinitely many $SO(2)$ subgroups in $SO(3)$. We cover these shortcomings in more details in Sec. 4.

1.1. Contributions and Outline

Building upon the SCNN framework developed by Cesa et al. (2022), we propose a general solution to obtain a learnable degree of equivariance in SCNNs. Parameterised by an interpretable likelihood distribution over the transformation group, our solution models partial equivariance by using connections that are not available in an equivariant setting.

In Sec. 3.1, we exploit the interpretation of the kernel projection as an implicit uniform averaging over the group. We show that this uniform distribution can be modelled using *Fourier coefficients* in Sec. 2.2. By updating these coefficients through backpropagation, we enable the learning of a likelihood distribution over any compact group in nD, including continuous groups. The learnt likelihood corresponds to the model's inherent degree of equivariance and is easily obtained with the inverse Fourier transform without forward passes, achieving an **interpretable** degree of equivariance. Moreover, to prevent a layer from erroneously regaining previously lost equivariance, we propose employing *KL-divergence* (Kullback & Leibler, 1951) between subsequent layers. Additionally, our approach allows the user to tune the *band-limiting* of the Fourier coefficients as means of regularisation and improved computational performance.

To evaluate our approaches, we conduct an extensive series of experiments in Sec. 5, comparing our method with various baselines. These include regular SCNNs with pre-tuned degrees of equivariance through group restrictions (Weiler & Cesa, 2019), RPPs (Finzi et al., 2021) and Relaxed SCNNs (Wang et al., 2022b), as these are, to our knowledge, the only approaches directly applicable to SCNNs. Our experiments assess the performance impact on various computer vision datasets. Additionally, we analyse the interpretability and correctness of the learnt likelihood distributions and study the influence of bandlimiting and KL-divergence on both performance and interpretability. Finally, we compare data-efficiency and generalisation capabilities between our approach and the baselines. Through these experiments, we offer the following contributions:

- We develop a *probabilistic* approach to model partial equivariances in SCNNs and show that it achieves competitive performance on partially symmetric datasets.
- Our findings reveal that the learnt likelihood distributions accurately represent the degree of equivariance, enhancing interpretability.

- We demonstrate that bandlimiting serving as a useful tool to regularise the degree of equivariance.
- We show that it outperforms other methods in generalising from discrete to continuous group symmetries.
- We show that our approach achieves high data-efficiency on fairly symmetric datasets.

2. Background on Steerable CNNs

This section briefly covers the framework of SCNNs. For a comprehensive overview of all mathematical preliminaries we refer to Apx. B. We restrict ourselves to convolutional networks acting on Euclidean spaces $X = \mathbb{R}^n$ and compact groups $H \leq O(n)$, resulting in networks equivariant to transformations in $G = (\mathbb{R}^n, +) \rtimes H$, i.e. the *semidirect product* between translations $(\mathbb{R}^n, +)$ and H.

2.1. Steerable CNNs

As the name suggests, each layer l in a G-equivariant neural network satisfies the *equivariance constraint*:

$$\forall x \in X, \forall g \in G: \quad l(g \cdot x) = g \cdot l(x), \quad (1)$$

meaning that these networks have the property to commute with the action of a symmetry group G on their input, intermediate features and output. In this paper, we focus on SCNNs. These networks operate on *feature fields*, which are comparable to feature maps, except that they are associated with a well defined action of G and a geometric *type*. This type is described by an orthogonal group *representation* $\rho\colon H \to \mathbb{R}^{d_\rho \times d_\rho}$ that determines the transformation law of the d_ρ-dimensional feature vectors in the feature field. Specifically, a feature field of type ρ is a map $f\colon \mathbb{R}^n \to \mathbb{R}^{d_\rho}$ that, given a group action $g = (t, h) \in (\mathbb{R}^n, +) \rtimes H$, transforms by spatially moving feature vectors from $h^{-1}(x - t)$ to x and uses $\rho(h)$ to transform each individual feature[1].

The simplest example of feature fields are scalar fields, used in conventional CNNs. These fields transform according to the trivial representation $\psi_0(h) = 1$. GCNNs are special cases and transform according to the *regular representation* of G. As neural networks are often composed of multiple feature fields f_i, we often describe the entire feature field as a *direct sum* $f = \bigoplus_i f_i$ of the individual feature fields, i.e. by stacking them along the channel dimension. The resulting feature space consequently transforms under the *direct sum representation* $\rho = \bigoplus_i \rho_i$; see Def. B.27.

The goal of SCNNs is to form a general framework to obtain equivariance, regardless of the types of the feature fields. To this end, it has been shown that the most general equivariant maps between feature spaces are convolutions

---

[1]This is modelled via the *induced representation* $\left(\left[\mathrm{Ind}_H^{(\mathbb{R}^n, +) \rtimes H} \rho \right] (th) f \right)(x) = \rho(h) f(h^{-1}(x - t))$.

with H-steerable kernels Weiler et al. (2018a). Given input and output feature field types $\rho_{\text{in}}\colon H \to \mathbb{R}^{d_{\text{in}} \times d_{\text{in}}}$ and $\rho_{\text{out}}\colon H \to \mathbb{R}^{d_{\text{out}} \times d_{\text{out}}}$ respectively, an H-steerable kernel is a convolution kernel $K\colon \mathbb{R}^n \to \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ that satisfies the *steerability constraint* $\forall h \in H, x \in \mathbb{R}^n$:

$$K(x) = \rho_{\text{out}}(h)K(h^{-1}x)\rho_{\text{in}}(h)^{-1}. \tag{2}$$

Therefore, in order to build SCNNs we need to find a basis of the vector space of such H-steerable kernels that can be used to parameterise the convolutions.

2.2. Generating Steerable Bases

Cesa et al. (2022) derives a general parameterisation of H-steerable kernels. In this section, we offer a summary of an *alternative derivation* that arrives at the same result, but with a probabilistic interpretation; see Apx. C.2 for a detailed proof. In our derivation, we consider a general Euclidean space $\mathbb{R}^n$, and compact group $H \leq O(n)$ in $G = (\mathbb{R}^n, +) \rtimes H$. As such, we can assume that all representations are *orthogonal*, and thus $\rho(g^{-1}) = \rho(g)^{-1} = \rho(g)^\top$.

Irrep Decomposition While ρ_{in} and ρ_{out} can be any type of representation, through *irrep decomposition* (Thm. B.30) we can decompose any orthogonal representation into a direct sum of convenient *irreducible representations* (or *irreps*): $\rho(h) = Q \left[\bigoplus_{i \in I} \psi_i(h) \right] Q^T$, where ψ_i are irreps in the set of irreps $\widehat{H}$ of H, I is an index set ranging over $\widehat{G}$, and Q is a change of basis. Weiler & Cesa (2019) show that the solution for any ρ_{in} and ρ_{out} can be obtained from the solution of their irreps decomposition. Therefore, we can assume the input and output representations to be irreps ψ_l and ψ_J. Furthermore, it is convenient to use the vectorised form of our kernel: $\kappa(\cdot) = \text{vec}\,(K(\cdot))\colon X \to \mathbb{R}^{d_{\text{in}} \cdot d_{\text{out}}}$. Finally, as the constraint is linear, it can be solved by projecting unconstrained square integrable kernel $\widehat{\kappa} \in L^2(\mathbb{R}^n)$ via integration; a similar idea was used in (Van der Pol et al., 2020). As this is essentially performing a uniform averaging over the group H, we can safely inject a uniform likelihood distribution $\mu(h) = 1 \,\forall h \in H$, assuming H is a compact subgroup of a group with a finite Haar measure. Thus[2]:

$$\kappa(x) = \int_{h \in H} \mu(h)(\psi_l \otimes \psi_J)(h)\widehat{\kappa}(h^{-1}x)dh. \tag{3}$$

Tensor Product $(\psi_l \otimes \psi_J)$ is a *tensor product representation* (Def. B.35). It can be decomposed into a sum of irreps $\psi_{j'}$ via *Clebsch-Gordan decomposition* (Thm. B.37):

$$\psi_l \otimes \psi_J = \sum_{j'}^{} \sum_s^{[j'(Jl)]} \left[\text{CG}_s^{j'(Jl)} \right]^\top \psi_{j'} \text{CG}_s^{j'(Jl)}. \tag{4}$$

---
[2] $\text{vec}(ABC) = (C^\top \otimes A)\,\text{vec}(B)$, with Kronecker product $\otimes$.

$[j'(Jl)]$ is the *multiplicity* of irrep $\psi_{j'}$. $\text{CG}_s^{j'(Jl)}$ are the *Clebsch-Gordan coefficients* for the s-th occurrence of $\psi_{j'}$.

Steerable Basis An *unconstrained* kernel $\widehat{\kappa}$ can be parameterised via weights $W_{j,k} \in \mathbb{R}^{d_J d_l \times d_j}$ and an H-steerable basis (Def. B.46) $\mathcal{B} = \left\{ Y_j^k\colon \mathbb{R}^n \to \mathbb{R}^{d_j} \mid \psi_j \in \widehat{H}, k \right\}$:

$$\widehat{\kappa}(x) = \sum_{j,k} W_{j,k} Y_j^k(x). \tag{5}$$

Let $\text{unvec}(\cdot)$ be the reverse of the vectorisation. We apply this parameterisation and the *Clebsch-Gordan decomposition* in Eq. 3, and substitute $W_{j,j',k,s} = \text{vec}\left(\text{CG}_s^{j'(Jl)} W_{jk} \right)$:

$$\kappa(x) = \sum_{j,k} \sum_{j'} \sum_s^{[j'(Jl)]} \left[\text{CG}_s^{j'(Jl)} \right]^\top$$
$$\text{unvec}\left[\int \mu(h)\,(\psi_j \otimes \psi_{j'})(h)dh\, W_{j,j',k,s} \right] Y_j^k(x) \tag{6}$$

Here the integral contains a tensor product, and can thus be decomposed through the *Clebsch-Gordan decomposition*:

$$\int \mu(h)\,(\psi_j \otimes \psi_{j'})\,(h)dh =$$
$$Q \bigoplus_i \bigoplus_r^{[i(jj')]} \left(\int \mu(h)\psi_i(h)dh \right) Q^\top \tag{7}$$

Peter Weyl and inverse Fourier Transform The *Peter-Weyl theorem* (Peter & Weyl, 1927) (Thm. B.42) states that the entries of *complex* irreps of a compact group H span the vector space $L^2(H)$. However, in this work we consider *real* irreps, which may contain redundant entries. A proper basis can be constructed by selecting an independent subset of $n_i = \frac{d_i}{[0(ii)]}$ columns of each irrep ψ_i [3], where $[0(ii)]$ is the multiplicity of the trival irrep ψ_0 in $\psi_i \otimes \psi_i$, as in Eq. 4. We denote the non-redundant columns of $\psi_i(h)$ with $\overline{\psi}_i(h) \in \mathbb{R}^{d_i \times n_i}$ and define the map $\mathcal{R}_{\psi_i}\colon \mathbb{R}^{d_i \times n_i} \to \mathbb{R}^{d_i \times d_i}$ that performs the inverse projection.

Using $\mathcal{R}_{\psi_i}$, we can describe $\mu(h)$ through its inverse Fourier Transform (IFT):

$$\mu(h) = \sum_{\psi_i \in \widehat{H}} \sqrt{d_i}\text{Tr}\left(\psi_i(h)^\top \mathcal{R}_{\psi_i}\left(\widehat{\mu}(\overline{\psi}_i) \right) \right) \tag{8}$$

Due to the orthogonality of the Fourier coefficients, by expanding μ via its IFT, the integral in Eq. 7 simplifies to:

$$\int \mu(h)(\psi_j \otimes \psi_{j'})\,(h)dh = Q \bigoplus_i \bigoplus_r^{[i(jj')]} \frac{\mathcal{R}_{\psi_i}\left(\widehat{\mu}(\overline{\psi}_i) \right)}{\sqrt{d_i}} Q^\top \tag{9}$$

---
[3] This construction only works under a proper choice of basis for ψ_i; $[0(ii)] \in \{1, 2, 4\}$ depending on the type of irrep ψ_i. See (Cesa et al., 2022) Apx. C.3 for more details.

As $\mu(h)$ is uniform, only the Fourier coefficients $\widehat{\mu}(\psi_0) = 1$ corresponding to the *trivial* representation ψ_0 with size $d_{\psi_0} = 1$ are non-zero. As such, the summation selects only certain relevant columns $\{Q_r = \mathrm{CG}_r^{0(jj')}\}_r^{[0(jj')]}$ from Q:

$$\int \mu(h)(\psi_j \otimes \psi_{j'})(h)dh = \sum_r^{[0(jj')]} Q_r Q_r^\top. \tag{10}$$

One can verify[4] that the set $\{\mathrm{unvec}(Q_r)\}_r^{[0(jj')]}$ forms a basis for all equivariant maps from ψ_j to $\psi_{j'}$. Hence, whenever $\psi_j \neq \psi_{j'}$, $[0(jj')] = 0$ (*Schur's Lemma* Apx. B.41); in this case, Eq. 10 results in a null-map and the corresponding weights can be discarded. Then, Eq. 6 turns into:

$$\kappa(x) = \sum_{j,k} \sum_s^{[j(Jl)]} \left[\mathrm{CG}_s^{j(Jl)} \right]^\top \mathrm{unvec} \left[\sum_r^{[0(jj)]} c_r^j W_{j,k,s,r} \right] Y_j^k(x), \tag{11}$$

where $Q_r^\top \in \mathbb{R}^{1 \times d_j d_{j'}}$ and $W_{j,j',k,s}$ have been merged into a single weight $W_{j,k,s,r}$ to reduce parameters.

3. Learning Equivariance

This section details how we obtain an interpretable and learnable degree of equivariance in SCNNs.

3.1. Leveraging the Averaging Operator

In SCNNs, equivariance stems from the uniform averaging over the group H performed during the projection of the unconstrained kernels, ensuring that each element $h \in H$ is equally weighted. To learn the degree of equivariance, we propose to replace the uniform likelihood distribution $\mu(h)$ with a learnable likelihood distribution $\lambda(h)$. As a result, we substitute $\mu(h)$ with $\lambda(h)$ in Eq. 9:

$$\int \lambda(h)(\psi_j \otimes \psi_{j'})(h)dh = Q \bigoplus_i \bigoplus_r^{[i(jj')]} \frac{\mathcal{R}_{\psi_i}\left(\widehat{\lambda}(\overline{\psi}_i)\right)}{\sqrt{d_i}} Q^\top \tag{12}$$

While the uniformity of $\mu(h)$ meant only the Fourier coefficient of the trivial representation had to be considered, $\lambda(h)$ is not necessarily uniform. As such, the Fourier coefficient of any representation ψ_i can be nonzero, and the previous simplifications can no longer be performed. Thus, using weights $W_{j,j',k,s} \in \mathbb{R}^{d_j d_{j'}}$ the final projection becomes[5]:

$$\kappa(x) = \sum_{j,k} \sum_{j',s} \left[\mathrm{CG}_s^{j'(Jl)} \right]^\top \mathrm{unvec} \left[c^{jj'} W_{j,j',k,s} \right] Y_j^k(x), \tag{13}$$

---

[4]Eq. 10 essentially defines the orthogonal projection of a (vectorised) matrix into the space of (vectorised) matrices commuting with ψ_j and $\psi_{j'}$.

[5]Note here that the summation over j' has returned, as the projection no longer amounts to a nullmap if $\psi_j \neq \psi_{j'}$.

where $c^{jj'}$ is defined using the Fourier coefficients:

$$c^{jj'} = Q \left(\bigoplus_i \bigoplus_r^{[i(jj')]} \frac{\mathcal{R}_{\psi_i}\left(\widehat{\lambda}(\overline{\psi}_i)\right)}{\sqrt{d_i}} \right). \tag{14}$$

To learn the degree of equivariance we propose to initialise the Fourier coefficients such that $\lambda(h)$ is a uniform distribution, and the projection is initially fully H-equivariant:

$$\widehat{\lambda}(\psi_i) = \begin{cases} 1 & i = 0 \\ \mathbf{0}_{d_i \times n_i} & i \neq 0 \end{cases}. \tag{15}$$

These Fourier coefficients $\widehat{\lambda}(\psi_i)$ are stored as learnable parameters and updated through back-propagation during training, permitting a partially H-equivariant projection[6].

3.2. Ensuring Interpretable Equivariance

Through IFT it is possible to construct the underlying likelihood distribution $\lambda(h)$ from the corresponding learnt Fourier coefficients $\widehat{\lambda}(\psi_i)$. However, a few steps are required to ensure interpretable likelihoods.

Normalisation To ensure consistency in the magnitude of the likelihood distribution we normalise the learnt likelihood distribution λ to a PDF. This improves interpretability, and also ensures that any potential weight regularisation cannot simply uniformly increase the scale of the learnt likelihood in favour of increasing the kernel weights. To achieve this, we apply the Softmax non-linearity to ensure λ's non-negativity and normalisation $\int \lambda(h)dh = 1$:

$$\widehat{\lambda} = \mathrm{FT}\left(\sigma\left(\mathrm{IFT}\left(\widehat{\lambda}' \right) \right) \right), \tag{16}$$

where $\widehat{\lambda}$ is the stack of Fourier coefficients of the density λ, $\widehat{\lambda}'$ the stack of learnable Fourier coefficients of its logits, and $\sigma(\cdot)$ is the Softmax non-linearity. As such, the following is computed for each h in a discrete sampling set $\mathcal{H} \subset H$ of N elements, using $z_\lambda = \frac{1}{N} \sum_{n=1}^N e^{\lambda'(n) - \max(\lambda')}$[7]:

$$\lambda(h) = \sigma(\lambda'(h)) = \frac{e^{\lambda'(h) - \max(\lambda')}}{z_\lambda}. \tag{17}$$

While other normalisation methods are also valid [8], we opted for Softmax normalisation as the z-term and logits can be used directly to compute the KL-divergence described later in this section.

---

[6]In practice, only the Fourier coefficients of irreps appearing in one of the Clebsch-Gordan decompositions need to be considered.

[7]We subtract the max value to make softmax numerically stable.

[8]E.g., the use Parseval's identity might be a straightforward method to normalise the Fourier coefficients without sampling.

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. We show in Apx. C.3 that it is possible for the model to freely shift the likelihood distribution by shifting the weights accordingly, diminishing the interpretability of the distributions.

To address this concern, we propose constraining the Fourier coefficients by aligning the maximum likelihood with the identity element $e \in H$. The theoretical justification lies in the intrinsic impossibility of violating equivariance when the identity element is applied. Hence, we desire:

$$\forall h \in H \colon \lambda(h) \le \lambda(e) \tag{18}$$

Since directly enforcing this constraint on the Fourier coefficients is hard, we opt for a soft alignment constraint:

$$D_{\text{align}}(\lambda) = \max(\lambda) - \lambda(e). \tag{19}$$

This error can be added as a loss term during training. Provided that the likelihood λ has been sampled using a sufficient number of samples N, this error term is equal to zero if Eq. 18 holds, and greater than zero otherwise.

While this regularisation is rather sparse, as only the largest likelihood contributes to the loss, we found that it is sufficient in ensuring proper alignment and therefore did not consider less sparse approaches.

Regaining Equivariance Once a layer breaks the equivariance, this equivariance cannot be accurately regained in subsequent layers. Since our approach allows independent parameterisations of the distributions in subsequent layers, the learnt distributions do not necessarily describe such a monotonically decreasing equivariance. To this end, we propose to employ KL-divergence as regularisation between the layers. As the goal is to ensure that layer $n + 1$ does not regain equivariance previously lost by layer n, we treat layer n as reference distribution for layer $n + 1$. To prevent layer $n + 1$ from influencing layer n, we detach the likelihood of layer n and treat it as constant (see Apx D.2 for more details on the potential advantage of this detaching, and on monotonically decreasing equivariance in general). Given the likelihoods λ_n and λ_{n+1}, we show in Apx. C.4 that KL-divergence can be computed using the z-terms and Fourier coefficients of the logits and normalised distribution:

$$\begin{aligned} D_{KL}(\lambda_{n+1} \,\|\, \lambda_n) &= \widehat{\boldsymbol{\lambda_{n+1}}}^\top \widehat{\boldsymbol{\lambda'_{n+1}}} - \max(\lambda'_{n+1}) \\ &- \log z_{\lambda_{n+1}} - \widehat{\boldsymbol{\lambda_{n+1}}}^\top \widehat{\boldsymbol{\lambda'_n}} + \max(\lambda'_n) + \log z_{\lambda_n}. \end{aligned} \tag{20}$$

3.3. Bandlimiting

While our approach allows for a flexible and interpretable degree of equivariance, it also significantly increases the

number of learnable parameters. In the equivariant setting in Eq. 11, for each irrep ψ_j from the steerable basis, the kernel is parameterised by $[0(jj)]$ single weights $W_{j,k,s,r}$. Instead, our projection in Eq. 13 is parameterised by a vector of size $d_j d_{j'} = \sum_i [i(jj')] \cdot d_i$ for each pair of ψ_j and $\psi_{j'}$.

Recall the composition of the partially equivariant projection matrix $c^{jj'}$ in Eq. 14. The size of the matrix is dependant on the number of irreps ψ_i appearing in the tensor product $\psi_j \otimes \psi_{j'}$. For example, if we consider the group $SO(3)$, whose irreps are indexed by their rotational frequency, $\psi_j \otimes \psi_{j'}$ contains all irreps ψ_i with frequencies in the range $i \in |j - j'|, \dots, j + j'$. As such, we propose to reduce the number of parameters through a tunable bandlimit L on the maximum frequency of irreps ψ_i, reducing the parameter space and the flexibility of the distributions.

We note that band-limiting is *not strictly required* to obtain a reasonable number of parameters compared to other partially equivariant approaches. Instead, it is an *optional tool* to reduce computational complexity and filter higher frequencies as a form of regularisation (see Apx. G.5.3)

3.4. Implementation

Our implementation[9] relies on the `escnn` library (Cesa et al., 2022), providing an intuitive environment for configuring SCNNs with compact groups H and obtaining the corresponding H-steerable basis $\mathcal{B}$. For our solution, we adapt Algorithm 1 in (Cesa et al., 2022), which we present in Algorithm 1. An important consideration is that the `escnn` library is optimised for equivariant SCNNs. As such, our implementation on top of this library is likely sub-optimal. E.g., unlike in regular SCNNs acting on a static space X, our current implementation resamples the steerable basis after each forward pass, which may not be necessary.

4. Related Work

Group CNNs (GCNNs) (Cohen & Welling, 2016a), extend the application of the regular convolution operator to the domain of a group to obtain group equivariance using discrete groups. Steerable CNNs (SCNNs) (Cohen & Welling, 2016b) provide a more general theory of equivariance, offering more flexibility, the ability to model continuous groups and some computational advantages for large groups. Previous works proposed other strategies to construct equivariance, see for example (Worrall et al., 2017; Kondor & Trivedi, 2018; Bekkers, 2019; Finzi et al., 2020)

Several methods have recently been proposed to obtain varying degrees of equivariance in equivariant NNs. Weiler &

---

[9]We will publish the implementation at `https://github.com/QUVA-Lab/partial-escnn` and integrate it into the `escnn` library `https://github.com/QUVA-Lab/escnn`.

Algorithm 1 Generate a learnable equivariant H-steerable basis on space X

Require: $\rho_{\text{in}} = \psi_l$ and $\rho_{\text{out}} = \psi_J$, H-steerable basis $\mathcal{B} = \left\{Y_j^k\right\}_j^k$, $\widehat{H} = \{\psi_j\}_j$, $x \in X, L \in \mathbb{N}^+$, stack of logit coefficients $\widehat{\lambda'}$.

1: $\text{CG}^{lJ}, \{[j'(Jl)]\}_{j'} \leftarrow \text{decompose}(\psi_l \otimes \psi_J)$ Eq. 4 (Thm. B.37)

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

3:     **for all** $j' \in \widehat{H} : [j'(Jl)] > 0, s \le [j'(Jl)]$ **do**

4:        $\left\{\widehat{\lambda}(\overline{\psi}_i)\right\}_i \leftarrow \text{unstack}\left(\text{FT}\left(\sigma\left(\text{IFT}\left(\widehat{\lambda'}\right)\right)\right)\right)$ Eq. 17

5:        $Q, \{[i(jj')]\}_i \leftarrow \text{decompose}(\psi_j \otimes \psi_{j'})$ Eq. 7 (Thm. B.37)

6:        $Q \leftarrow \text{bandlimit}(Q, L)$ Sec. 3.3

7:        $c^{jj'} \leftarrow Q \cdot \text{diagonal}\left(\text{repeat}\left(\left\{\widehat{\lambda}(\overline{\psi}_i)\right\}_i\right)\right)$ Repeats $[i(jj')]$ times each $\mathcal{R}_{\psi_i}\left(\widehat{\lambda}(\overline{\psi}_i)\right)$ and diagonalises result (Eq. 14)

8:        $K_{jkj's}(x) \leftarrow \text{unvec}\left(\left[\text{CG}_s^{j'(Jl)}\right]^\top \cdot c^{jj'} \cdot Y_j^k(x)\right)$ Eq. 13

Cesa (2019) offer a manual solution via group restrictions. Other works allow the degree of equivariance to be learnt by the network. Partial GCNNs (Romero & Lohit, 2022) parameterise a learnable probability distribution over the elements of the subgroup H of G. Rather than uniformly sampling group elements as in traditional GCNNs, Partial GCNNs use this learnt probability distribution to guide the sampling of group elements, achieving a form of partial H-equivariance as a result. Non-stationary GCNNs (van der Ouderaa et al., 2022) utilised non-stationary kernels to achieve partial G-equivariance. Proposed by Finzi et al. (2021), Residual Pathway Priors (RPP) offer a more versatile strategy that can be applied across a variety of equivariant networks. In RPPs, each intermediate feature field comprises (at least) two layers: one that maintains G-equivariance and another with softer constraints, with a predetermined prior controlling the contribution. Based on non-stationary GCNNs and RPPs, van der Ouderaa et al. (2023) took a more unified approach and employed Bayesian model selection using differentiable Laplace approximations. They learnt layer-wise equivariances as hyper-parameters, offering a framework for discovering symmetries at different scales. Dehmamy et al. (2021) define group convolution in terms of the Lie-algebra: by learning the algebra basis, their model is able to learn the required symmetries. Wang et al. (2022a) use lift expansion to achieve partial equivariance for Q learning by concatenating encoded non-equivariant features to encoded equivariant features in an encoder-decoder setting. Wang et al. (2022b) introduce RSteer and RGroup, SCNN and GCNN based approaches that break equivariance through a regularised function mapping between group element and partially equivariant weights. Finally, Maile et al. (2023) propose an equivariance-aware neural architecture search method to discover equivariance during training.

To learn the equivariance with respect to a specific subgroup S of H, RPPs require a residual connection for each of these subgroups. This is computationally infeasible, as it would require an infinite number of residual connections for continuous groups. Furthermore, enumerating such subgroups is complicated for 3D rotation groups. Specifically, the collection of rotations around any arbitrary 3D axis constitutes a subgroup of $SO(3)$. However, sampling uniformly over the space of 3D axes is not straightforward and requires a large number of samples and, therefore, additional residual layers. Thus, RPPs are unfeasible for learning equivariance to an unknown subgroup S of a large or continuous group H.

Like RPPs, our approach models partial equivariance as a combination of equivariant and non-equivariant mappings. Indeed, RPP is comparable to our method under a *specific and non-learnable* choice of distribution λ. Furthermore, similar to Partial GCNNs, our approach uses a learnt likelihood over the group as a sampling mechanism. Whereas Partial GCNNs directly sample this likelihood to perform group convolutions, our solution implicitly integrates the learnt likelihood within the SCNN framework. Through this likelihood, we can model equivariance with respect to any subgroup S of H, or even subsets S that are not proper groups, without additional layers. See Apx. D.1 for an in-depth comparison between these approaches.

5. Experiments and Evaluation

We explore the capabilities of our approach in terms of overall performance, generalisability, data-efficiency and interpretability on 2D and 3D classification tasks using various groups $H \le O(n)$ (See Apx. F.2). In our 2D experiments we use DDMNIST, our adapted double-digit version of MNIST (LeCun et al., 1998). Here, the individual digits in each sample can be randomly and independently transformed during training according to a chosen subgroup; see Apx. F.1.2. For our 3D experiments we use the Organ, Synapse and Nodule subsets from the biomedical collection MedMNIST (Yang et al., 2023), as these datasets vary in their symmetries. In these experiments, we use 7-layer CNNs, SCNNs, SCNNs with a *group restriction* to no equivariance in the last two layers, RPPs and our Partial SCNNs (PSCNNs). Each (partially) equivariant model outputs *trivial* irreps in the final convolution layer to obtain

Network Group	Partial Equivariance	Symmetries for individual digits					
		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	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	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	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	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	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	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	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	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	0.935 (0.008)	0.890 (0.005)	0.870 (0.011)	0.899 (0.004)	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	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	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	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 1. Test accuracies on various `DDMNIST` data symmetries. The first column indicates the equivariance group, where CNN is a regular CNN. The second column indicates the method used to break the equivariance. For each symmetry, the highest accuracy for each network group is underlined, with the highest overall accuracy in **bold**. Standard deviations over 5 runs are in parentheses.

structural invariance. See Apx. F.2 for more details.

To provide a broader comparison between our approach and alternative approaches, we perform experiments on the `JetFlow` and `Smoke` 2D simulation datasets with partial rotational symmetries as employed by Wang et al. (2023), where they compare their approach to a collection of partial, fully and non-equivariant approaches. For our experiments we simply port their best RSteer configuration as reported by their repository, to our approach without additional fine-tuning. See (Wang et al., 2023) and Apx. F.1.4 & F.2 for more details on the setup and alternative approaches.

All our models are trained with the following objective:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \alpha_{\text{align}}\mathcal{L}_{\text{align}} + \alpha_{\text{KL}}\mathcal{L}_{\text{KL}}. \quad (21)$$

$\mathcal{L}_{\text{KL}}$ and $\mathcal{L}_{\text{align}}$ are the mean average KL-divergence between consecutive layer pairs, and alignment loss for each layer, with tunable weights α_{KL} and α_{align}. Apx. G provides additional experiments and *ablation studies* showing that these two terms are necessary for interpretable likelihoods.

5.1. Benchmarks

DDMNIST Tab. 1 contains the classification accuracies for `DDMNIST`. For all symmetries and choices of H, non-equivariant CNNs outperform fully H-invariant SCNNs. The performance gap is relatively small for C_1 symmetries, but significantly larger for other symmetries. However, employing any of the partially equivariant methods results in improved performance compared to the regular CNN for all symmetries except C_1. The confusion matrices in Fig. 2a show that the invariant $O(2)$ SCNN is unable to distinguish between digit reversals (e.g. 37 and 73). It seems that our $O(2)$ PSCNN manages to eliminate these confusions by breaking equivariance. Furthermore, our approach,

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	**0.873** (0.002)	0.738 (0.009)	0.607 (0.006)
	RPP	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	0.868 (0.009)	0.743 (0.004)	0.592 (0.008)
	RPP	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 2. Test accuracies on `MedMNIST` datasets.

particularly the $SO(2)$ variant, outperforms the *restricted* SCNN and RPP methods across each data symmetry in most cases. Finally, taking a group that is too large –e.g. taking $H = O(2)$ rather than $SO(2)$ for D_1 data symmetries– results in a higher drop in performance for the fully equivariant SCNN than for the partially equivariant approaches.
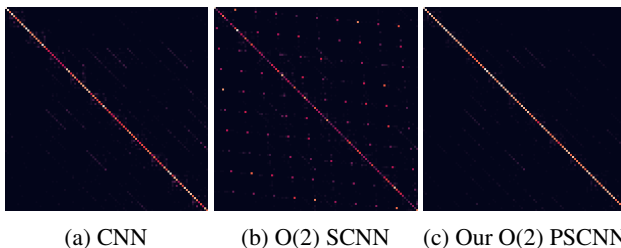

| (a) CNN | (b) O(2) SCNN | (c) Our O(2) PSCNN |

Figure 2. Confusion matrices for `DDMNIST` with $O(2)$ symmetries. Labelled 0-99 from top to bottom and left to right.

MedMNIST Tab. 2 contains the classification accuracies for `MedMNIST`. On **Nodule**, the performance differences are negligible between CNNs, SCNNs and our PSCNNs, with RPPs lagging behind. On **Synapse** the performance differences are larger, with PSCNNs significantly outperforming all models, while regular SCNNs outperform both CNNs and RPPs. Finally, on **Organ** the invariant SCNN is

significantly outperformed by all other models, with RPPs achieving the highest performance. In Apx. G.2 we show that the poor SCNN performance is largely due to confusions between left and right versions of some organs. Overall, these results suggest that **Nodule** is highly symmetric, **Synapse** is fairly symmetric –while still containing relevant non-invariant features– and **Organ** is highly non-symmetric.

Smoke and JetFlow Tab. 3 presents parameter counts and test RMSE scores for `JetFlow` and `Smoke` using two methodologies; *Future* denotes testing on the same simulation as the trainset, but in different time-steps in the simulation while *domain* corresponds with testing at the same time-step but in different simulations. All equivariant models use $C4$-equivariance, requiring no band-limiting for our approach (See Apx. G.5.3 for band-limiting experiments). Our approach uses fewer parameters than MLP and the partially equivariant RSteer, Lift, and RPP approaches, with more parameters compared to CNN, Combo, RGroup and e2cnn. Furthermore, our approach outperforms all other approaches on the `Smoke` dataset, particularly using the *domain* testset. On `JetFlow` our approach performs on-par with RSteer, while outperforming Lift and e2cnn.

	Smoke			**JetFlow**		
Model	Future	Domain	Params (M)	Future	Domain	Params (k)
MLP	1.38 (0.06)	1.34 (0.03)	8.33*	-	-	510*
CNN	1.21 (0.01)	1.10 (0.05)	0.25*	-	-	10*
e2cnn	1.05 (0.06)	0.76 (0.02)	0.62*	0.21 (0.02)	0.27 (0.03)	21*
RPP	0.96 (0.10)	0.82 (0.01)	4.36*	0.16* (0.01)	0.19* (0.01)	145*
Combo	1.07 (0.00)	0.82 (0.02)	0.53*	-	-	19*
CLCNN	0.96 (0.05)	0.84 (0.10)	-	-	-	-
Lift	0.82 (0.01)	0.73 (0.02)	3.32*	0.18 (0.02)	0.21 (0.04)	479*
RGroup	0.82 (0.01)	0.73 (0.02)	1.88*	-	-	63*
RSteer	0.80 (0.00)	0.67 (0.01)	5.60*	0.17 (0.01)	**0.16** (0.01)	185*
Ours	**0.77** (0.01)	**0.57** (0.00)	3.12	**0.15** (0.00)	0.17 (0.01)	105

Table 3. RMSE *future* and *domain* test scores on `Smoke` and `JetFlow` datasets. Adapted from Tab. 1 and 2 from (Wang et al., 2022b). * Indicates our replication attempt, see Apx. F.2.

5.2. Inspecting the Learnt Likelihood Distributions

Following sections are best viewed in colour. To differentiate between elements in $O(2)$ we use colour coding: blue signifies a basic rotation, and red a reflection and rotation.

Fig. 3 shows the resulting likelihoods and corresponding equivariance error[10] for layers 4 and 5 of our $O(2)$ PSCNN trained on `DDMNIST` with $O(2)$ and C_1 symmetries. Firstly, we observe that a lower likelihood results in a higher equivariance error. Next, in layer 4 equivariance is maintained for π for $O(2)$ symmetries, but not C_1 symmetries. Most notably, on $O(2)$ symmetries a high likelihood appears at π in layer 5. As this is one of the latest layers in the model, this corresponds to a global transformation. Indeed, a global rotation and reflection effectively swap the digits twice, so

---

[10]The error is computed as the difference between the error vector norms between our model and a fully invariant $O(2)$-SCNN.
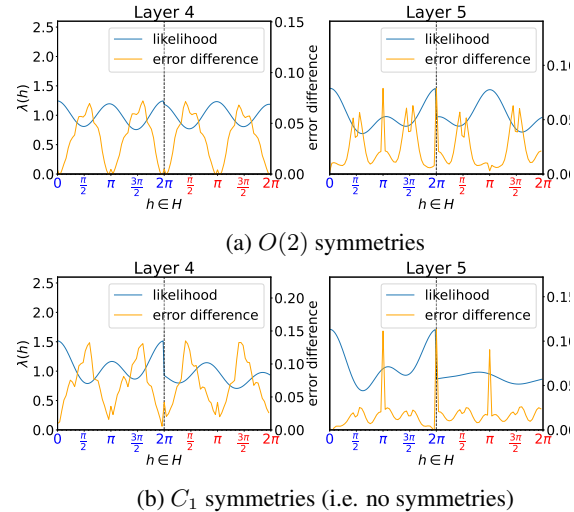

(a) $O(2)$ symmetries

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

Figure 3. Learnt likelihood λ and error difference for layers 4 and 5 of our $O(2)$ PSCNN trained on `DDMNIST` with $O(2)$ and C_1 symmetries. Dotted line marks the $O(2)$ reflection domain transition. Note the scale of the error between the plots.

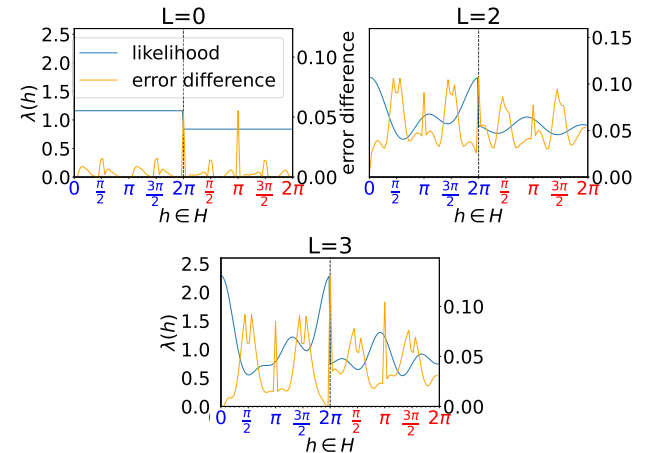this is an example of correct global equivariance.

Figure 4. Likelihoods and errors of the fifth $O(2)$ PSCNN layer trained on $SO(2)$ `DDMNIST` under various bandlimits L.

5.3. Investigating the Effect of Bandlimiting

Tab. 4 shows the classification accuracies on `DDMNIST` of our $O(2)$ PSCNN with various levels of bandlimitting compared to an invariant SCNN. Here, we observe that while all PSCNN models outperform the SCNN models, higher values for L achieve higher results. However, there seems to be diminishing returns on this dataset for $L > 2$. Notably, a bandlimiting of $L = 0$ already yields a significant performance improvement compared to the SCNN on $SO(2)$ symmetries, as this allows the model to reduce equivariance for reflections that do not appear in $SO(2)$. Fig. 4 shows the likelihoods for the fifth layer of our $O(2)$ PSCNN trained

on $SO(2)$ symmetries using $L \in \{0, 2, 3\}$. Apx. G.5.3 contains more band-limiting results, including the effects on the parameter space, showing that band-limiting is not necessarily required to obtain a reasonable number of parameters.

		Bandlimit L			
G	**SCNN**	**0**	**1**	**2**	**3**
$SO(2)$	0.391 (0.012)	0.469 (0.010)	<u>0.894</u> (0.011)	<u>0.894</u> (0.004)	0.889 (0.013)
$O(2)$	0.399 (0.014)	0.402 (0.003)	0.780 (0.009)	<u>0.819</u> (0.010)	0.817 (0.007)

Table 4. G-DDMNIST test accuracies using our $O(2)$ PSCNN with various bandlimits L and SCNN without bandlimiting.

5.4. 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 **Organ** and highly symmetric **Nodule** datasets. See Fig. 5 for the results. On the symmetric **Nodule** subset our approach consistently achieves the highest performance, particularly on smaller train sets. Conversely, on the non-symmetric **Organ** dataset both RPP and CNN outperform our PSCNN, especially on smaller train sets. This is likely due to the fact that RPPs and CNNs, unlike our approach, do not initialise as equivariant due to the regular convolutions. When little data is available our approach therefore struggles to break equivariance. On highly symmetric datasets, the reverse applies.
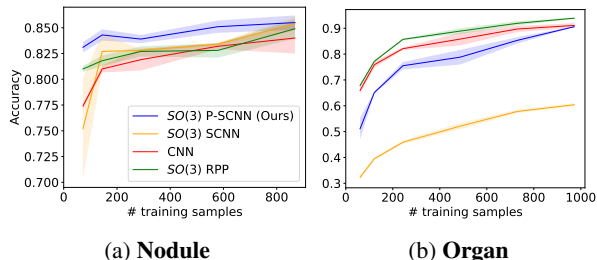

(a) **Nodule** (b) **Organ**

Figure 5. Data ablation study on **Organ** and **Nodule**.

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

Table 5. Generalisability results for models trained on discrete D_4 symmetries. All models are evaluated on D_4 and $O(2)$ symmetries. Δ shows the difference in performance between the two cases.

5.5. Generalisability

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 generalisability we train the models on DDMNIST with discrete D_4 symmetries and compare their D_4 and $O(2)$ performance (Tab. 5). Our models consistently outperform RPPs, both in terms of performance and performance regression. However, *restricted* SCNN shows a smaller performance regression, likely due to the fixed equivariance.

6. Limitations and Future Work

Although our approach offers an interpretable solution to learn the degree of equivariance in SCNNs, it does have a few limitations. Most notable, our approach is limited in breaking equivariances for compact groups, and thus cannot model partial translation equivariance. Furthermore, compared to regular SCNNs, our approach uses significantly more parameters, even with band-limiting. Moreover, the steerable basis needs to be recomputed after each update, unlike regular SCNNs applied fixed grids, increasing computational cost. Besides, our approach requires an alignment and, optional, KL-divergence term, resulting in additional hyper-parameter tuning. Nonetheless, we have observed our approach to be fairly robust to the choice of these terms.

While we include a comparison of parameter space, due to the current sub-optimal implementation we did not perform a more detailed computational analysis. Similarly, we do not provide a theoretical proof that the learnt equivariance corresponds to the true equivariance. Both would be interesting directions for future research. Furthermore, we have only considered *reducing* the degree of equivariance during training by initialising a uniform likelihood, however, it would be interesting to initialise a non-uniform likelihood distribution as a prior. Finally, we only consider band-limiting to filter high frequencies, but it could also be employed to ensure a lower-limit to the learnable equivariance.

7. Conclusion

This work presents a solution to generate partially H-steerable kernels from a likelihood distribution over H and proposes an algorithm to parameterise this likelihood distribution in terms of its Fourier coefficient within the flexible framework of SCNNs. Furthermore, through the employment of normalisation, bandlimiting, KL-divergence, and our alignment loss, we ensure an interpretable likelihood distribution with user-tunable flexibility describing the degree of equivariance. Our experiments show that such a learnable degree of equivariance can result in performance improvements compared to fully equivariant approaches. Finally, in addition to theoretical benefits, our approach shows competitive performance compared to alternative partially equivariant approaches as well as *restricted* SCNNs in task-specific performance, data-efficiency and generalisation.

Acknowledgements

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Armato III, S. G., McLennan, G., Bidaut, L., McNitt-Gray, M. F., Meyer, C. R., Reeves, A. P., Zhao, B., Aberle, D. R., Henschke, C. I., Hoffman, E. A., 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. 41

Bekkers, E. J. B-spline CNNs on lie groups. *arXiv preprint arXiv:1909.12057*, 2019. 5

Bekkers, E. J., Vadgama, S., Hesselink, R. D., van der Linden, P. A., and Romero, D. W. Fast, expressive SE (n) equivariant networks through weight-sharing in position-orientation space. *arXiv preprint arXiv:2310.02970*, 2023. 37

Bilic, P., Christ, P., Li, H. B., Vorontsov, E., Ben-Cohen, A., Kaissis, G., Szeskin, A., Jacobs, C., Mamani, G. E. H., Chartrand, G., et al. The liver tumor segmentation benchmark (lits). *Medical Image Analysis*, 84:102680, 2023. 40

Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021. 1

Callister, M., Baldwin, D., Akram, A., Barnard, S., Cane, P., Draffan, J., Franks, K., Gleeson, F., Graham, R., Malhotra, P., et al. British thoracic society guidelines for the investigation and management of pulmonary nodules: accredited by nice. *Thorax*, 70(Suppl 2):ii1–ii54, 2015. 41

Cesa, G., Lang, L., and Weiler, M. A program to build E(n)-equivariant steerable CNNs. In *International Conference on Learning Representations*, 2022. 2, 3, 5, 28, 29, 30, 33, 37, 38, 42

Cohen, T. and Welling, M. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999. PMLR, 2016a. 1, 5

Cohen, T. S. and Welling, M. Steerable CNNs. *arXiv preprint arXiv:1612.08498*, 2016b. 1, 5

Dehmamy, N., Walters, R., Liu, Y., Wang, D., and Yu, R. Automatic symmetry discovery with lie algebra convolutional network. *Advances in Neural Information Processing Systems*, 34:2503–2515, 2021. 6

Finzi, M., Stanton, S., Izmailov, P., and Wilson, A. G. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *International Conference on Machine Learning*, pp. 3165–3176. PMLR, 2020. 5

Finzi, M., Benton, G., and Wilson, A. G. Residual pathway priors for soft equivariance constraints. *Advances in Neural Information Processing Systems*, 34:30037–30049, 2021. 1, 2, 6, 36, 42, 43, 45, 46

Franzen, D. and Wand, M. General nonlinearities in SO(2)-equivariant cnns. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 9086–9098. Curran Associates, Inc., 2021. 37

Gould, M. K., Donington, J., Lynch, W. R., Mazzone, P. J., Midthun, D. E., Naidich, D. P., and Wiener, R. S. 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. 41

Gray, E. G. Axo-somatic and axo-dendritic synapses of the cerebral cortex: an electron microscope study. *Journal of anatomy*, 93(Pt 4):420, 1959. 40

Holl, P., Koltun, V., Um, K., and Thuerey, N. phiflow: A differentiable pde solving framework for deep learning via physical simulations. In *NeurIPS workshop*, volume 2, 2020. 41

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 46

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 34

Klemann, C. J. and Roubos, E. W. The gray area between synapse structure and function—gray's synapse types i and ii revisited. *Synapse*, 65(11):1222–1230, 2011. 40

Kondor, R. and Trivedi, S. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *International Conference on Machine Learning*, pp. 2747–2755. PMLR, 2018. 5

Kuipers, T. P. and Bekkers, E. J. Regular SE(3) group convolutions for volumetric medical image analysis. *arXiv preprint arXiv:2306.13960*, 2023. 40

Kullback, S. and Leibler, R. A. On information and sufficiency. *The annals of mathematical statistics*, 22(1): 79–86, 1951. 2, 34

Lang, L. and Weiler, M. A wigner-eckart theorem for group equivariant convolution kernels. *arXiv preprint arXiv:2010.10952*, 2020. 29

Larici, A. R., Farchione, A., Franchi, P., Ciliberto, M., Cicchetti, G., Calandriello, L., Del Ciello, A., and Bonomo, L. Lung nodules: size still matters. *European respiratory review*, 26(146), 2017. 41

LeCun, Y., Bengio, Y., et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995. 1, 45

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 6, 40

MacMahon, H., Austin, J. H., Gamsu, G., Herold, C. J., Jett, J. R., Naidich, D. P., Patz Jr, E. F., and Swensen, S. J. Guidelines for management of small pulmonary nodules detected on ct scans: a statement from the fleischner society. *Radiology*, 237(2):395–400, 2005. 41

Maile, K., Wilson, D. G., and Forré, P. Equivariance-aware architectural optimization of neural networks. In *The Eleventh International Conference on Learning Representations*, 2023. 6

Peter, F. and Weyl, H. Die vollständigkeit der primitiven darstellungen einer geschlossenen kontinuierlichen gruppe. *Mathematische Annalen*, 97(1):737–755, 1927. 3

Petrache, M. and Trivedi, S. Approximation-generalization trade-offs under (approximate) group equivariance. *arXiv preprint arXiv:2305.17592*, 2023. 1

Poulenard, A. and Guibas, L. J. A functional approach to rotation equivariant non-linearities for tensor field networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13174–13183, 2021. 37

Romero, D. W. and Lohit, S. Learning partial equivariances from data. *Advances in Neural Information Processing Systems*, 35:36466–36478, 2022. 1, 6, 35, 36

Sheng, M. and Kim, E. The postsynaptic organization of synapses. *Cold Spring Harbor perspectives in biology*, 3 (12):a005678, 2011. 41

Somogyi, P., Tamas, G., Lujan, R., and Buhl, E. H. Salient features of synaptic organisation in the cerebral cortex. *Brain research reviews*, 26(2-3):113–135, 1998. 40

van der Ouderaa, T., Romero, D. W., and van der Wilk, M. Relaxing equivariance constraints with non-stationary continuous filters. *Advances in Neural Information Processing Systems*, 35:33818–33830, 2022. 1, 6, 35

van der Ouderaa, T. F., Immer, A., and van der Wilk, M. Learning layer-wise equivariances automatically using gradients. *arXiv preprint arXiv:2310.06131*, 2023. 6

Van der Pol, E., Worrall, D., van Hoof, H., Oliehoek, F., and Welling, M. MDP homomorphic networks: Group symmetries in reinforcement learning. *Advances in Neural Information Processing Systems*, 33:4199–4210, 2020. 3

Wang, D., Walters, R., Zhu, X., and Platt, R. Equivariant q learning in spatial action spaces. In *Conference on Robot Learning*, pp. 1713–1723. PMLR, 2022a. 6, 45

Wang, D., Zhu, X., Park, J. Y., Platt, R., and Walters, R. A general theory of correct, incorrect, and extrinsic equivariance. *arXiv preprint arXiv:2303.04745*, 2023. 7

Wang, R., Walters, R., and Yu, R. Approximately equivariant networks for imperfectly symmetric dynamics. In *International Conference on Machine Learning*, pp. 23078–23091. PMLR, 2022b. 1, 2, 6, 8, 35, 36, 41, 42, 45, 59

Weiler, M. and Cesa, G. General E(2)-equivariant steerable CNNs. *Advances in neural information processing systems*, 32, 2019. 2, 3, 5, 30, 38, 45

Weiler, M., Geiger, M., Welling, M., Boomsma, W., and Cohen, T. S. 3D steerable CNNs: Learning rotationally equivariant features in volumetric data. *Advances in Neural Information Processing Systems*, 31, 2018a. 3, 38

Weiler, M., Hamprecht, F. A., and Storath, M. Learning steerable filters for rotation equivariant cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 849–858, 2018b. 38

Weiler, M., Forré, P., Verlinde, E., and Welling, M. Coordinate independent convolutional networks–isometry and gauge equivariant convolutions on riemannian manifolds. *arXiv preprint arXiv:2106.06020*, 2021. 1

Worrall, D. E., Garbin, S. J., Turmukhambetov, D., and Brostow, G. J. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE*

conference on computer vision and pattern recognition, pp. 5028–5037, 2017. 5, 38

Xu, X., Zhou, F., Liu, B., Fu, D., and Bai, X. Efficient multiple organ localization in ct image using 3D region proposal network. *IEEE transactions on medical imaging*, 38(8):1885–1898, 2019. 40

Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., and Ni, B. Medmnist v2-a large-scale lightweight benchmark for 2D and 3D biomedical image classification. *Scientific Data*, 10(1):41, 2023. 1, 6, 40

Table of Contents

APPENDIX

A. Overview of Appendix

These sections cover the supplementary materials of our work. First, in Appendix B we provide an extensive mathematical background. Next, Appendix C covers additional and more in-depth derivations from concepts and solutions in the paper. In Appendix E we provide additional and more in-depth implementation details. Appendix F contains an extensive overview of the experimental details. Finally, in Appendix G we present additional experiments.

B. Mathematical Preliminaries

In this section, we cover the mathematical preliminaries required for the work in this paper. We will start with the basics of Group Theory and Representation Theory, work our way through *equivariance* and *invariance* after which we close with *Peter Weyl Theorem* and *Fourier Transforms*.

B.1. Group Theory

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

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 B.2. *(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:*

$$GL(\mathbb{R}^n) = \left\{ O \in \mathbb{R}^{n \times n} \mid O^{-1}O = I_n \right\} \tag{22}$$

The identity element e is the identity matrix I_n of size $n \times n$. Since the matrix multiplication is associative, this definition satisfies the axioms from Definition B.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 B.3. *(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 B.4. *(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 B.5. *($\mathbb{Z}/n\mathbb{Z}$) The set $\{e = 0, 1, 2, \ldots, n - 1\}$ with the associative group law sum modulo n:*

$$+ : (a, b) \mapsto (a + b) \qquad \mod n \tag{23}$$

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 B.6. *(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 B.7. *(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 B.8. *(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 B.9. *(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 B.10. *(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:*

$$\forall x, y \in X, \exists g \in G \text{ s.t. } g \cdot x = y, \tag{24}$$

where $\cdot$ denotes the group action from Def B.9.

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 B.11. *(**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. B.5. A homomorphism f can be established between these two groups and can be defined as*

$$f : e^{ik\frac{2\pi}{n}} \mapsto k. \tag{25}$$

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

$$C_N = \left\{e = g^0, g, g^2, \ldots, g^{n-1} \mid g^m = g^n \iff m \equiv n \bmod N\right\}. \tag{26}$$

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 B.12. *(**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 $\Theta = \{\theta\}$:

$$\psi(\Theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \tag{27}$$

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

$$SO(n) = \left\{O \in \mathbb{R}^{n\times n} \mid O^\top O = \mathbf{I}_n \text{ and } \det(O) = 1\right\}. \tag{28}$$

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

Definition B.13. *(**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 B.14. *(**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 B.15. *(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. B.12, 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:

$$gH = gSO(2) = \left\{ O \in \mathbb{R}^{2 \times 2} \mid O^\top O = \mathbf{I}_n \text{ and } \det(O) = -1 \right\}. \tag{29}$$

Definition B.16. *(Normal Subgroup)* *Given a group G with a subgroup $H \leq G$, H is a normal subgroup of G iff:*

$$\forall g \in G: \quad gH = Hg \tag{30}$$

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 B.17. *(C_N as a Subgroup of $SO(2)$)* *In Examples B.12 and B.11 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. B.11, 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:

$$C_N \mapsto SO(2), \quad g^k \mapsto r_{k\frac{2\pi}{N}}. \tag{31}$$

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 B.18. *(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:*

$$(k_1, h_1) \cdot (k_2, h_2) = (k_1 * k_2, h_1 + h_2). \tag{32}$$

$K \times H$ *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 B.19. *(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 B.20. *(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:*

$$(n_1, h_1) \cdot (n_2, h_2) = (n_1 * \phi(h_1, n_2), h_1 + h_2) \tag{33}$$

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 B.21. *(Orthogonal Group $O(n)$) In Ex. B.12 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:*

$$\forall x \in X = \mathbb{R}^n : \quad r_\Theta, x \mapsto r_\Theta \cdot x = \psi(\Theta) x \tag{34}$$

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:

$$\forall x \in X = \mathbb{R}^n : \quad f_p, x \mapsto f_p \cdot x = \begin{bmatrix} p & 0 \\ 0 & \boldsymbol{I}_{n-1} \end{bmatrix} x \tag{35}$$

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

$$O(n) = \left\{ O \in \mathbb{R}^{n \times n} \mid O^\top O = \boldsymbol{I}_n \right\} \tag{36}$$

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:

$$(r_{\Theta_1}, f_{p_1}) \cdot (r_{\Theta_2}, f_{p_2}) = r_{\Theta_1} f_{p_1} r_{\Theta_2} f_{p_2} \tag{37}$$
$$= r_{\Theta_1} r_{\phi(f_{p_1})\Theta_2} f_{p_1} f_{p_2} \tag{38}$$
$$= (r_{\Theta_1} r_{\phi(f_{p_1})\Theta_2}, f_{p_1} f_{p_2}) \tag{39}$$
$$= (r_{\Theta_1} r_{\phi(f_{p_1})\Theta_2}, f_{p_1} f_{p_2}) \tag{40}$$

The action ϕ can be identified as:

$$\phi : C_2 \times SO(2) \to (r_{\Theta_1}, f_{p_2}), SO(2) \mapsto r_{\phi(f_{p_2})\Theta_1} \tag{41}$$

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

$$O(2) = C_2 \rtimes_\phi SO(2) \tag{42}$$

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 B.22. *(**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. B.12) 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:*

$$(t_{v_1}, r_{\Theta_1}) \cdot (t_{v_2}, r_{\Theta_2}) = t_{v_1} r_{\Theta_1} t_{v_2} r_{\Theta_2} \tag{43}$$

$$= t_{v_1} t_{\psi(\Theta_1) v_2} r_{\Theta_1} r_{\Theta_2} \tag{44}$$

$$= (t_{v_1} t_{\psi(\Theta_1) v_2}, r_{\Theta_1} r_{\Theta_2}) \tag{45}$$

$$= (t_{v_1 + \psi(\Theta_1) v_2}, r_{\Theta_1} r_{\Theta_2}) \tag{46}$$

The action ϕ can be identified as:

$$\phi : (\mathbb{R}^n, +) \times SO(n) \to (\mathbb{R}^n, +), (t_{v_2}, r_{\Theta_1}) \mapsto t_{\psi(\Theta_1) v_2} \tag{47}$$

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

$$SE(n) = (\mathbb{R}^n, +) \rtimes_\phi SO(n) \tag{48}$$

B.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 B.23. *(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*

$$\rho : G \to GL(V) \quad s.t. \quad \forall g_1, g_2 \in G\colon \quad \rho(g_1 g_2) = \rho(g_1)\rho(g_2) \tag{49}$$

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 B.24. *(Trivial Representation)* *The trivial representation $\rho : G \to GL(\mathbb{R})$ maps any group element to the identity. Therefore,*

$$\forall g \in G\colon \quad \rho(g) = 1. \tag{50}$$

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

Example B.25. *(Rotation Matrices)* *As shown in Example B.12, an example of a representation of the group $SO(2)$ are the two-dimensional rotation matrices:*

$$\psi : SO(2) \to GL(\mathbb{R}^2), r_\theta \mapsto \psi(r_\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{51}$$

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

Definition B.26. *(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:*

$$\forall g \in G\colon \quad \rho'(g) = Q\rho(g)Q^{-1} \tag{52}$$

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

$$\rho'(g_1)\rho'(g_2) = Q\rho(g_1)Q^{-1}Q\rho(g_2)Q^{-1} = Q\rho(g_1)\rho(g_2)Q^{-1} \tag{53}$$

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

Definition B.27. *(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:*

$$(\rho_1 \oplus \rho_2)(g) = \begin{bmatrix} \rho_1(g) & 0 \\ 0 & \rho_2(g) \end{bmatrix}. \tag{54}$$

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.:

$$\bigoplus_i \rho_i(g) = \rho_1(g) \oplus \rho_2(g) \oplus \rho_3(g) \oplus \dots. \tag{55}$$

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:

$$\rho(g) = \begin{bmatrix} \rho_1(g) & 0 \\ 0 & \rho_2(g) \end{bmatrix}, \tag{56}$$

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 B.28. *(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. B.24) 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 B.29. *(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. B.12) 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. B.12.

Orthogonal Group $O(2)$ The Orthogonal group (Ex. B.21) 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. B.11) 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:* $\psi_k^{SO(2)}(r_\theta) = \begin{bmatrix} \cos(k\theta) & -\sin(k\theta) \\ \sin(k\theta) & \cos(k\theta) \end{bmatrix} = \psi(k\theta), \quad k \in \left\{1, \ldots, \lfloor \frac{N-1}{2} \rfloor\right\}$

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

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

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:* $\psi_{1,k}^{D_N}(r_\theta p) = \begin{bmatrix} \cos(k\theta) & -\sin(k\theta) \\ \sin(k\theta) & \cos(k\theta) \end{bmatrix} \begin{bmatrix} f & 0 \\ 0 & 1 \end{bmatrix} = \psi(k\theta) f_p,$

$$k \in \left\{1, \ldots, \lfloor \tfrac{N-1}{2} \rfloor\right\}, p \in \{-1, 1\}$$

And the N-even irreps:

- *N-even irrep:* $\psi_{0,N/2}^{SO(2)}(r_\theta p) = \cos\left(\tfrac{N}{2}\theta\right) = \pm 1$

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

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 B.30. *(Irreps Decomposition (Peter-Weyl theorem part 1)) Any unitary (or orthogonal) representation $\rho : G \to V$ of a compact group 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*

$$\rho(g) = Q \left[\bigoplus_{i \in I} \psi_i(g) \right] Q^{-1}, \tag{57}$$

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 $\psi_j \in \widehat{G}$:

$$\rho(g) = Q \left[\bigoplus_{j \in \widehat{G}} \bigoplus_{s}^{[j]} \psi_j(g) \right] Q^{-1}, \tag{58}$$

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

For proofs, it is often only required to consider the irreps. A more specific method of irreps decomposition will be introduced in Section B.2.1.

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

Definition B.31. *(Regular Representation) Let G be a finite group, and let $\mathbb{R}^{|G|}$ be a vector space. Each basis vector $\boldsymbol{e}_g$ of $\mathbb{R}^{|G|}$ is associated with an element $g \in G$.*

The regular representation of G, denoted as $\rho_{reg}^G : G \to GL(\mathbb{R}^{|G|})$, is defined as the map that takes an element $\widetilde{g} \in G$ and associates it with a permutation matrix $\rho_{reg}^G(\widetilde{g})$ in $GL(\mathbb{R}^{|G|})$.

The action of $\rho_{reg}^G(\widetilde{g})$ on a basis vector $\boldsymbol{e}_g$ is defined as:

$$\rho_{reg}^{G}(\widetilde{g}) \cdot \boldsymbol{e}_g = \boldsymbol{e}_{\widetilde{g}g}. \tag{59}$$

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

Example B.32. *(**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
$\rho_{reg}^{D_2}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

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 B.33. *(**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:*

$$(\mathscr{L}_g f)(h) = f(g^{-1}h) \quad \forall h, g \in G, \tag{60}$$

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

$$(\mathscr{R}_g f)(h) = f(hg) \quad \forall h, g \in G, \tag{61}$$

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 B.34. *(**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:*

$$Res_H^G(\rho) : H \to GL(\mathbb{R}^n), \quad h \mapsto \rho|_H(h) \tag{62}$$

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. B.30).

B.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 B.35. *(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:*

$$(\rho_1 \otimes \rho_2)(g) = \rho_1(g) \otimes \rho_2(g) \tag{63}$$

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 B.36. *(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:*

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{bmatrix} \tag{64}$$

or, more precisely:

$$A \otimes B = \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & \dots & a_{11}b_{1q} & \dots & \dots & a_{1n}b_{11} & a_{1n}b_{12} & \dots & a_{1n}b_{1q} \\ a_{11}b_{21} & a_{11}b_{22} & \dots & a_{11}b_{2q} & \dots & \dots & a_{1n}b_{21} & a_{1n}b_{22} & \dots & a_{1n}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & a_{11}b_{p2} & \dots & a_{11}b_{pq} & \dots & \dots & a_{1n}b_{p1} & a_{1n}b_{p2} & \dots & a_{1n}b_{pq} \\ \vdots & \vdots & & \vdots & \ddots & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & \ddots & \vdots & \vdots & & \vdots \\ a_{m1}b_{11} & a_{m1}b_{12} & \dots & a_{m1}b_{1q} & \dots & \dots & a_{mn}b_{11} & a_{mn}b_{12} & \dots & a_{mn}b_{1q} \\ a_{m1}b_{21} & a_{m1}b_{22} & \dots & a_{m1}b_{2q} & \dots & \dots & a_{mn}b_{21} & a_{mn}b_{22} & \dots & a_{mn}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{p1} & a_{m1}b_{p2} & \dots & a_{m1}b_{pq} & \dots & \dots & a_{mn}b_{p1} & a_{mn}b_{p2} & \dots & a_{mn}b_{pq} \end{bmatrix} \tag{65}$$

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 B.37. *(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 B.30, 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:*

$$(\rho_l \otimes \rho_J)(g) = \left[C^{lJ} \right]^\top \left(\bigoplus_j \overset{[j(Jl)]}{\underset{s}{\bigoplus}} \psi_j(g) \right) C^{lJ} \tag{66}$$

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:

$$(\rho_l \otimes \rho_J)(g) = \underbrace{\begin{bmatrix} [C_{j_1}^{lJ}]^\top \\ [C_{j_2}^{lJ}]^\top \\ \vdots \\ [C_{j_n}^{lJ}]^\top \end{bmatrix}}_{[C^{lJ}]^\top} \underbrace{\begin{bmatrix} \psi_{j_1}(g) & 0 & \cdots & 0 \\ 0 & \psi_{j_2}(g) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \psi_{j_n}(g) \end{bmatrix}}_{\left(\bigoplus_j \bigoplus_s^{[j(Jl)]} \psi_j(g)\right)} \underbrace{\begin{bmatrix} C_{j_1}^{lJ} & C_{j_2}^{lJ} & \cdots & C_{j_n}^{lJ} \end{bmatrix}}_{C^{lJ}} \tag{67}$$

Here, $C_{j_1}^{lJ}$ through $C_{j_n}^{lJ}$ 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:

$$(\rho_l \otimes \rho_J)(g) = \sum_j \sum_s^{[j(Jl)]} \left[\mathrm{CG}_s^{j(Jl)}\right]^\top \psi_j(g)\,\mathrm{CG}_s^{j(Jl)} \tag{68}$$

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.

B.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 B.38. *(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*

$$\forall x \in X, \forall g \in G\colon \quad f(g \cdot x) = g \cdot f(x) \tag{69}$$

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 B.39. *(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*

$$\forall x \in X, \forall g \in G\colon \quad f(g \cdot x) = f(x). \tag{70}$$

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

$$\forall y \in Y, \forall g \in G : \quad g \cdot y = y. \tag{71}$$

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 B.40. *(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:*

$$\forall \boldsymbol{v} \in V_1, \forall g \in G\colon \quad W\rho_1(g)\boldsymbol{v} = \rho_2(g)W\boldsymbol{v} \tag{72}$$

from which follows that:

$$\forall g \in G\colon \quad W\rho_1(g) = \rho_2(g)W. \tag{73}$$

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 B.41. *(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 $\forall g \in G\colon \quad \psi_2(g)W = W\psi_1(g)$. 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. B.26) where W is the change of basis (Q from Def. B.26) between ψ_1 and ψ_2.*

B.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. B.29 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 B.42. *(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. B.30).*

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. B.33):

$$f \mapsto \rho(g) : [\mathscr{L}_g f](h) = f(g^{-1}h) \tag{74}$$

This representation is unitary, and therefore decomposes according to Thm. B.30. 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 $\left\{ \sqrt{d_\psi} [\psi(g)]_{ij} \mid \psi \in \widehat{G}, j \leq d_\psi \right\}$.

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. B.29). Here, the second column of an irrep $\psi_k^{SO(2)}$ can be obtained by multiplying the first column with the anti-symmetric matrix $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$:

$$\begin{bmatrix} -\sin(k\theta) \\ \cos(k\theta) \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cos(k\theta) \\ \sin(k\theta) \end{bmatrix}. \tag{75}$$

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

Definition B.43. *(Endomorphism Basis) Let G be a compact group and $\psi_i : G \to GL(V)$ be a real irrep of G. 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 $g \in G$.*

Now, 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. B.37).

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 the endomorphism space End_{ψ_i} 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 (Cesa et al., 2022) 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 $\overline{\psi}_i(g)$:

$$\psi_i(g) = \begin{bmatrix} c_0^i \overline{\psi}_i(g) & | & c_1^i \overline{\psi}_i(g) & | & \dots & | & c_{m_i}^i \overline{\psi}_i(g) \end{bmatrix} \tag{76}$$

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 follows:

$$\mathcal{C}_{\psi_0^{SO(2)}} = \left\{ [1] \right\} \tag{77}$$

$$\mathcal{C}_{\psi_k^{SO(2)}} = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \right\}. \tag{78}$$

$$\tag{79}$$

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.

$$\mathcal{C}_{\psi^{O(2)}} = \left\{ I_{d_{\psi^{O(2)}} \times d_{\psi^{O(2)}}} \right\}. \tag{80}$$

Cesa et al. (2022, 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:

$$\left\{ \sqrt{d_\psi} \psi_i(g)_j \mid \psi_i \in \widehat{G}, 1 \leq i, j \leq n_i \right\}, \tag{81}$$

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 B.44. (*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:*

$$f(g) = \sum_{\psi_j \in \widehat{G}} \sqrt{d_j} Tr\left(\psi_j(g)^\top \mathcal{R}_{\psi_j}\left(\widehat{f}(\psi_j) \right) \right), \tag{82}$$

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 B.45. (*Fourier Transform*) *Given a group G and a function $f : G \to \mathbb{R}$, the Fourier transform of f is given as:*

$$\widehat{f}(\psi_j) = \sqrt{d_j} \int_G f(g) \mathcal{R}_{\psi_j}^{-1}\left(\psi_j(g) \right) dg \tag{83}$$

if G is an infinite group, and

$$\widehat{f}(\psi_j) = \sqrt{d_j} \frac{1}{|G|} \sum_{g \in G} f(g) \mathcal{R}_{\psi_j}^{-1}\left(\psi_j(g) \right) \tag{84}$$

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 (Cesa et al., 2022; Lang & Weiler, 2020).

Definition B.46. (*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:*

$$\left\{ Y_j^{km} : \mathbb{R}^n \to \mathbb{R} \mid \psi_j \in \widehat{H}, m \leq d_j \right\}. \tag{85}$$

This is an application of the Peter-Weyl theorem (Thm. B.42), 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:

$$Y_j^k(g \cdot x) = \psi_j(g) Y_j^k(x) \quad \forall g \in G, \ x \in \mathbb{R}^n \tag{86}$$

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 (Lang & Weiler, 2020).

C. Additional Derivations

In this section we cover additional and more in-depth derivations. In C.1 we discuss the irreps decomposition. In Section C.2 we solve the equivariant constraint in more detail. In Section C.3 we cover the alignment problem in more detail. Finally, in Section C.4 we show a derivation for our solution in Eq. 20 regarding the computation of KL-divergence on the Fourier domain.

C.1. Irreps Decomposition

The steerability constraint depends 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, (Cesa et al., 2022; Weiler & Cesa, 2019) 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. B.30). Therefore, the kernel constraint from 2 can be written as:

$$k(hx) = Q_{\text{out}}^{-1} \left[\bigoplus_{i \in I_{\text{out}}} \psi_i(h) \right] Q_{\text{out}} k(x) Q_{\text{in}}^{-1} \left[\bigoplus_{j \in I_{\text{in}}} \psi_j(h)^{-1} \right] Q_{\text{in}} \quad \forall h \in H, x \in \mathbb{R}^n. \tag{87}$$

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

$$\widetilde{k}(hx) = \left[\bigoplus_{i \in I_{\text{out}}} \psi_i(h) \right] \widetilde{k}(x) \left[\bigoplus_{j \in I_{\text{in}}} \psi_j(h)^\top \right] \quad \forall h \in H, x \in \mathbb{R}^n. \tag{88}$$

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

$$\underbrace{\begin{bmatrix} \widetilde{k}^{i_1 j_1}(hx) & \widetilde{k}^{i_1 j_2}(hx) & \dots \\ \widetilde{k}^{i_2 j_1}(hx) & \widetilde{k}^{i_2 j_2}(hx) & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}}_{\widetilde{k}(hx)} = \underbrace{\begin{bmatrix} \psi_{i_1}(h) & & \\ & \psi_{i_2}(h) & \\ & & \ddots \end{bmatrix}}_{\left[\bigoplus_{i \in I_{\text{out}}} \psi_i(h) \right]} \underbrace{\begin{bmatrix} \widetilde{k}^{i_1 j_1}(x) & \widetilde{k}^{i_1 j_2}(x) & \dots \\ \widetilde{k}^{i_2 j_1}(x) & \widetilde{k}^{i_2 j_2}(x) & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}}_{\widetilde{k}(x)} \underbrace{\begin{bmatrix} \psi_{j_1}(h)^\top & & \\ & \psi_{j_2}(h)^\top & \\ & & \ddots \end{bmatrix}}_{\left[\bigoplus_{j \in I_{\text{in}}} \psi_j(h)^\top \right]}. \tag{89}$$

Thus, this constraint decomposes into multiple independent constraints:

$$\widetilde{k}^{ij}(hx) = \psi_i(h)\widetilde{k}^{ij}(x)\psi_j(h)^\top \quad \forall h \in H, x \in \mathbb{R}^n, \tag{90}$$

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 (Weiler & Cesa, 2019), 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.

C.2. Solving Equivariant Kernel Constraint

Here we provide a more complete derivation for the kernel constraint for fully equivariant steerable convolutions. Consider the kernel constraint:

$$k(x) = \psi_J(h)k(h^{-1}x)\psi_l(h)^\top \quad \forall h \in H, x \in \mathbb{R}^n \tag{91}$$

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:

$$k(x) = \Pi_H(\widehat{K})(x) = \int_{h \in H} \mu(h)\psi_J \widehat{K}(h^{-1}x)\psi_l(g)^\top dg \tag{92}$$

Applying columnwise vectorisation using $\kappa(\cdot) = \text{vec}(k(\cdot))$:

$$\kappa(x) = \text{vec}(\Pi_H(\widehat{K})(x)) = \int_{h \in H} \mu(h)(\psi_l \otimes \psi_J)(h)\widehat{\kappa}(h^{-1}x)dh \tag{93}$$

To parameterise the vectorised unconstrained kernel $\widehat{\kappa} : \mathbb{R}^n \to \mathbb{R}^{d_j d_l}$, 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. B.46), and a weight matrix $W_{j,k}$ of shape $d_j d_l \times d_j$:

$$\widehat{\kappa(x)} = \sum_{j,k} W_{j,k} Y_j^k(x). \tag{94}$$

Noting that $Y_j^k(h \cdot x) = \psi_j(h) Y_j^k(x)$ (Def. B.46), we re-write Eq. 93 using this steerable basis parameterisation:

$$\kappa(x) = \sum_{j,k} \left[\int_{h \in H} \mu(h)(\psi_l \otimes \psi_J)(h) W_{jk} \psi_j(h)^\top dh \right] Y_j^k(x). \tag{95}$$

$$\tag{96}$$

We can decompose the tensor product using the *Clebsch-Gordan decomposition* (Thm. B.37):

$$(\psi_l \otimes \psi_J)(h) = \sum_{j'} \sum_s^{[j'(Jl)]} \left[\text{CG}_s^{j'(Jl)} \right]^\top \psi_{j'}(h) \, \text{CG}_s^{j'(Jl)}. \tag{97}$$

Using this decomposition, Eq. 95 becomes:

$$\kappa(x) = \sum_{j,k} \left[\int_{h \in H} \mu(h) \sum_{j'} \sum_s^{[j'(Jl)]} \left[\text{CG}_s^{j'(Jl)} \right]^\top \psi_{j'}(h) \, \text{CG}_s^{j'(Jl)} W_{jk} \psi_j(h)^\top dh \right] Y_j^k(x). \tag{98}$$

$$\tag{99}$$

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

$$\kappa(x) = \sum_{j,k} \sum_{j'} \sum_s^{[j'(Jl)]} \left[\text{CG}_s^{j'(Jl)} \right]^\top \left[\int_{h \in H} \mu(h)\psi_{j'}(h) \, \text{CG}_s^{j'(Jl)} W_{jk} \psi_j(h)^\top dh \right] Y_j^k(x). \tag{100}$$

$$\tag{101}$$

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 $\text{unvec}(\cdot)$, which reverses the columnwise vectorisation given by $\text{vec}(\cdot)$. Using the property $\text{vec}(ABC) = (C^\top \otimes A)\text{vec}(B)$, we re-write Eq. 100 to:

$$\kappa(x) = \sum_{j,k} \sum_{j'} \sum_s^{[j'(Jl)]} \left[\text{CG}_s^{j'(Jl)} \right]^\top \text{unvec} \left[\int_{h \in H} \mu(h)(\psi_j \otimes \psi_{j'})(h) W_{j,j',k,s} dh \right] Y_j^k(x) \tag{102}$$

Consequently, the integral contains a tensor product and can thus be decomposed through the *Clebsch-Gordan* decomposition (Thm. B.37)::

$$\int_{h\in H} \mu(h)\,(\psi_j \otimes \psi_{j'})\,(h)W_{j,j',k,s}dh = \int_{h\in H} \mu(h)\left(Q\left(\bigoplus_i \overset{[i(jj')]}{\bigoplus_r} \psi_i(h)\right)Q^\top\right)W_{j,j',k,s}dh \tag{103}$$

$$= Q\bigoplus_i \overset{[i(jj')]}{\bigoplus_r}\left(\int_{h\in H}\mu(h)\psi_i(h)dh\right)Q^\top W_{j,j',k,s} \tag{104}$$

$$\tag{105}$$

If we consider the irreps ψ_i as basis-functions, as described in Theorem B.42, we can describe the function $\mu(h)$ in terms of its Fourier coefficients (Def. B.45):

$$\widehat{\mu}(\psi_i) = \sqrt{d_i}\int_{h\in H}\mu(h)\mathcal{R}_{\psi_i}^{-1}\left(\psi_i(h)\right)dh \quad \forall \psi_i \in \widehat{H}, \tag{106}$$

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. B.43). Therefore, we reformulate the integral in Equation 104 in terms of its Fourier coefficients:

$$\int_{h\in H}\mu(h)\,(\psi_j \otimes \psi_{j'})\,(h)W_{j,j',k,s}dh = Q\bigoplus_i \overset{[i(jj')]}{\bigoplus_r}\left(\frac{\mathcal{R}_{\psi_i}\left(\widehat{\mu}(\overline{\psi}_i)\right)}{\sqrt{d_i}}\right)Q^\top W_{j,j',k,s} \tag{107}$$

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 $\mathcal{C}_{\psi_0} = \left\{[1]\right\}$,

$$\int_{h\in H}\mu(h)\,(\psi_j \otimes \psi_{j'})\,(h)W_{j,j',k,s}dh = Q\underbrace{\begin{bmatrix} I_{[0(jj')]\times[0(jj')]} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}}_{\left(\oplus_i \oplus_r^{[i(jj')]} \frac{\widehat{\mu}(\psi_i)}{\sqrt{d_i}}\right)}Q^\top W_{j,j',k,s}, \tag{108}$$

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

$$\int_{h\in H}\mu(h)\,(\psi_j \otimes \psi_{j'})\,(h)W_{j,j'k,s}dh = \sum_r^{[0(jj')]} Q_r Q_r^\top W_{j,k,s,r}, \tag{109}$$

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 $W_{j,k,s,r}$, resulting in a single weight $W_{j,k,s,r} \in \mathbb{R}$.

When the representations, ψ_j and $\psi_{j'}$, are identical, the multiplicity $[0(jj')]$ of the trivial representation in the Clebsch-Gordan decomposition of $\psi_j \otimes \psi_{j'}$ is equal to the size of the endomorphism basis $|\text{End}_{\psi_j}| = c_{\psi_j}$. 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 109 can be re-written as follows:

$$\int_{h\in H}\mu(h)\,(\psi_j \otimes \psi_{j'})\,(h)W_{j,k,s,r}dh = \sum_r^{[0(jj')]} \text{vec}(c_r^J)W_{j,k,s,r} \tag{110}$$
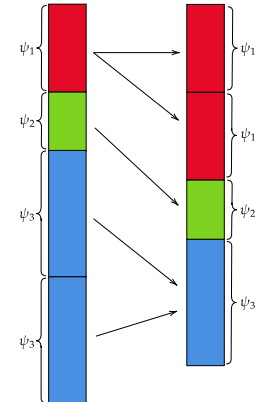

Figure 6. Mapping between different input and output representations. Representations are colour-coded. A mapping only (and always) exists between equal irreps.

Conversely, if $\psi_j \neq \psi_{j'}$, the Clebsch-Gordan decomposition lacks a trivial representation and, therefore, multiplicity $[0(lJ)] = 0$. This leads to a null-map as all emerging Fourier coefficients are zero. Consequently, the weight matrix is also projected to zeros, rendering it a null-map and making these weights redundant. See Figure 6 for a visualisation.

Replacing the integral in Eq. 102 with Eq. 110, and replacing j' with j gives:

$$\kappa(x) = \Pi_H(\widehat{K})(x) = \sum_{j,k} \sum_s^{[j'(Jl)]} \left[\mathrm{CG}_s^{j(Jl)}\right]^\top \mathrm{unvec} \left[\sum_r^{0[jj]} \mathrm{vec}(c_r^j) W_{j,k,s,r}\right] Y_j^k(x) \tag{111}$$

$$= \sum_{j,k} \sum_s^{[j'(Jl)]} \left[\mathrm{CG}_s^{j(Jl)}\right]^\top \left[\sum_r^{0[jj]} c_r^j W_{j,k,s,r}\right] Y_j^k(x) \tag{112}$$

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 (Cesa et al., 2022), where the summations have been removed in favour of additional indexing.

C.3. Alignment Problem

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 projection integral for the partially equivariant SCNN from Eq. 12, where we insert the constrained weight matrices $W_{j,j',k,s}$:

$$\int \lambda(h)(\psi_j \otimes \psi_{j'})(h) dh W_{j,j',k,s} = Q \sum_i^{[i(jj')]} \sum_r \frac{\mathcal{R}_{\psi_i}\left(\widehat{\lambda}(\overline{\psi_i})\right)}{\sqrt{d_i}} Q^\top W_{j,j',k,s}. \tag{113}$$

Note here that the change of basis $Q^\top$ is not absorbed in the weights $W_{j,j',k,s}$. To illustrate the problem, we perform the projection given by $\xi_{\mathcal{R}_h\lambda}(W'_{j,j',k,s})$, where $W'_{j,j',k,s} = (\psi_j \otimes \psi_{j'})(h)W_{j,j',k,s}$. Here, the likelihood distribution undergoes a transformation through the right-action of an element $h \in H$ using the right-regular representation $\mathcal{R}_h$ (Def. B.33).

$$\int \lambda(h)(\psi_j \otimes \psi_{j'})(h) dh W'_{j,j',k,s} = Q \left(\bigoplus_i^{[i(jj')]} \bigoplus_r \frac{\mathcal{R}_{\psi_i}\left(\widehat{[\mathcal{R}_h \cdot \lambda]}(\overline{\psi_i})\right)}{\sqrt{d_i}}\right) Q^\top \widehat{W}' \tag{114}$$

$$= Q \left(\bigoplus_i^{[i(jj')]} \bigoplus_r \frac{\mathcal{R}_{\psi_i}\left(\widehat{\lambda}(\overline{\psi_i})\right) \psi_i(h)^\top}{\sqrt{d_i}}\right) Q^\top W'_{j,j',k,s} \tag{115}$$

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

$$= Q \left(\bigoplus_i^{[i(jj')]} \bigoplus_r \frac{\mathcal{R}_{\psi_i}\left(\widehat{\lambda}(\overline{\psi_i})\right)}{\sqrt{d_i}}\right) Q^\top Q \left(\bigoplus_i^{[i(jj')]} \bigoplus_r \psi_i(h)^\top\right) Q^\top W'_{j,j',k,s} \tag{116}$$

Recall that the second direct-sum over ψ_i is simply equal to the Clebsch-Gordan decomposition of the tensor product $(\psi_j \otimes \psi_{j'})(h)$. Therefore:

$$= Q \left(\bigoplus_i^{[i(jj')]} \bigoplus_r \frac{\mathcal{R}_{\psi_i}\left(\widehat{\lambda}(\overline{\psi_i})\right)}{\sqrt{d_i}}\right) Q^\top (\psi_j \otimes \psi_{j'})(h)^\top \widehat{W}' \tag{117}$$

since $W'_{j,j',k,s} = (\psi_j \otimes \psi_{j'})(h)W_{j,j',k,s}$

$$\text{vec}\left(\xi_{\mathscr{R}_h\lambda}\left(\widehat{W'}\right)\right) = Q\left(\bigoplus_i \bigoplus_r^{[i(jj')]} \frac{\mathcal{R}_{\psi_i}\left(\widehat{\lambda}(\overline{\psi_i})\right)}{\sqrt{d_i}}\right) Q^\top (\psi_j \otimes \psi_{j'})(h)^\top (\psi_j \otimes \psi_{j'})(h)\widehat{W} \tag{118}$$

$$= Q\left(\bigoplus_i \bigoplus_r^{[i(jj')]} \frac{\mathcal{R}_{\psi_i}\left(\widehat{\lambda}(\overline{\psi_i})\right)}{\sqrt{d_i}}\right) Q^\top \widehat{W} \tag{119}$$

$$= \int \lambda(h)(\psi_j \otimes \psi_{j'})(h)dh W_{j,j',k,s} \tag{120}$$

As a result, performing these transformations on the likelihood λ and the weights $W_{j,j',k,s}$ 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.

C.4. KL-Divergence on Fourier domain

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) (Kullback & Leibler, 1951). 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 (Kingma & Welling, 2013). Given a distribution P along with a reference distribution Q, KL-divergence is defined as:

$$D_{KL}(P \| Q) = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx. \tag{121}$$

Or, in a discretised setting:

$$D_{KL}(P \| Q) = \sum_{x \in X} p(x) \log\left(\frac{p(x)}{q(x)}\right). \tag{122}$$

Note that, due to the weighting of the logarithmic term by $p(x)$, the KL divergence is not a symmetrical measure, thus $D_{KL}(P \| Q) \neq D_{KL}(Q \| P)$; 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. 122 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. 122:

$$D_{KL}(\lambda_1 \| \lambda_0) = \sum_{x \in X} \lambda_1(x) \log \lambda_1(x) - \sum_{x \in X} \lambda_1(x) \log \lambda_0(x) \tag{123}$$

$$= \boldsymbol{\lambda_1}^\top \log \boldsymbol{\lambda_1} - \boldsymbol{\lambda_1}^\top \log \boldsymbol{\lambda_0}, \tag{124}$$

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:

$$D_{KL}(\lambda_1 \| \lambda_0) = \widehat{\boldsymbol{\lambda_1}}^\top \widehat{\log \boldsymbol{\lambda_1}} - \widehat{\boldsymbol{\lambda_1}}^\top \widehat{\log \boldsymbol{\lambda_0}} \tag{125}$$

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 with the normalisation in Section 3.2, their logarithms are not directly available without first computing them. Fortunately, through the normalisation approach in Section 3.2, the unnormalised likelihood distribution (or log likelihood) relates to the logarithm of the normalised distribution:

$$\log \lambda(x) = \log \left(\frac{e^{\lambda'(x) - \max(\lambda')}}{z} \right) \tag{126}$$

$$= \lambda'(x) - \max(\lambda') - \log z. \tag{127}$$

Again, using the orthogonality of the Fourier transform, we can simply add these additional terms to Eq. 125 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:

$$D_{KL}(\lambda_1 \,||\, \lambda_0) = \widehat{\boldsymbol{\lambda_1}}^\top \widehat{\boldsymbol{\lambda_1'}} - \max(\lambda_1') - \log z_1 - \widehat{\boldsymbol{\lambda_1}}^\top \widehat{\boldsymbol{\lambda_0'}} + \max(\lambda_0') + \log z_0 \tag{128}$$

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.

D. Additional Theoretical Details

D.1. Comparison with Alternative Partially Equivariant Approaches

In this section, we provide a more in depth comparison with some related approaches to learn approximate equivariance. A first important difference with previous works is that we only support equivariance breaking over compact groups and, therefore, cannot break translation equivariance. Hence, we structure this section in three parts: first we compare with methods considering group convolution, but restrict our consideration to compact groups (in this case, a GCNN is equivalent to an equivariant MLP in our framework); then, we compare specifically with RPP for the case of generic equivariant MLPs, since this method is closely related to ours; finally, we compare with methods that support steerable CNNs (where, however, we only break equivariance to the compact rotation and reflection subgroup).

Compact group GCNNs We first consider the special case of GCNNs. Recall that group convolution can be expressed as a linear operator with kernel $k : G \times G \to \mathbb{R}$:

$$[k \star f] (h) := \int_{g \in G} k(h, g) f(g) dg \tag{129}$$

which is equivariant to G iff the kernel can be written as a 1-argument kernel $k(h, g) = \hat{k}(h^{-1}g)$. At a high level, besides the underlying implementations, both Romero & Lohit (2022) and Wang et al. (2022b) use the equivariant kernel $\hat{k}(h^{-1}g)$ weighted by a learnable component $w(g)$, i.e. $k(h, g) = \hat{k}(h^{-1}g)w(g)$ (in a more or less flexible way; e.g. in (Romero & Lohit, 2022), $w(g)$ is the learned probability distribution over G). Similarly, van der Ouderaa et al. (2022) break equivariance by using *stationary filters*, i.e. by replacing the 1-argument kernel $\hat{k}(h^{-1}g)$ with a 2-arguments kernel $k(h^{-1}g, g)$ conditioned on the input location g but initialised to be dependent only on the first argument $h^{-1}g$. While more flexible than the previous two approaches, stationary filters still follows the same principle, i.e. *breaking equivariance by learning a dependency on the absolute input location g.*

Instead, we follow a different principle. Ignoring the bandlimiting, our approach learns the unconstrained kernel $k : G \times G \to \mathbb{R}$ and a likelihood distribution λ over G and performs the projection to the equivariant subspace of 1-argument kernels via group averaging (Reynold's operator), weighted by λ, that is:

$$\bar{k}(h, g) = \int_e k(eh, eg) \lambda(e) de . \tag{130}$$

One can verify that $\bar{k}(h, g)$ is equivalent to a one argument kernel $\hat{k}(h^{-1}g)$ when λ is a uniform distribution over G, i.e. it is G-equivariant. In other words, our method learns the degree of weight sharing to apply on a learnable unconstrained kernel.

Additionally, we note that in (Romero & Lohit, 2022), a smaller support for the learnt probability distribution does not change the number and expressiveness of filters, but only the number of input locations in their field of view, thereby reducing the computational cost. In our case, instead, the computational cost is constant, but the reduced support leads to more learnable weights used to parameterise the filters. For example, if λ is a uniform distribution over the subgroup $H < G$ and both G and H are finite groups, then the projected operator $\bar{k}$ becomes equivalent to $\frac{|G|}{|H|} \times \frac{|G|}{|H|}$ independent H-convolutions - with a total complexity equivalent to a single G-convolution. Conversely, the method proposed by Romero & Lohit (2022) would result in a single H-convolution.

Residual Pathway Priors   RPP (Finzi et al., 2021) is more closely related to our approach, for the case of a generic compact-group equivariant MLP (not necessarily using the regular representation and, therefore, not necessarily equivalent to a GCNN). Indeed, let W be the weight matrix of a linear layer and G the equivariance group of interest, such that an equivariant layer should satisfy $\rho_{out}(g)W = W\rho_{in}(g)$, that is $\text{vec}(W) = (\rho_{in} \otimes \rho_{out})(g)\text{vec}(W)$. An RPP prior on the vectorised weights $\text{vec}(W)$ has the form $\text{vec}(W) \sim \mathcal{N}(0, (\sigma_a^2 + \sigma_b^2)QQ^T + \sigma_b^2 PP^T)$ as in Sec. 4 of (Finzi et al., 2021), with Q a basis for the G equivariant subspace and P a basis for its complement. This formulation is comparable to our framework with the following **fixed likelihood function**:

$$\lambda(g) \propto \frac{1}{\sigma_b^2}\delta_e(g) + \left(\frac{1}{\sigma_a^2 + \sigma_b^2} - \frac{1}{\sigma_b^2}\right) \tag{131}$$

where $\delta_e(g)$ is a Dirac delta function centred at the identity $e \in G$, combined with an L2 weight decay regularisation term on the final projected weights; indeed, plugging this distribution in the averaging operator leads to:

$$\Pi_\lambda(\text{vec}(W)) = \int_G (\rho_{in} \otimes \rho_{out})(g)\,\text{vec}(W)\lambda(g)dg \tag{132}$$

$$\propto \frac{1}{\sigma_b^2}\text{vec}(W) + \left(\frac{1}{\sigma_a^2 + \sigma_b^2} - \frac{1}{\sigma_b^2}\right)\Pi_G(\text{vec}(W)) \tag{133}$$

$$= \frac{1}{\sigma_b^2}P\alpha + \frac{1}{\sigma_a^2 + \sigma_b^2}Q\beta \tag{134}$$

for some weight vector α, β. An L2 weight decay term on $\Pi_\lambda(\text{vec}(W))$ would to the same regularisation used in RPP. However, in our experiments we typically used a weight decay regularisation directly on $\text{vec}(W)$ rather than on $\Pi_\lambda(\text{vec}(W))$; we did not explore the effect of this choice but this connection with RPP might be an interesting direction to further improve our method.

Steerable CNNs   The RSteer method from (Wang et al., 2022b) can be written as (from their open source code):

$$[k \star f](x) = \int_y dy \left[\sum_b \kappa_b(y - x)w_b(y - x)\right] f(y) \tag{135}$$

where $\{\kappa_b\}_b$ is the basis for equivariant steerable kernels. Essentially, rather than linearly combining the basis elements with a set of learnable weights, the authors use scalar weight functions $w_b : \mathbb{R}^n \to \mathbb{R}$; since these scalar functions are not steerable - $w_b(y - x) \neq w_b(g(y - x))$ - this causes the equivariance breaking. This can be thought as tweaking the underlying steerable basis, breaking its steerable property in a learnable way. We do not currently see any direct connection with our method.

D.2. Monotonically Decreasing Equivariance

In Section 3.2 we discussed employing KL-divergence as a soft constraint to ensure that the degree of equivariance is monotonically decreasing in the network. This choice was justified with the fact that once equivariance has been broken it cannot be recovered in a subsequent layer, and thus to accurately represent this a layer should never model a likelihood distribution that suggests it has re-gained equivariance that a previous layer has lost. However, in their experiments with Partial-GCNN, Romero & Lohit (2022) find that a higher degree of equivariance in the last layer compared to previous layers

often results in superior performance compared to when enforcing a monotonically decreasing equivariance by applying a ReLU activation over the differences in subsequent likelihood distributions.

In our experiments, we observed that using such a regularisation generally resulted in improved performance and stability (Apx. G.5.2). Furthermore, with or without the employment of our regularisation, we have not necessarily observed that our models consistently tended to become significantly more uniformly equivariant in the final layer. However, this could be due to the employment of KL-divergence in favour of their approach, and the fact that we detach the Fourier coefficients of layer n when computing the KL-divergence between layer n and $n + 1$, ensuring that backpropagation can only push layer $n + 1$ to be a subset of layer n while being unable to push layer n to be close to layer $n + 1$. This prevents the regularisation from overly reducing equivariance in earlier layers. Finally, it is important to note that since our constraint is a soft-constraint, our approach can still model such an increase in equivariance in later layers if required, as long as the KL-divergence is not weighted too strongly in the training objective.

E. Additional Implementation Details

This section covers additional details regarding implementations of the theoretical procedures described in the paper. All code implementations are performed as an extension of the PyTorch `escnn` library (Cesa et al., 2022).

E.1. Non-linearites

A GCNN layer maps an input in $\mathbb{R}^n \times GL(\mathbb{R}^{|H|})$ or $\mathbb{R}^n$ to regular features $GL(\mathbb{R}^{|H|})$. 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. 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. B.44) (Franzen & Wand, 2021; Poulenard & Guibas, 2021; Cesa et al., 2022). 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:

$$f_{\text{out}}(\boldsymbol{x}) = \mathcal{F}\left(\gamma\left(\mathcal{F}^{-1}\left(f_{\text{in}}\left(\boldsymbol{x}\right)\right)\right)\right) \quad \forall \boldsymbol{x} \in \mathbb{R}^n, \tag{136}$$

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 $= \text{FT}^{\dagger 11}$. Using these matrices, the Fourier-based activation function can be efficiently implemented as follows:

$$f_{\text{out}}(\boldsymbol{x}) = \text{FT}\, \gamma\left(\text{IFT}\, f_{\text{in}}(\boldsymbol{x})\right). \tag{137}$$

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 $SO(3)$, it is possible to use features of the quotient space $Q = H/K$ rather than the full space H Cesa et al. (2022); Bekkers et al. (2023), where $K \leq H$ is a subgroup of H. Although the exact definition of a quotient space is outside the scope of this paper, the quotient space $Q = H/K$ can be interpreted as the space of H with the subspace K factored out. We refer

---

[11] Here $\text{FT}^{\dagger}$ is the pseudo-inverse of FT

to (Cesa et al., 2022) 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   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 (Worrall et al., 2017; Weiler et al., 2018b), 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. (2018a) 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 & Cesa (2019) have shown that they are generally outperformed by the Fourier-based non-linearities.

E.2. Sharing Equivariance

Algorithm 1 enables the learning of partially equivariant mappings in SCNNs, 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. Using the provides framework, it is also possible to create equivariant MLPs (E-MLPs) by taking a steerable basis $\mathcal{B} = \left\{ Y_j^k : \mathbb{R} \to 1 \right\}_{j=0}^{k=0}$ and consider a space $X = \{0\}$. 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.

E.3. Interpretability losses

This section discusses the implementation details of the losses introduced in Section 3.2, 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. 20 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:

$$\mathcal{L}_{\text{KL}} = \frac{1}{|\tau|} \sum_{(m,n) \in \tau} D_{\text{KL}}(\lambda_m \,||\, \lambda_n) \tag{138}$$

from Eq. 20

$$= \frac{1}{|\tau|} \sum_{(m,n) \in \tau} \widehat{\boldsymbol{\lambda_m}}^{\top} \widehat{\boldsymbol{\lambda_m'}} - \max(\lambda_m') - \log z_m - \widehat{\boldsymbol{\lambda_m}}^{\top} \widehat{\boldsymbol{\lambda_n'}} + \max(\lambda_n') + \log z_n. \tag{139}$$

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 $n =$ 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. 19 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. 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.

$$\mathcal{L}_{\text{align}} = \frac{1}{|\Lambda|} \sum_{\lambda_i \in \Lambda} D_{\text{align}}(\lambda_i) \tag{140}$$

F. Experimental Details

This section contains an outline of our evaluation methodology used to obtain our results. First, in Section F.1 we outline the three main classes of datasets, along with their variations. Section F.2 contains descriptions of the model architectures used for the varying experimental setups.

F.1. Datasets

In this paper 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.

F.1.1. VECTORS

In our additional results we also evaluate our approach on equivariant MLPs. This can be achieved by taking an steerable basis $\mathcal{B} = \left\{ Y_j^k : \mathbb{R} \to 1 \right\}_{j=0}^{k=0}$ and consider a space $X = \{0\}$.

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.

F.1.2. DDMNIST

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 `DDMNIST` dataset, which is an adaptation of the popular MNIST dataset (LeCun et al., 1998). Whereas the original MNIST dataset contains $60,000$ 28×28 training images of single handwritten digits numbered 0 through 9, our `DDMNIST` 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 6 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 6. Overview of available symmetries for our `DDMNIST` 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 `DDMNIST` 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 for an example.

F.1.3. MEDICAL MNIST

To evaluate our approach on 3D groups, we use several volumetric datasets from the MedMNIST dataset collection (Yang et al., 2023). 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 (Kuipers & Bekkers, 2023), 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 (Bilic et al., 2023; Xu et al., 2019) 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` (Yang et al., 2023) focusses on 3D microscopy scans of neural synapses in adult rats to classify synapses as excitatory or inhibatory. 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(Gray, 1959). However, there are other potential discriminating features (Klemann & Roubos, 2011; Somogyi
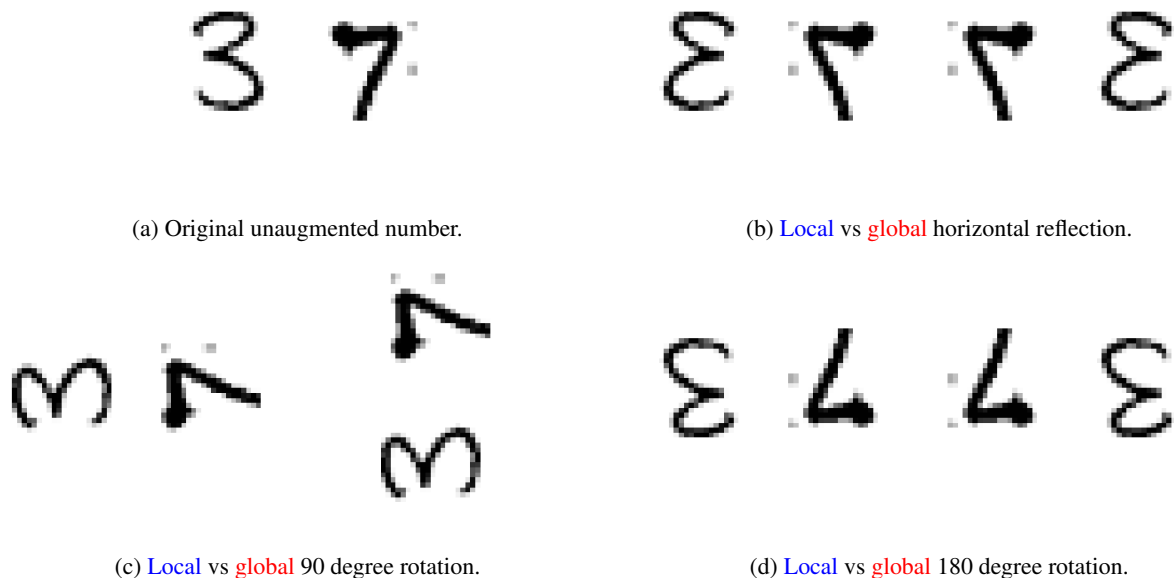
(a) Original unaugmented number.

(b) Local vs global horizontal reflection.

(c) Local vs global 90 degree rotation.

(d) Local vs global 180 degree rotation.

Figure 7. 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.

et al., 1998; Sheng & Kim, 2011). Although biomedically informed decisions are outside the scope of this paper, 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 (Armato III et al., 2011) 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 (Callister et al., 2015; MacMahon et al., 2005; Gould et al., 2013; Larici et al., 2017). 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.

F.1.4. SMOKE AND JETFLOW

Finally, to provide a more comprehensive comparison between alternative partially equivariant approaches we employ the Smoke and JetFlow datasets used by Wang et al. (2022b). They compare their RSteer and RGroup approach with various other approaches, allowing us to compare our models with their results.

Smoke   Generated by PhiFlow (Holl et al., 2020), the synthetic 64×64 2D Smoke dataset contains smoke simulations that allows for various types of equivariance breaking. In our case, we focuses on the partial rotation equivariances. In this setting, the dataset contains 40 simulations with different inflow positions and buoyant forces of the smoke. The inflow location and buoyant forces exhibit C_4 rotation symmetry, but the buoyant factor varies based on the inflow position, thus breaking rotation equivariance.

JetFlow   The 2D JetFlow dataset (Wang et al., 2022b) contains 62×23 real experimental turbulence velocity fields measured in NASA multi-stream jets. These fields are measured by 24 stations at different locations.

F.2. Model Architectures

For `Vectors`, `DDMINST` and `MedMNIST`, we use separate base structures 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 (Finzi et al., 2021), 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. For the `Smoke` and `JetFlow` datasets we employ the set-ups and implementations from Wang et al. (2022b).

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 8. This structure has four main configurations, see Table 7 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.

Configuration	Equivariance	Non-linearity
MLP	Non-equivariant	ELU
E-MLP	$O(2)$	Gated/FourierELU
PE-MLP (Preliminary) (ours)	Partial $O(2)$	Gated/FourierELU
PE-MLP (Probabilistic) (ours)	Partial $O(2)$	Gated/FourierELU

Table 7. Overview of the four main MLP architectures. FourierELU refers to a Fourier-based non-linearity using a pointwise ELU non-linearity acting on the sampled signal.

Figure 8. The base structure of the MLP networks. In and out refer to the number of input and output features.

2D CNN For the `DDMNIST` 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 9a 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 (Cesa et al., 2022)	Gated/FourierELU
RPP	SCNN + Residual Pathway Prior	RPP (Finzi et al., 2021)	Gated
R-SCNN	Restricted SCNN	Restriction (Cesa et al., 2022)	Gated
P-SCNN (ours)	Partial/Probabilistic Steerable CNN	Ours	Gated

Table 8. Overview of the main configurations of our base 2D CNN used for our `DDMNIST` 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 8. Here,
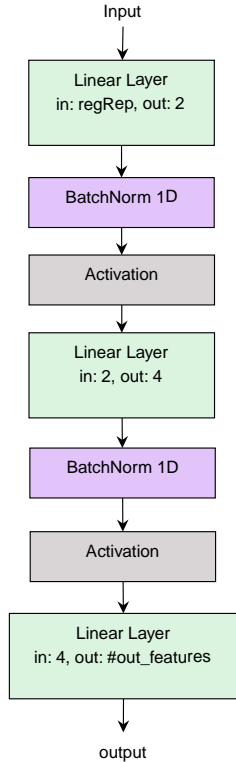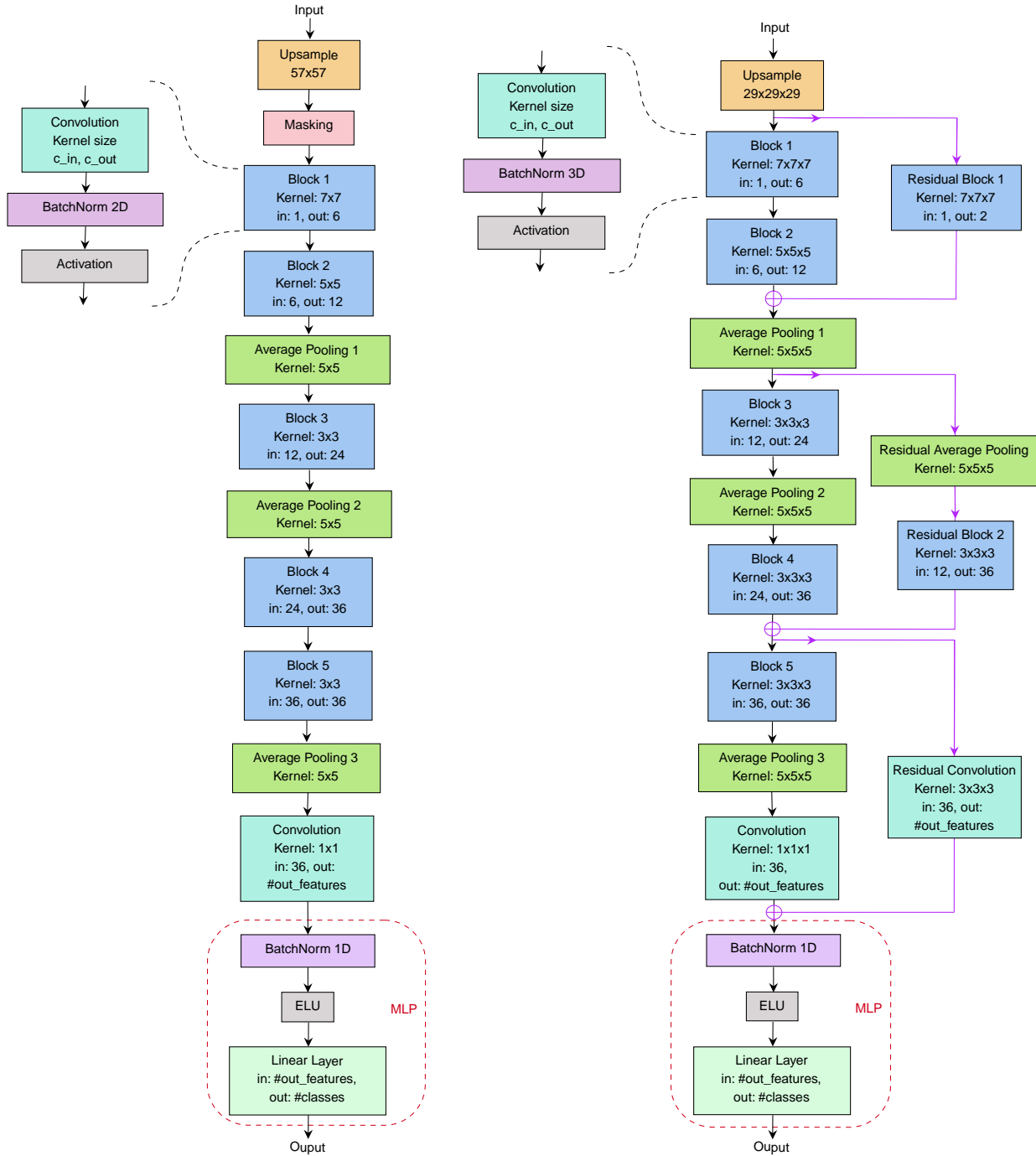
(a) Base 2D CNN configuration

(b) Base 3D CNN configuration

Figure 9. Diagrams of our 2D and 3D CNN base configurations.

the RPP configuration (Finzi et al., 2021) 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 DDMNIST. 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 9. In line with the MLP structure, we map to the trivial representation during the final 1×1 convolution to guarantee invariance by default.

Table 10 contains the maximum level of bandlimiting of the intermediate features for each group. Table 11 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 12 presents the strides and padding for each of the layers for the SCNN-based configurations.

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 9. Maximum level of bandlimiting of the intermediate features for each group.

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 10. Maximum level of bandlimiting of the intermediate for each supported transformation group.

Layer	Features Bandlimiting
1	4
2	3
3	*Trivial*

Table 11. 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	Padding 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 12. 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 10). 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.

3D CNNs For `MedMNIST` we use a modified version of our base structure used for `DDMNIST` 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 8. A diagram of our modified base-structure can be found in Figure 9b. 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 9. Similarly to the experiments on `DDMNIST`, we set the default bandlimiting of the likelihood distributions to $L = 2$.

Smoke/JetFlow CNNs Finally, for the experiments on the `Smoke` and `JetFlow` experiments we employ the same network structures as employed by Wang et al. (2022b). As these details are not available in the main paper, we rely on the corresponding code repository for the model architectures as well as the hyper-parameter tuning. For each of our models, we simply take the configuration as reported for their RSteer model in their code repository and directly port that to our approach without additional fine-tuning, besides for the α_{KL} and α_{align} regularisation weights. Tab. 13 contains an overview of the various approaches in their comparison, along with citation and explanation. All (partially) equivariant approaches from the main results in Sec. 5.1 use C_4-equivariance with regular feature fields. The band-limiting experiments in Apx. G.5.3 use $SO(2)$-equivariance encoded via irrep fields up to a frequency of 2.

	Model	Citation	Description
(non) equivariant approaches	MLP	-	Regular non-equivariant MLP
	CNN	LeCun et al. (1995)	Regular CNN
	e2cnn	Weiler & Cesa (2019)	SCNNs created with 2D predecessor of escnn
Partially equivariant approaches	RPP	Finzi et al. (2021)	Residual Pathway Priors
	Combo	Wang et al. (2022b)	Non-equivariant convolutions followed by equivariant convolutions
	CLCNN	Wang et al. (2022b)	Locally connected neural networks with equivariance constraints imposed in the loss function
	Lift	Wang et al. (2022a)	Lift expansion; non-equivariant features are lifted to the dimensionality of equivariant features and concatenated
	RGroup	Wang et al. (2022b)	GCNNs with weights conditioned on the group element
	RSteer	Wang et al. (2022b)	SCNNs with weights conditioned on the group element

Table 13. Overview of the various models used for the `Smoke` and `JetFlow` datasets employed by Wang et al. (2022b). RSteer and RGroup are part of their main contribution, whereas CLCNN, Lift and Combo are their alternative approaches.

As only the final configuration for `RSteer` is reported in their repository, we obtain the parameter counts by taking the `RSteer` configuration and porting that to any of the other approaches. The parameter counts we obtained in Sec. 5.1 therefore do not necessarily represent the parameter counts corresponding to the models used in the original paper to obtain the results. To sanity check, we verify that the models obtain performance comparable to the results reported by Wang et al. (2022b). This was not the case for CLCNN, so we did not include it in our results.

F.3. Equivariance Error

To verify the validity of the learnt likelihood distributions, we include measured equivariance errors across the group. However, instead of simply computing the error, 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:

$$\epsilon_l(h) = \frac{1}{|B|} \sum_{b \in B} \frac{||h \cdot l(b) - l(h \cdot b)||_2}{||l(b)||_2}, \quad \forall h \in H. \tag{141}$$

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.

F.4. Training details

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 (Kingma & Ba, 2014). For our *probabilistic* PE-MLP and our P-SCNN we use the following objective function:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \alpha_{\text{align}}\mathcal{L}_{\text{align}} + \alpha_{\text{KL}}\mathcal{L}_{\text{KL}} \tag{142}$$

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 14 for more specific training details on the individual datasets, including these weighting factors.

Dataset	Vectors	DoubleMNIST	OrganMNIST3D	SynapseMNIST3D	NoduleMNIST3D	Smoke	JetFlow
Learning rate	5×10^{-5}	5×10^{-5}	5×10^{-5}	1×10^{-5}	1×10^{-5}	1×10^{-3}	1×10^{-3}
batch size	1024	256	32	32	32	16	16
# of epochs	100	50	100	100	100	100	100
Loss function	MSE	Cross-entropy	Cross-entropy	Cross-entropy	Cross-entropy	MSE	MSE
Evaluated Epoch	Last	Best on validate	Last	Last	Last	Best on validate	Best on validate
α_{align}	5	5	5	5	5	5	5
α_{KL}	25	3	1	1	1	3	3

Table 14. 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 (Finzi et al., 2021).

G. Additional Experiments

In this section we cover additional experiments. In these experiments, we include *preliminary* approach to learning the degree of equivariance. In contrast to our main approach, this approach does not parameterise the degree of equivariance through a likelihood distribution. Instead, for each irrep pair $(\psi_j, \psi_{j'})$ we construct the projection matrix $c^{jj'}$ as in the fully equivariant state. Then we directly parameterise this projection matrix for each irrep pair $(\psi_j, \psi_{j'})$. As a result, this approach has significantly less weight sharing, and therefore a reduced consistency in terms of breaking equivariance.

In Section G.1 we cover the aforementioned comparison between the preliminary approach and our main approach. Section G.2 contains the confusion matrices obtained on OrganMNIST3D from the MedMNIST collection. Section G.3 provides additional likelihood distributions, including ones on our 3D models. In Section G.4 we compare the concept of shared versus layer-wise equivariance in E-MLPs. Section G.5 covers our experiments regarding KL-divergence and alignment loss, as well as additional results from our bandlimiting experiments. Here we show that these two terms are essential for interpretable likelihood distributions. Finally, in Section G.6 we combine the tuning of bandlimiting and the use of stronger (less equivariant) non-linearities to maximise performance.

G.1. Preliminary Approach compared to Probabilistic Approach

To assess the influence of weight sharing 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 described above, and one with a small amount of added noise to the matrices that are initialised as zeros. Table 15 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	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)	**0.001** (0.001)

Table 15. 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.

G.2. Benchmarks on Biomedical Image Classification

Here we provide additional confusion matrices for our results on `MedMNIST`

(a) CNN

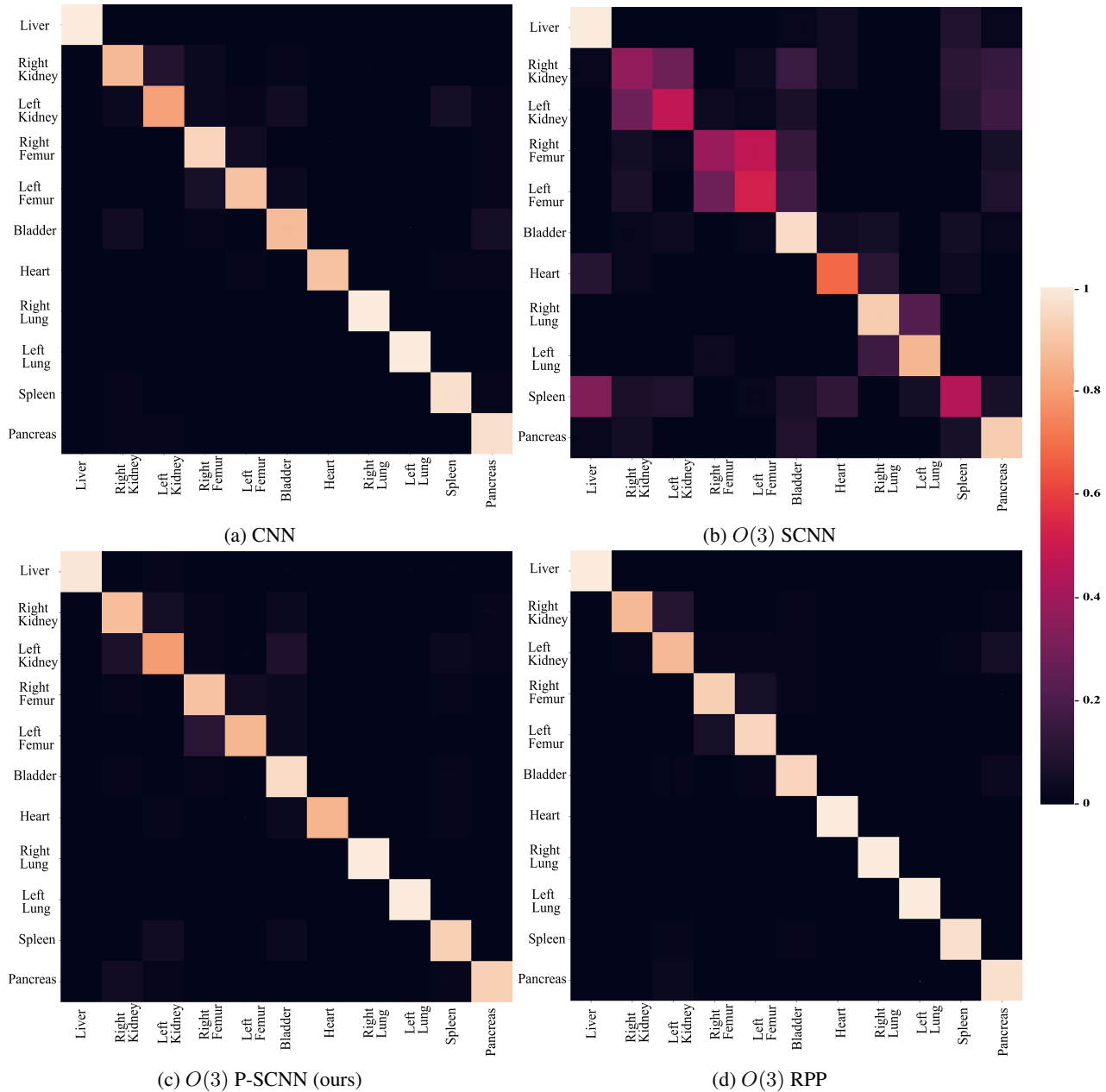(b) $O(3)$ SCNN

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

(d) $O(3)$ RPP

Figure 10. Confusion matrices obtained on OrganMNIST3D symmetries using a regular CNN, an $O(3)$ SCNN, $O(3)$ RPP and our $O(3)$ P-SCNN.

The confusion matrices in Figure 10 provide some insight into this behaviour on `OrganMNIST3D`. Here, the CNN's confusion matrix in Figure 10a 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 10b, 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 10c, 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 10d manages to show an additional slight improvement for these specific organs.

G.3. Inspecting the Learnt Likelihood distributions

Here we present additional learnt likelihood distributions for the 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.
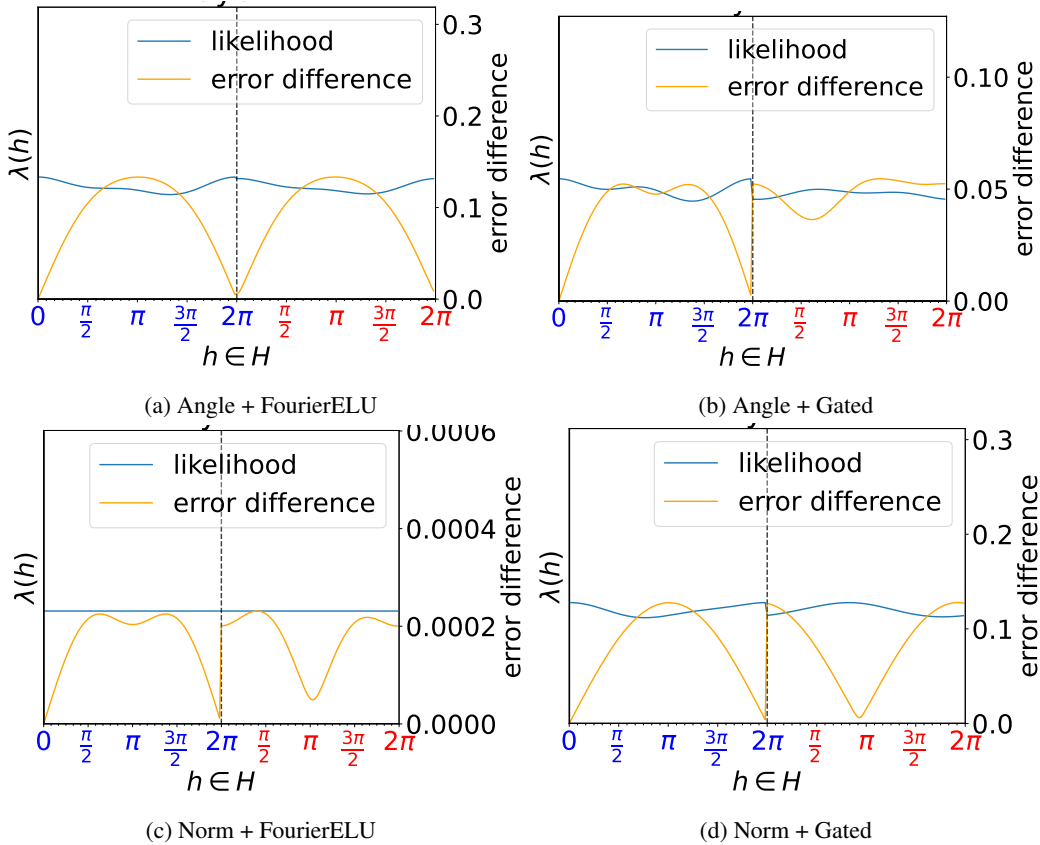

(a) Angle + FourierELU

(b) Angle + Gated

(c) Norm + FourierELU

(d) Norm + Gated

Figure 11. 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.

Vectors Figure 11 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. 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 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.

An analysis of Figure 11 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 11c.

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

49

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.
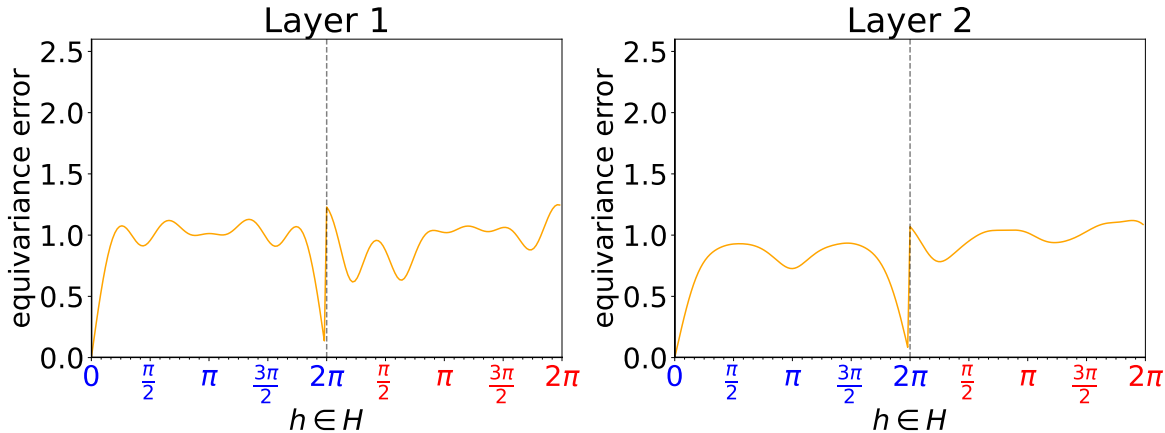
(a) Preliminary approach on angle regression task.
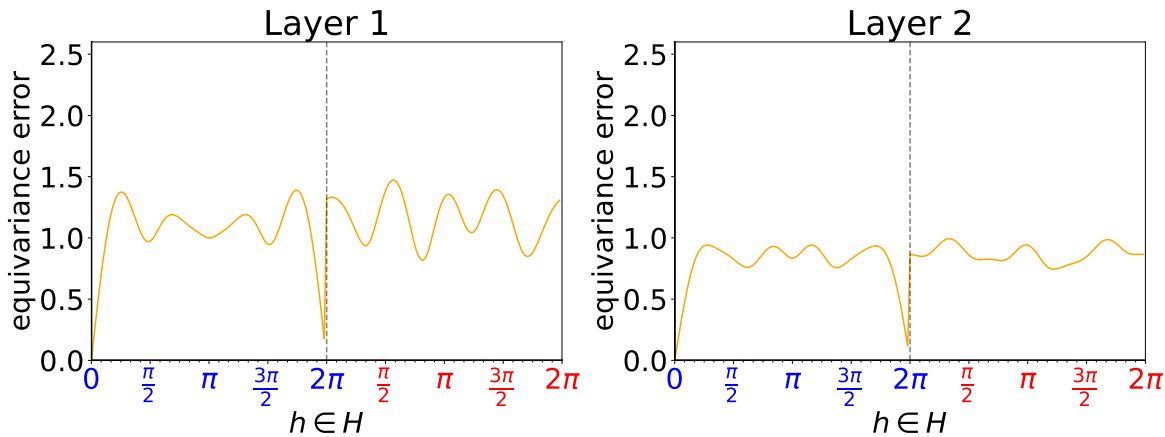
(b) Preliminary approach on norm regression task.

Figure 12. 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.

Finally, the equivariance errors in the *probabilistic* approach manifest low-frequency signals, whereas those in the *preliminary* approach, as seen in Figure 12, 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.

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 13. Note that earlier layers correspond to smaller-scale features and later layers correspond to larger-scale features.
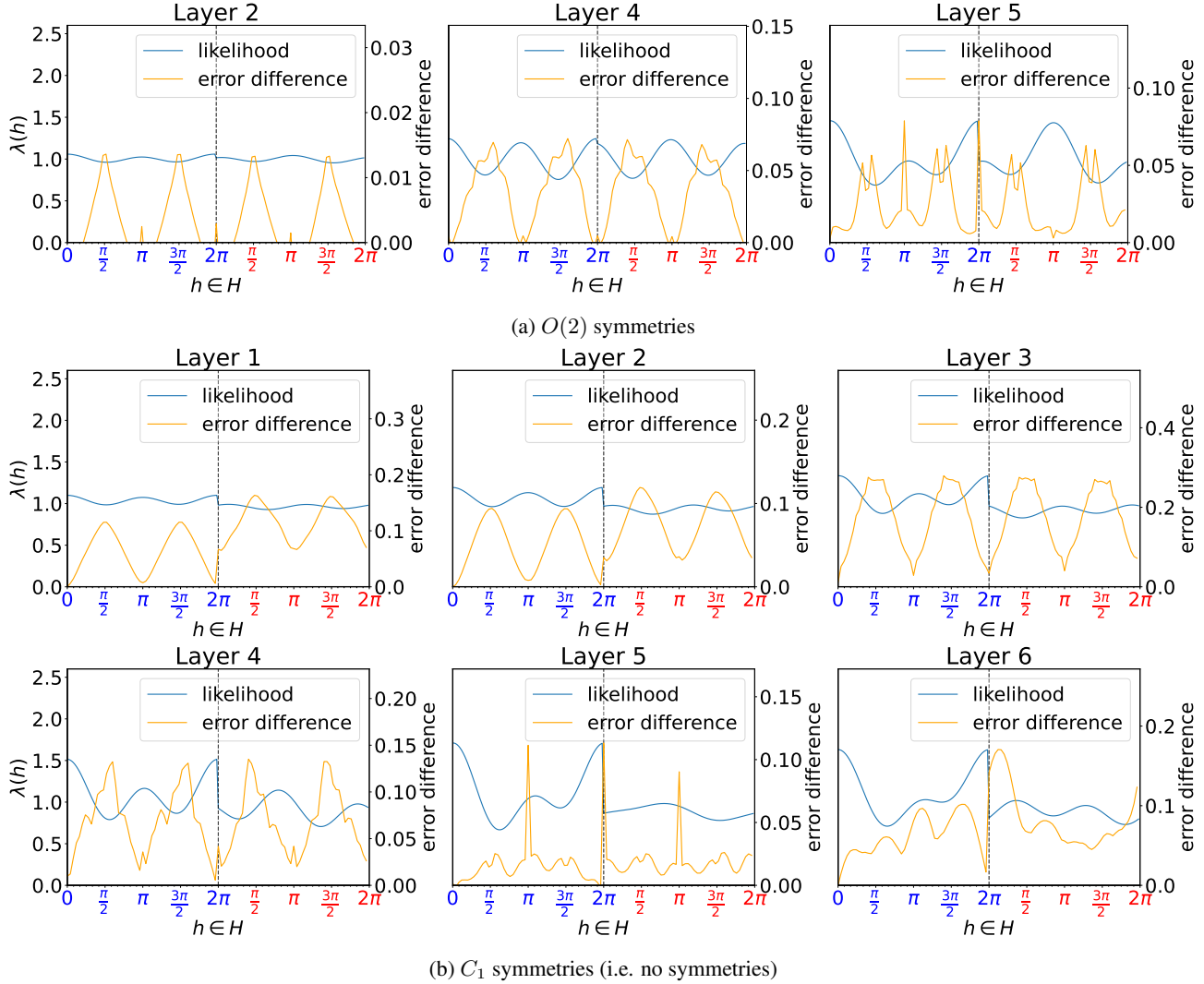
(a) $O(2)$ symmetries

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

Figure 13. 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.

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 13a, 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 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 14 for a visualisation.

Comparatively, the model trained on C_1 symmetries in Figure 13b breaks equivariance much more rapidly throughout the network compared to the previous model, with higher equivariance errors as a result. Most notable, reflection equivariance
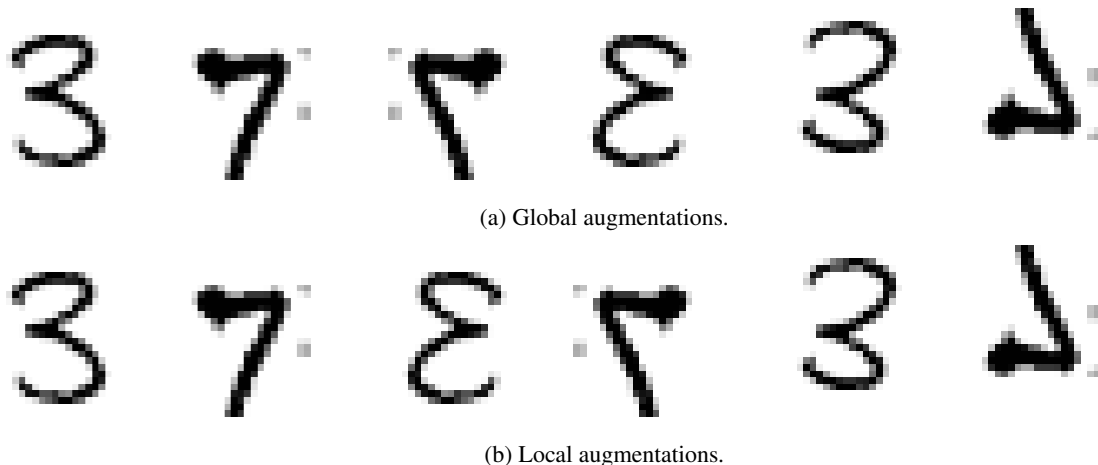
(a) Global augmentations.

(b) Local augmentations.

Figure 14. 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.

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 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 15 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.*

G.4. 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 E.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 G.1, both using the Gated non-linearity.
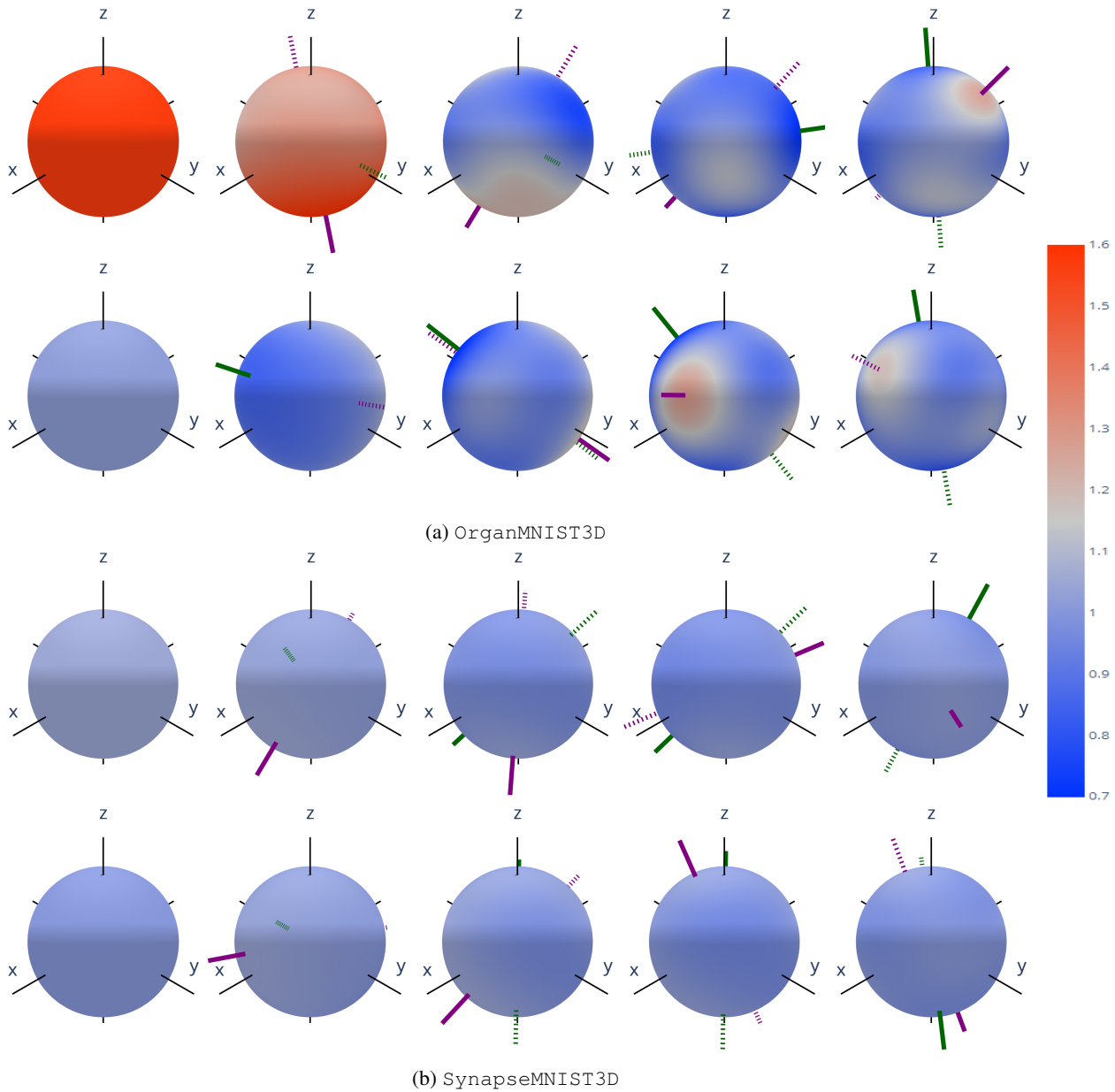
(a) `OrganMNIST3D`

(b) `SynapseMNIST3D`

Figure 15. 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.

Table 16 contains the MSE regression errors for both configurations. Figure 16 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 16 to Figure 11b and 11d

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 11b also contains higher frequencies. This more closely follows our expectations regarding the symmetries of the tasks in this dataset described in Section F.1.1.

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

Table 16. 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.
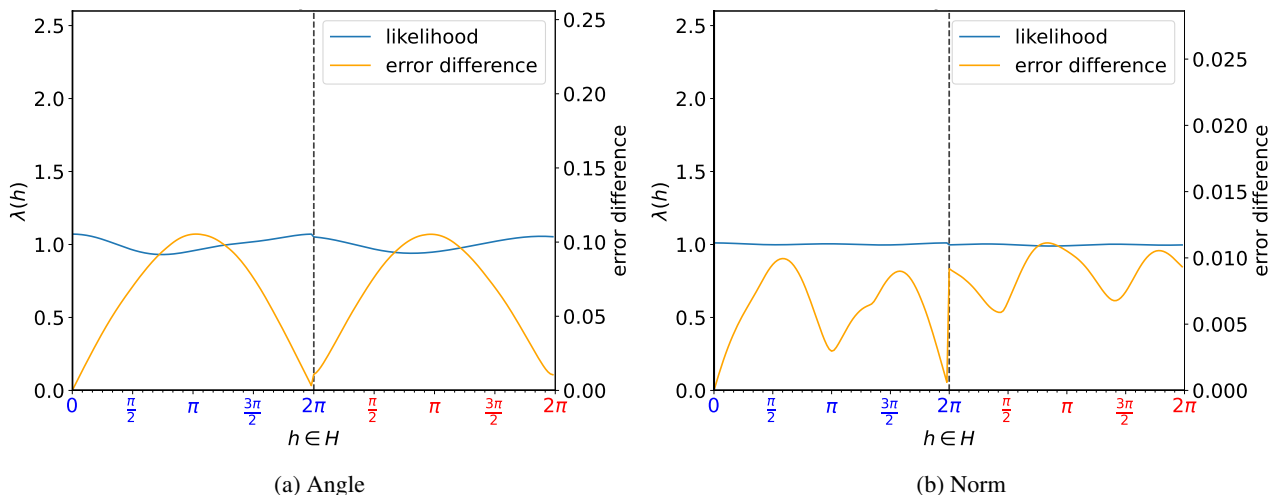

(a) Angle (b) Norm

Figure 16. 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.

G.5. Investigating the Effect of Regularisation Methods

In Section 3.2, 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 G.5.1 and G.5.2.

Additionally, in Section 3.3, we proposed to bandlimit the likelihood distribution as further method of regularisation. As described in Section F.2, all previous SCNN experiments used a bandlimit of $L = 2$. Section G.5.3 explores the effect of higher and lower levels of bandlimiting on the performance and degrees of equivariance.

G.5.1. 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 17 and 18, respectively. Furthermore, Figure 17 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)

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

Table 17. 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.

Table 18. 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.
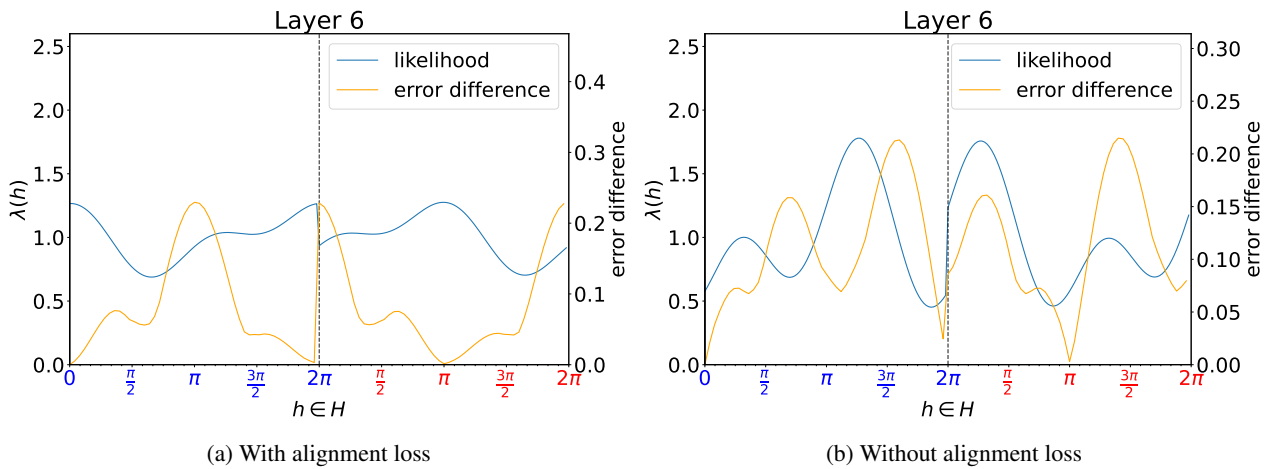

(a) With alignment loss

(b) Without alignment loss

Figure 17. 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 17 and 18 show a moderate increase in performance from the addition of alignment loss in the training objective. More importantly, Figure 17 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.

G.5.2. KL-DIVERGENCE

Similarly to the previous section, we present our quantitative results for `Vectors` and `Double MNIST` with $O(2)$ symmetries in Tables 19 and 20 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 18 and 19 respectively.

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

Table 19. 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 20. 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
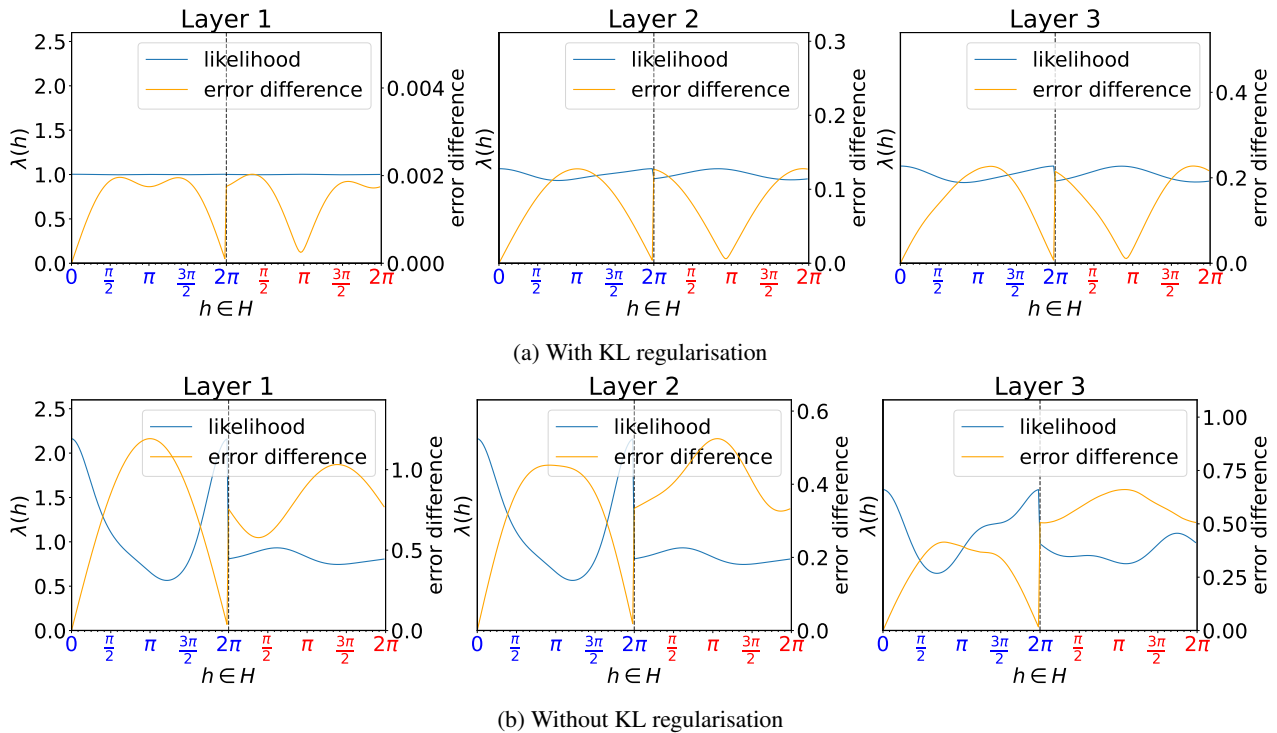

(b) Without KL regularisation

Figure 18. 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 19 and 20, 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 18 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 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 20. 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,
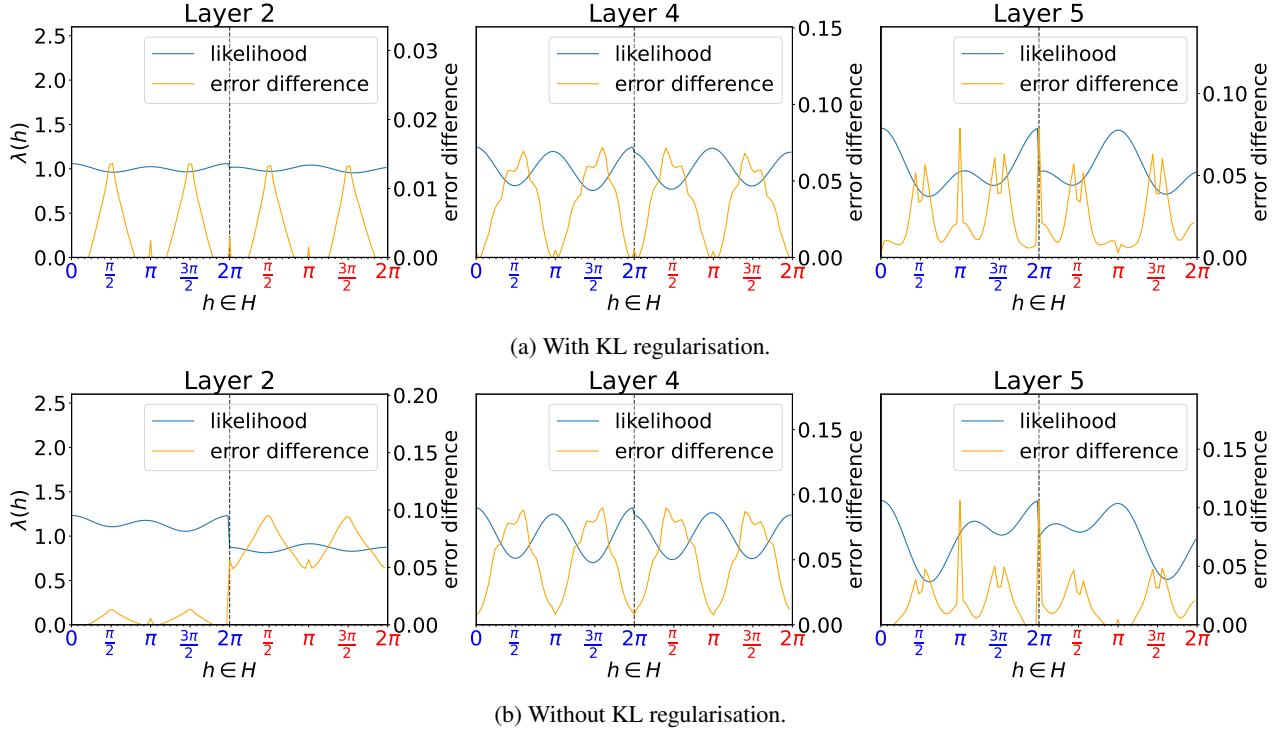
(a) With KL regularisation.

(b) Without KL regularisation.

Figure 19. 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.

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.

Figure 20. Development of KL-divergence during training with and without KL-regularisation.

For `Double MNIST`, Figure 19 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.
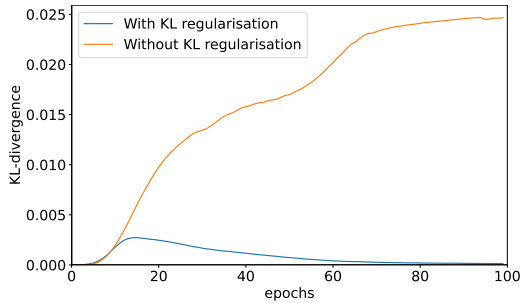
G.5.3. BANDLIMITING

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`, the subsets of `MedMNIST` and `Smoke`. We present the results in Tables 21 and 22. 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 21. Furthermore, due to the rapid increase in complexity, the highest level of bandlimiting for $O(3)$ is set at $L = 2$.

Double MNIST Table 21 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 | |
		$SO(2)$	$O(2)$
	None	0.474 (0.016)	0.403 (0.021)
$SO(2)$	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)
	None	0.391 (0.012)	0.399 (0.014)
$O(2)$	0	0.469 (0.010)	0.402 (0.003)
	1	<u>0.894</u> (0.011)	0.780 (0.009)
	2	<u>0.894</u> (0.004)	<u>0.819</u> (0.010)
	3	0.889 (0.013)	0.817 (0.007)
	4	0.891 (0.006)	<u>0.819</u> (0.018)

Table 21. `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 <u>underlined</u>. Standard deviations over 5 runs are denoted in parentheses.
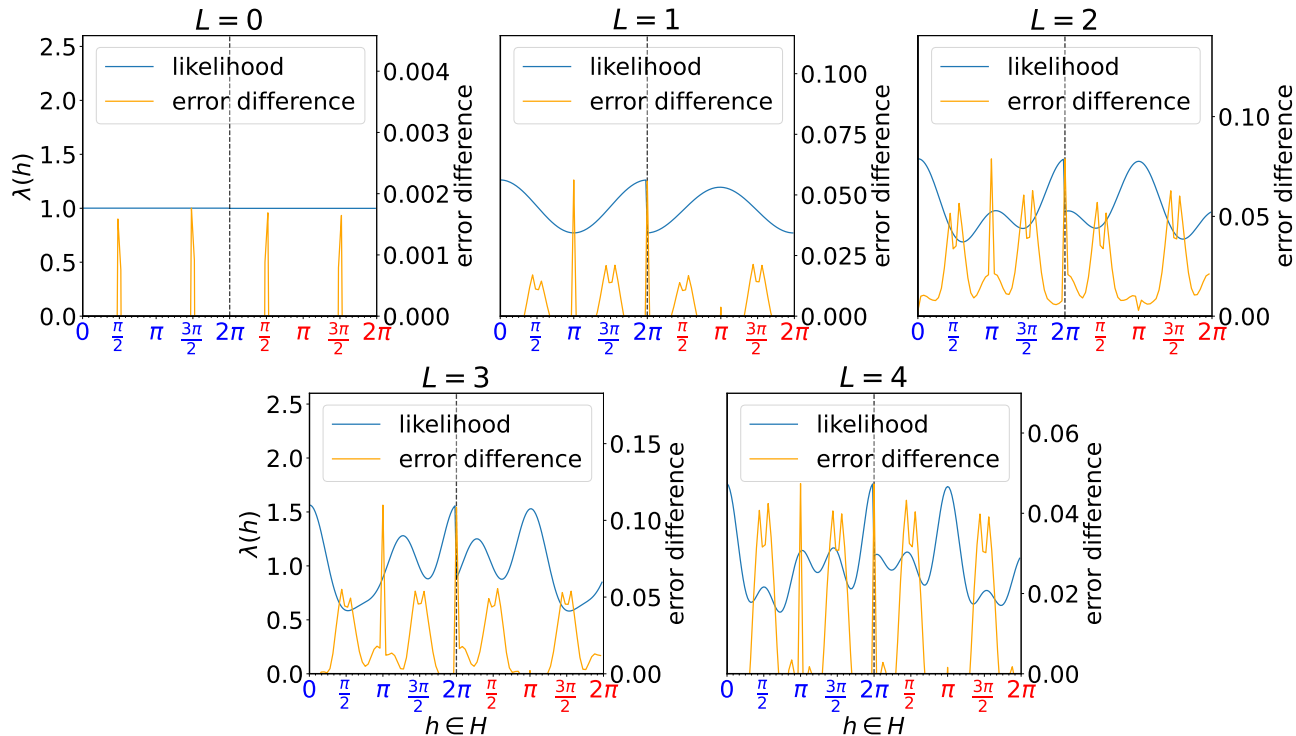

Figure 21. Likelihoods of the fifth layer of our $O(2)$ P-SCNN trained on $O(2)$ `Double MNIST` using a bandlimit of $L = 0$ through $L = 4$. Note that the scale of the equivariance error varies between the plots.

Under $O(2)$ symmetries, there is a more substantial performance benefit going from $L = 1$ to $L = 2$. Figure 21 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 21.

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 21 shows a significantly more complex and, therefore, less easily interpretable likelihood distribution.

MedMNIST The results in Table 22 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
	None	0.738 (0.009)	<u>0.873</u> (0.005)	0.607 (0.006)
$SO(3)$	1	0.753 (0.018)	<u>0.873</u> (0.007)	0.862 (0.012)
	2	0.770 (0.030)	0.871 (0.001)	<u>0.902</u> (0.006)
	3	**<u>0.771</u>** (0.030)	0.872 (0.010)	<u>0.902</u> (0.002)
	4	0.750 (0.006)	0.867 (0.009)	0.896 (0.003)
	None	0.743 (0.004)	0.868 (0.009)	0.592 (0.008)
$O(3)$	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	<u>0.769</u> (0.013)	0.873 (0.002)	**0.905** (0.004)

Table 22. `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 <u>underlined</u>. Standard deviations over 5 runs are denoted in parentheses.

Smoke While the main experiments on the `Smoke` dataset are performed using (partial) C_4 equivariance, which does not require band-limiting in our approach, we perform additional experiments experiments using $SO(2)$ equivariance using both our approach and the RSteer approach from Wang et al. (2022b) by changing the feature type to irrep fields up to frequency 2. The results can be found in Tab. 23. Here we present the task-specific performance, as well as the number of parameters for each configuration. Additionally, we also report the number of parameters for a set-up using larger 5×5 kernels instead of the default 3×3. In terms task-performance we observe that for our approach, as long as $L > 1$, the performance is nearly identical to RSteer for the various levels of band-limiting, while outperforming RPP. Furthermore, from the number of parameters we see that for the default set-up with 3×3 kernels our approach without band-limiting ($L = 6$) achieves a comparable number of parameters with RPP and RSteer, with noticeably lower parameter counts for $L \leq 2$. However, when using 5×5 kernels our approach uses significantly fewer parameters compared to both RPP and RSteer regardless of the band-limit L.

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. Furthermore, band-limiting is not necessarily required to obtain a reasonable parameter space compared to other partially equivariant approaches.

G.6. 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

Model	L	RMSE Future	RMSE Domain	Params (M) using default 3×3 kernels	Hypothetical params (M) using 5×5 kernels
RPP	-	0.81 (0.01)	0.70 (0.04)	6.43	17.24
RSteer	-	**0.78** (0.01)	**0.58** (0.00)	5.57	27.75
Ours	1	0.85 (0.01)	0.63 (0.00)	2.80	5.05
	2	**0.78** (0.01)	**0.58** (0.02)	4.30	7.84
	4	0.79 (0.02)	0.61 (0.03)	5.95	10.83
	6	**0.78** (0.01)	0.59 (0.03)	6.36	11.65

Table 23. `Smoke` RMSE scores and parameter counts (in millions, M) comparing $SO(2)$ equivariant RSteer with our $SO(2)$-PSCNN using various levels of band-limiting. Note that a band-limit of $L = 6$ equates to performing no band-limiting at all. Standard deviations over 5 runs are denoted in parentheses.

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 E.1, 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 24, 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)	-	**0.885** (0.003)	0.817 (0.007)
		4	0.911 (0.003)	-	0.876 (0.006)	<u>0.819</u> (0.018)

Table 24. 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.