Geometric deep learning and equivariant neural networks

N.B. When citing this work, cite the original published paper.

(article starts on next page)

Check for
updates

Geometric deep learning and equivariant neural networks

Jan E. Gerken[1,2,3] · **Jimmy Aronsson**[1] · **Oscar Carlsson**[1] · **Hampus Linander**[4] ·
Fredrik Ohlsson[5] · **Christoffer Petersson**[1,6] · **Daniel Persson**[1]

Abstract
We survey the mathematical foundations of geometric deep learning, focusing on group equivariant and gauge equivariant neural networks. We develop gauge equivariant convolutional neural networks on arbitrary manifolds $\mathcal{M}$ using principal bundles with structure group K and equivariant maps between sections of associated vector bundles. We also discuss group equivariant neural networks for homogeneous spaces $\mathcal{M} = G/K$, which are instead equivariant with respect to the global symmetry G on $\mathcal{M}$. Group equivariant layers can be interpreted as intertwiners between induced representations of G, and we show their relation to gauge equivariant convolutional layers. We analyze several applications of this formalism, including semantic segmentation and object detection networks. We also discuss the case of spherical networks in great detail, corresponding to the case $\mathcal{M} = S^2 = \mathrm{SO}(3)/\mathrm{SO}(2)$. Here we emphasize the use of Fourier analysis involving Wigner matrices, spherical harmonics and Clebsch–Gordan coefficients for $G = \mathrm{SO}(3)$, illustrating the power of representation theory for deep learning.

1 Introduction

Deep learning is an approach to machine learning that uses multiple transformation layers to extract hierarchical features and learn descriptive representations of the input data. These learned features can be applied to a wide variety of classification and regression tasks. Deep learning has for example been enormously successful in tasks such as computer vision, speech recognition and language processing. However, despite the overwhelming success of deep neural networks we are still at a loss for explaining exactly why deep learning works so well. One way to address this is to explore the underlying mathematical framework. A promising direction is to consider symmetries as a fundamental design principle for network architectures. This can be implemented by constructing deep neural networks that are compatible with a symmetry group G that acts transitively on the input data. This is directly relevant for instance in the case of spherical signals where G is a rotation group. In practical applications,

---

J. Aronsson and O. Carlsson have contributed equally to this study.

✉ Daniel Persson
daniel.persson@chalmers.se

Extended author information available on the last page of the article

it was found that equivariance improves per-sample efficiency, reducing the need for data augmentation (Müller et al. 2021). For linear models, this has been proven mathematically (Elesedy and Zaidi 2021).

Even more generally, it is natural to consider the question of how to train neural networks in the case of "non-flat" data. Relevant applications include fisheye cameras (Coors et al. 2018), biomedicine (Boomsma and Frellsen 2017; Elaldi et al. 2021), and cosmological data (Perraudin et al. 2019), just to mention a few situations where the data is naturally curved. Mathematically, this calls for developing a theory of deep learning on manifolds, or even more exotic structures, like graphs or algebraic varieties. This rapidly growing research field is referred to as *geometric deep learning* (Bronstein et al. 2017).

In this introduction we shall provide a birds-eye view on the subject of geometric deep learning, with emphasis on the mathematical foundations. We will gradually build up the formalism, starting from a simple semantic segmentation model which already illustrates the role of symmetries in neural networks. We discuss group and gauge equivariant convolutional neural networks, which play a leading role in the paper. The introduction concludes with a summary of our main results, a survey of related literature, as well as an outline of the paper.
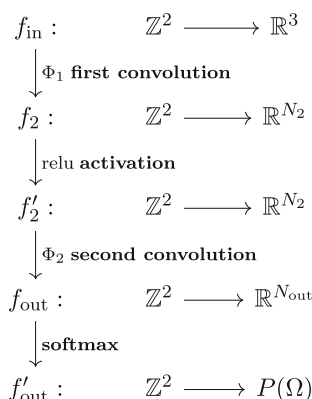
1.1 Warm up: a semantic segmentation model

The basic idea of deep learning is that the learning process takes place in multi-layer networks known as deep neural networks of "artificial neurons", where each layer receives data from the preceding layer and processes it before sending it to the subsequent layer. Suppose one wishes to categorize some data sample x according to which class y it belongs to. As a simple example, the input sample x could be an image and the output y could be a binary classification of whether a dog or a cat is present in the image. The first layers of a deep neural network would learn some basic low-level features, such as edges and contours, which are then transferred as input to the subsequent layers. These layers then learn more sophisticated high-level features, such as combinations of edges representing legs and ears. The learning process takes place in the sequence of hidden layers, until finally producing an output $\hat{y}$, to be compared with the correct image class y. The better the learning algorithm, the closer the neural network predictions $\hat{y}$ will be to y on new data samples it has not trained on. In short, one wishes to minimize a *loss function* that measures the difference between the output $\hat{y}$ and the class y.

More abstractly, let us view a neural network as a nonlinear map $\mathcal{N}$ between a set of input variables X and output variables $Y \supseteq \mathcal{N}(X)$. Suppose one performs a transformation T of the input data. This could for instance correspond to a translation or rotation of the elements in X. The neural network is said to be *equivariant* to the transformation T if it satisfies

$$\mathcal{N}(Tx) = T'\mathcal{N}(x), \tag{1}$$

for any input element x and some transformation T' acting on Y. A special case of this is that the transformation T' is the identity in which case the network is simply *invariant* under the transformation, i.e. $\mathcal{N}(Tx) = \mathcal{N}(x)$. This is for instance the case of convolutional neural networks used for image classification problems, for which we have invariance with respect to translations of the image. A prototypical example of a problem which requires true *equivariance* is the the commonly encountered problem of semantic segmentation in computer vision. Intuitively, this follows since the output is a pixel-wise segmentation mask which must transform in the same way as the input image. In the remainder of this section

$$f_{\text{in}} : \qquad \mathbb{Z}^2 \longrightarrow \mathbb{R}^3$$

$\Big\downarrow \Phi_1$ **first convolution**

$$f_2 : \qquad \mathbb{Z}^2 \longrightarrow \mathbb{R}^{N_2}$$

$\Big\downarrow$ relu **activation**

$$f_2' : \qquad \mathbb{Z}^2 \longrightarrow \mathbb{R}^{N_2}$$

$\Big\downarrow \Phi_2$ **second convolution**

$$f_{\text{out}} : \qquad \mathbb{Z}^2 \longrightarrow \mathbb{R}^{N_{\text{out}}}$$

$\Big\downarrow$ **softmax**

$$f_{\text{out}}' : \qquad \mathbb{Z}^2 \longrightarrow P(\Omega)$$

Fig. 1 Semantic segmentation model. In the first row the name of the first feature map (i.e. the input image) is given by f_{in}, it maps the domain $\mathbb{Z}^2$ to RGB values and has support on $[0, d_1] \times [0, d_2]$. It is followed by the first convolution, resulting in a feature map f_2 that now maps into a N_2 dimensional space corresponding to the N_2 filters of the convolution. After a point-wise activation the second convolution results in a feature map f_{out} that associates an N_{out}-dimensional vector to each point in the domain. This vector is then mapped to a probability distribution over the N_{out} classes using the softmax operator

we will therefore discuss such a model and highlight its equivariance properties in order to set the stage for later developments.

An image can be viewed as a compactly supported function $f : \mathbb{Z}^2 \to \mathbb{R}^{N_c}$, where $\mathbb{Z}^2$ represents the pixel grid and $\mathbb{R}^{N_c}$ the color space. For example, the case of $N_c = 1$ corresponds to a gray scale image while $N_c = 3$ can represent a color RGB image. Even though values in color space are typically restricted, for example grayscale values between 0 and 1, the color channel vectors of input data can be viewed as elements of the larger space $\mathbb{R}^{N_c}$. Analogous to images, a feature map f_i associated with layer i in a CNN can be viewed as a map $\mathbb{Z}^2 \to \mathbb{R}^{N_i}$, where $\mathbb{R}^{N_i}$ is the space of feature representations.

Consider a neural network $\mathcal{N}$ classifying each individual pixel of RGB images $f_{\text{in}} : \mathbb{Z}^2 \to \mathbb{R}^3$, supported on $[0, d_1] \times [0, d_2] \subset \mathbb{Z}^2$, into N_{out} classes using a convolutional neural network. Let the sample space of N_{out} classes be denoted Ω and let $P(\Omega)$ denote the space of probability distributions over Ω.

The network as a whole can be viewed as a map

$$\mathcal{N} : L^2(\mathbb{Z}^2, \mathbb{R}^3) \to L^2\left(\mathbb{Z}^2, P(\Omega)\right), \tag{2}$$

where $L^2(X, Y)$ denote the space of square integrable functions with domain X and co-domain Y. This class of functions ensures well-defined convolutions and the possibility to construct standard loss functions.

The co-domain $L^2\left(\mathbb{Z}^2, P(\Omega)\right)$ of $\mathcal{N}$ is usually referred to as semantic segmentations since an element assigns a semantic class probability distribution to every pixel in an input image.

For simplicity, let the model consist of two convolutional layers where the output of the last layer, f_{out}, maps into an N_{out}-dimensional vector space, followed by a softmax operator to produce a probability distribution over the classes for every element in $[0, d_1] \times [0, d_2]$. See Fig. 1 for an overview of the spaces involved for the semantic segmentation model.

The standard 2d convolution, for example Φ_1 in Fig. 1, is given by

$$[\kappa_1 \star f_{\text{in}}] (x, y) = \sum_{(x', y') \in \mathbb{Z}^2} \kappa_1(x' - x, y' - y) f_{\text{in}}(x', y'), \tag{3}$$

where $\kappa_1 \in L^2(\mathbb{Z}^2, \mathbb{R}^{N_2 \times 3})$ is the convolution kernel for Φ_1. This is formally a cross-correlation, but it can be transformed into a convolution by redefining the kernel.

We can rewrite this convolution group theoretically as

$$[\kappa_1 \star f_{\text{in}}](x, y) = \sum_{(x', y') \in \mathbb{Z}^2} L_{(x,y)} \kappa_1(x', y') f_{\text{in}}(x', y'), \tag{4}$$

where $L_{(x,y)}$ is the left-translation operator

$$L_{(x,y)} \kappa(x', y') = \kappa(x' - x, y' - y). \tag{5}$$

In the context of convolutions the terms kernel and filter appear with slight variations in their precise meaning and mutual relationship in the literature. We will generally use them interchangeably throughout this paper.

This convolution is equivariant with respect to translations $(x, y) \in \mathbb{Z}^2$, i.e.

$$\left[\kappa \star \left(L_{(x,y)} f_{\text{in}}\right)\right](x', y') = \left(L_{(x,y)} [\kappa \star f_{\text{in}}]\right)(x', y'). \tag{6}$$

The point-wise activation function and softmax operator also satisfy this property,

$$\left[\text{relu}\left(L_{(x,y)} f_{\text{in}}\right)\right](x', y') = \text{relu}\left(f_{\text{in}}(x' - x, y' - y)\right) = \left[L_{(x,y)} \text{relu}(f_{\text{in}})\right](x', y'), \tag{7}$$

where $\text{relu}(x) = \max\{0, x\}$, so that the model as a whole is equivariant under translations in $\mathbb{Z}^2$. Note that this equivariance of the model ensures that a translated image produces the corresponding translated segmentation,

$$\mathcal{N}(L_{(x,y)} f_{\text{in}}) = L_{(x,y)} \mathcal{N}(f_{\text{in}}), \tag{8}$$

as illustrated in Fig. 2. The layers in this particular model turn out to be equivariant with respect to translations but there are many examples of non-equivariant layers such as max pooling. Exactly what restrictions equivariance implies for a layer in an artificial neural network is the topic of Sects. 2 and 3.

## 1.2 Group equivariant convolutional neural networks

Convolutional neural networks are ordinary feed-forward networks that make use of convolutional operations of the form (3). One of the main reasons for their power is their aforementioned *translation equivariance* (8), which implies that a translation of the pixels in an image produces an overall translation of the convolution. Since each layer is translation equivariant all representations will be translated when the input data is translated. Furthermore, the local support of the convolution allows for efficient weight sharing across the input data.

Notice that $\mathbb{Z}^2$ in the semantic segmentation model is a group with respect to addition, and the space of feature representations $\mathbb{R}^N$ is a vector space. It is therefore natural to generalize this construction by replacing $\mathbb{Z}^2$ with an arbitrary group G and $\mathbb{R}^N$ with a vector space V. The feature map then generalizes to

$$f : G \to V, \tag{9}$$

and the convolution operation (3) to

$$[\kappa \star f](g) = \int_G \kappa(g^{-1}h) f(h) \mathrm{d}h, \tag{10}$$
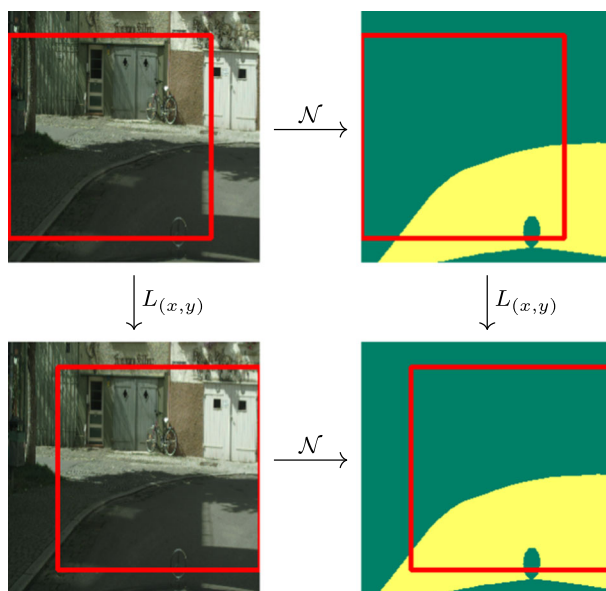
Fig. 2 $\mathbb{Z}^2$ equivariance of a semantic segmentation model classifying pixels into classes $\Omega = \{\text{road, non-road}\}$. The network $\mathcal{N}$ maps input images, indicated by the content of the red rectangles in the left column, to semantic masks indicated by the corresponding content of the red rectangles in the right column. Image and semantic mask from Cordts et al. (2016)

where dh is a left-invariant Haar measure on G. If G is a discrete group such as $\mathbb{Z}^2$, then dh becomes the counting measure and the integral reduces to a sum.

The generalized kernel $\kappa : G \to \text{Hom}(V, W)$ appearing in (10) is a function from the group to homomorphisms between V and some feature vector space W, which can in general be different from V. Consequently, the result of the convolution is another feature map

$$[\kappa \star f] : G \to W, \tag{11}$$

and in analogy with the terminology for ordinary CNNs we refer to the convolution (10) itself as a layer in the network. The general form of the convolution (10) is equivariant with respect to the left-translation by G:

$$[\kappa \star L_h f](g) = L_h [\kappa \star f](g), \tag{12}$$

motivating the term equivariant layer.

In the convolution (10), the kernel κ in the integral over G is transported in the group using the right action of G on itself. This transport corresponds to the translation of the convolutional kernel in (3) and generalizes the weight sharing in the case of a locally supported kernel κ.

In this paper we will explore the structure of group equivariant convolutional neural networks and their further generalizations to manifolds. A key step in this direction is to expose the connection with the theory of fiber bundles as well as the representation theory of G. To this end we shall now proceed to discuss this point of view.

It is natural to generalize the above construction even more by introducing a choice of subgroup[1] $K \leq G$ for each feature map, and a choice of representation ρ of K, i.e.,

$$\rho : K \to \mathrm{GL}(V), \tag{13}$$

where V is a vector space, and $\mathrm{GL}(V)$ denotes the general linear group on V. Consider then the coset space G/K and a vector bundle $E \xrightarrow{\pi} G/K$ with characteristic fiber V. Here, π is a continuous surjection that is often referred to as the projection of E down to the base space G/K. Sections of E are maps $s : G/K \to E$ which locally can be represented by vector-valued functions

$$f : G/K \to V. \tag{14}$$

These maps can be identified with the feature maps of a group equivariant convolutional neural network. Indeed, in the special case when $G = \mathbb{Z}^2$, K is trivial and $V = \mathbb{R}^N$, we recover the ordinary feature maps $f : \mathbb{Z}^2 \to \mathbb{R}^N$ of a CNN. When the representation ρ is non-trivial the network is called *steerable* (see Weiler et al. (2018); Weiler and Cesa (2019)).

As an example, consider spherical signals, i.e. the case in which the input feature map is defined on the two-sphere S^2 and can therefore be written in the form (14), since $S^2 \simeq \mathrm{SO}(3)/\mathrm{SO}(2)$. Here $\mathrm{SO}(n)$ denotes the special orthogonal group of $n \times n$ orthogonal matrices of unit determinant. One way to think of this quotient is to construct points on the sphere by using proper Euler angles $Z(\alpha)X(\beta)Z(\gamma) \in \mathrm{SO}(3)$ to rotate the north pole. The planar rotation $Z(\gamma) \in \mathrm{SO}(2)$ stabilizes the north pole and so the resulting point on the sphere only depends on $Z(\alpha)X(\beta)$, the angles functioning similarly to spherical coordinates (Grafarend and Kühnel 2011). Consequently, feature maps correspond to sections $f : \mathrm{SO}(3)/\mathrm{SO}(2) \to V$.

This construction allows us to describe G-equivariant CNNs using a very general mathematical framework. The space of feature maps is identified with the space of sections $\Gamma(E)$, while maps between feature maps, which we refer to as layers, belong to the space of so called G-equivariant intertwiners if they are equivariant with respect to the right action of G on the bundle E. This implies that many properties of group equivariant CNNs can be understood using the representation theory of G.

1.3 Group theory properties of machine learning tasks

After having laid out the basic idea of group equivariant neural networks, in this section we will make this more concrete by discussing the group theoretical properties of the common computer vision tasks of image classification, semantic segmentation and object detection.

Let us first focus on the case of classification. In this situation the input data consists of color values of pixels and therefore the input feature map f_{in} transforms under the regular representation π_{reg} of G, i.e. as a collection of scalar fields:

$$f_{\mathrm{in}}(x) \to \left[\pi_{\mathrm{reg}}(g)f_{\mathrm{in}}\right](x) = f_{\mathrm{in}}(\sigma^{-1}(g)x), \qquad g \in G, \tag{15}$$

where σ is a representation of G that dictates the transformation of the image. In other words, the color channels are not mixed by the spatial transformations. However, for image classification the identification of images should be completely independent of how they are transformed. For this reason we expect the network $\mathcal{N}(f_{\mathrm{in}})$ to be *invariant* under G,

$$\mathcal{N}(\pi_{\mathrm{reg}}(g)f_{\mathrm{in}}) = \mathcal{N}(f_{\mathrm{in}}), \qquad g \in G. \tag{16}$$

---

[1] We use the symbol $\leq$ to signify that K is a subgroup and not only a subset.

On the other hand, when doing semantic segmentation we are effectively classifying each individual pixel, giving a *segmentation mask* for the objects we wish to identify, as described in Sect. 1.1. This implies that the output features must transform in the same way as the input image. In this case one should not demand invariance of $\mathcal{N}$, but rather non-trivial *equivariance*

$$\mathcal{N}(\pi_{\text{reg}}(g)f_{\text{in}}) = \pi_{\text{reg}}(g)[\mathcal{N}(f_{\text{in}})], \qquad g \in G, \tag{17}$$

where the regular representation π_{reg} on the right-hand side transforms the output feature map of the network. The group-theoretic aspects of semantic segmentation are further explored in Sect. 5.

Object detection is a slightly more complicated task. The output in this case consists of bounding boxes around the objects present in the image together with class labels. We may view this as a generalization of the semantic segmentation, such that, for each pixel, we get a class probability vector $p \in \mathbb{R}^N$ (one of the classes labels the background) and three vectors $a, v_1, v_2 \in \mathbb{R}^2$ that indicate the pixel-position of the upper-left corner and two vectors that span the parallelogram of the associated bounding box. Hence, the output is an object $(p, a, v_1, v_2) \in \mathbb{R}^{N+6}$ for each pixel. The first N components of the output feature map $f : \mathbb{R}^2 \to \mathbb{R}^{N+6}$ transform as scalars as before. The three vectors a, v_1, v_2 on the other hand transform in a non-trivial two-dimensional representation ρ of G. The output feature map $f_{\text{out}} = \mathcal{N}(f_{\text{in}})$ hence transforms in the representation π_{out} according to

$$(\pi_{\text{out}}f_{\text{out}})(x) = \rho_{\text{out}}(g)f_{\text{out}}(\sigma^{-1}(g)x), \tag{18}$$

where $\rho_{\text{out}} = \text{id}_N \oplus \rho \oplus \rho \oplus \rho$. The network is then equivariant with respect to π_{reg} in the input and π_{out} in the output if

$$\mathcal{N}(\pi_{\text{reg}}(g)f_{\text{in}}) = \pi_{\text{out}}(g)[\mathcal{N}(f_{\text{in}})]. \tag{19}$$

For more details on equivariant object detection see Sects. 5.3 and 6.4.

1.4 Gauge equivariant networks

In the group equivariant networks discussed above, we exploited that the domain of the input feature map had global symmetries. Using inspiration from the physics of gauge theories and general relativity, the framework of equivariant layers can be extended to feature maps which are defined on a general manifold $\mathcal{M}$. A manifold can be thought of as consisting of a union of *charts*, giving rise to coordinates on $\mathcal{M}$, subject to suitable gluing conditions where the charts overlap. However, the choice of charts is arbitrary and tensors transform in a well-defined way under changes of charts. Such transformations are called *gauge transformations* in physics and correspond to the freedom of making local coordinate transformations across the manifold. Feature maps can in this context be viewed as sections of vector bundles associated to a principal bundle, called *fields* in physics parlance. A *gauge equivariant network* for such fields consists of layers which are equivariant with respect to change of coordinates in $\mathcal{M}$, such that the output of the network transforms like a tensor.

In order to realize gauge equivariant layers using convolutions, we need to shift the kernel across $\mathcal{M}$. In general, a manifold does not have any global symmetries to utilize for this. Instead, one may use *parallel transport* to move the filter on the manifold. This transport of features will generically depend on the chosen path. A gauge equivariant CNN is constructed precisely such that it is independent of the choice of path. In other words, making a coordinate transformation at $x \in \mathcal{M}$ and transporting the filter to $y \in \mathcal{M}$ should give the same result as

first transporting the filter from x to y, and then performing the coordinate transformation. Therefore, the resulting layers are gauge equivariant. The first steps toward a general theory of gauge equivariant convolutional neural networks on manifolds were taken in Cohen et al. (2019); Cheng et al. (2019).

1.5 Summary of results

This paper aims to put geometric aspects of deep learning into a mathematical context. Our intended audience includes mathematicians, theoretical physicists as well as mathematically minded machine learning experts.

The main contribution of this paper is to give a mathematical overview of the recent developments in group equivariant and gauge equivariant neural networks. We strive to develop the theory in a mathematical fashion, emphasizing the bundle perspective throughout. In contrast to most of the literature on the subject we start from the point of view of neural networks on arbitrary manifolds $\mathcal{M}$ (sometimes called "geometric deep learning"). This requires gauge equivariant networks which we develop using the gauge theoretic notions of principal bundles and associated vector bundles. Feature maps will be sections of associated vector bundles. These notions have been used previously in different equivariant architectures (Bronstein et al. 2021; Cheng et al. 2019; Cohen et al. 2019) and we present a unified picture. We analyze when maps between feature spaces are equivariant with respect to gauge transformations and define *gauge equivariant layers* accordingly in Sect. 2. Furthermore, we develop gauge equivariant convolutional layers for arbitrary principal bundles in Sect. 2.5 and thereby define gauge equivariant CNNs. In Sect. 2.5 we generalize the gauge equivariant convolution presented in Cheng et al. (2019) to the principal bundle setting.

Different principal bundles P describe different local (gauge) symmetries. One example of a local symmetry is the freedom to choose a basis in each tangent space or, in other words, the freedom to choose local frames of the frame bundle $P = \mathcal{L}\mathcal{M}$. In this case, local gauge transformations transform between different bases in tangent spaces. When the manifold is a homogeneous space $\mathcal{M} = G/K$, the global symmetry group forms a principal bundle $P = G$. Here, the local symmetry is the freedom to perform translations that do not move a given point, e.g. the north pole on S^2 being invariant to rotations about the z-axis, but we are more interested in the global translation symmetry for this bundle. Building on Aronsson (2022), we motivate group equivariant networks from the viewpoint of homogeneous vector bundles in Sect. 3 and connect these to the gauge equivariant networks. In Sect. 3.3, we discuss equivariance with respect to intensity; point-wise scaling of feature maps.

Furthermore, starting from a very general setup of a symmetry group acting on functions defined on topological spaces, we connect and unify various equivariant convolutions that are available in the literature.

Having developed the mathematical framework underlying equivariant neural networks, we give an overview of equivariant nonlinearities and in particular extend vector field non-linearities to arbitrary semi-direct product groups, cf. Proposition 5.3. We review the entire equivariant network architecture for semantic segmentation and object detection and the associated representations.

Finally, we consider spherical networks corresponding to data defined on the two-sphere $\mathcal{M} = S^2 = \mathrm{SO}(3)/\mathrm{SO}(2)$. For this case, we explain how convolutions can be computed in Fourier space and we give a detailed description of the convolution in terms of Wigner matrices and Clebsch-Gordan coefficients, involving in particular the decomposition of tensor products into irreducible representations of $\mathrm{SO}(3)$. This is well-known in the mathematical

physics community but we collect and present this material in a coherent way which we found was lacking in the literature. We illustrate the formalism in terms of object detection for the special Euclidean group SE(3), i.e. the group of (direct) Euclidean isometries of $\mathbb{R}^3$ (see Sect. 6.4).

1.6 Related literature

The notion of geometric deep learning was first discussed in the seminal paper of Bronstein, Bruna, LeCun and Szlam. Bronstein et al. (2017). They emphasized the need for neural networks defined on arbitrary data manifolds and graphs. In a different development, group equivariant convolutional neural networks, which incorporate global symmetries of the input data beyond the translational equivariance of ordinary CNNs, were proposed by Cohen and Welling (2016). Kondor and Trivedi (2018) proved that for compact groups G, a neural network architecture can be G-equivariant if and only if it is built out of convolutions of the form (10). The theory of equivariant neural networks on homogeneous spaces G/K was further formalized in Cohen et al. (2019) using the theory of vector bundles in conjunction with the representation theory of G. A proposal for including *attention* (Chen et al. 2018) into group equivariant CNNs was also put forward in Romero et al. (2020). Equivariant normalizing flows were recently constructed in Garcia Satorras et al. (2021).

The recent book (Bronstein et al. 2021) gives an in-depth overview of geometric deep learning. Our treatment is more mathematical than Bronstein et al. (2021), and we put stronger emphasis on the gauge equivariant formalism. The present paper may therefore be seen as complementary to Bronstein et al. (2021).

The case of neural networks on spheres has attracted considerable attention due to its extensive applicability. Group equivariant CNNs on S^2 were studied in Cohen et al. (2018) by implementing efficient Fourier analysis on $\mathcal{M} = S^2$ and $G = $ SO(3). In Gerken et al. (2022) the performance of group equivariant CNNs on S^2 was compared to standard non-equivariant CNNs trained with data augmentation. For the task of semantic segmentation it was demonstrated that the non-equivariant networks are consistently outperformed by the equivariant networks with considerably fewer parameters.

Some applications may benefit from equivariance with respect to azimuthal rotations, rather than arbitrary rotations in SO(3) (Toft et al. 2021). One such example is the use of neural networks in self-driving cars to identify vehicles and other objects.

The approach presented here follows (Cohen et al. 2018) and extends the results in the reference at some points. The Clebsch–Gordan nets introduced in Kondor et al. (2018) have a similar structure but use as nonlinearities tensor products in the Fourier domain, instead of point-wise nonlinearities in the spatial domain. Several modifications of this approach led to a more efficient implementation in Cobb et al. (2020). The constructions mentioned so far involve convolutions which map spherical features to features defined on SO(3). The construction in Esteves et al. (2018) on the other hand uses convolutions which map spherical features to spherical features, at the cost of restricting to isotropic filters. Isotropic filters on the sphere have also been realized by using graph convolutions in Defferrard et al. (2020). In Esteves et al. (2020), spin-weighted spherical harmonics are used to obtain anisotropic filters while still keeping the feature maps on the sphere.

A somewhat different approach to spherical signals is taken in Jiang et al. (2018), where a linear combination of differential operators acting on the input signal is evaluated. Although this construction is not equivariant, an equivariant version has been developed in Shen et al. (2021).

An important downside to many of the approaches outlined above is their poor scaling behavior in the resolution of the input image. To improve on this problem, McEwen et al. (2021) introduces scattering networks as an equivariant preprocessing step.

Aside from the equivariant approaches to spherical convolutions, much work has also been done on modifying flat convolutions in $\mathbb{R}^2$ to deal with the distortions in spherical data without imposing equivariance (Coors et al. 2018; Boomsma and Frellsen 2017; Su and Grauman 2017; Monroy et al. 2018).

A different approach to spherical CNNs was proposed in Cohen et al. (2019), were the basic premise is to treat S^2 as a manifold and use a gauge equivariant CNN, realized using the icosahedron as a discretization of the sphere. A general theory of gauge equivariant CNNs is discussed in Cheng et al. (2019). Further developments include gauge CNNs on meshes and grids (Haan et al. 2020; Wiersma et al. 2020) and applications to lattice gauge theories (Favoni et al. 2022; Luo et al. 2021). In this paper we continue to explore the mathematical structures underlying gauge equivariant CNNs and clarify their relation to GCNNs.

A further important case studied extensively in the literature are networks equivariant with respect to the Euclidean group E(n) or to SE(n). Such networks have been applied with great success to 3d shape classification (Weiler et al. 2018; Thomas et al. 2018), protein structure classification (Weiler et al. 2018), atomic potential prediction (Kondor 2018) and medical imaging (Müller et al. 2021; Worrall et al. 2017; Winkels and Cohen 2018).

The earliest papers in the direction of SE(n) equivariant network architectures extended classical GCNNs to 3d convolutions and discrete subgroups of SO(3) (Worrall et al. 2017; Winkels and Cohen 2018; Marcos et al. 2017; Weiler et al. 2018).

Our discussion of SE(n) equivariant networks is most similar to the SE(3) equivariant networks in Weiler et al. (2018), where an equivariance constraint on the convolution kernel of a standard 3d convolution is solved by expanding it in spherical harmonics. A similar approach was used earlier in Worrall et al. (2017) to construct SE(2) equivariant networks using circular harmonics. A comprehensive comparison of different architectures which are equivariant with respect to E(2) and various subgroups was given in Weiler and Cesa (2019).

Whereas the aforementioned papers specialize standard convolutions by imposing constraints on the filters and are therefore restricted to data on regular grids, Thomas et al. (2018); Kondor (2018) operate on irregular point clouds and make the positions of the points part of the features. These networks operate on non-trivially transforming input features and also expand the convolution kernels into spherical harmonics but use Clebsch–Gordan coefficients to combine representations.

So far, the applied part of the literature is mainly focused on specific groups (mostly rotations and translations in two and three dimensions and their subgroups). However, in the recent paper (Lang and Weiler 2020), a general approach to solve the kernel constraint for arbitrary compact groups is constructed by deriving a Wigner–Eckart theorem for equivariant convolution kernels. The implementation in Finzi et al. (2021) uses a different algorithm to solve the kernel constraint for matrix groups and allows to automatically construct equivariant CNN layers.

The review Esteves (2020) discusses various aspects of equivariant networks. The book Bronstein et al. (2021) gives an exhaustive survey of many of the developments related to geometric deep learning and equivariant CNNs.

It should be noted that it is by no means obvious that data domains exhibit symmetries, let alone are endowed with a manifold structure. In this paper we are focussing on situations where we do have such structures and how we may then use techniques and ideas from mathematics and theoretical physics to gain a deeper understanding of deep learning. That being said, it is natural to inquire about situations where this information is not available.

One possible direction is through the field of topological data analysis (TDA). This provides a framework to analyze the shape of data sets using techniques from topology. The key tool here is "persistent homology" which is an adaptation of homology to data in the form of point clouds. In the papers (Frosini and Jabłoński 2016; Bergomi et al. 2019; Conti et al. 2022) the authors introduce so called *group equivariant non-expansive operators* (GENEOs) in the context of TDA. This gives a different approach to the question of symmetries in neural networks, in which the topology and geometry of the data is not a priori given. Furthermore, cases where the set of transformations acting on the data do not form a group have been considered in the literature Bergomi et al. (2019), whereas we restrict our considerations to group transformations.

1.7 Outline of the paper

Our paper is structured as follows. In Sect. 2 we introduce gauge equivariant convolutional neural networks on manifolds. We discuss global versus local symmetries in neural networks. Associated vector bundles are introduced and maps between feature spaces are defined. Gauge equivariant CNNs are constructed using principal bundles over the manifold. We conclude Sect. 2 with a discussion of some concrete examples of how gauge equivariant CNNs can be implemented for neural networks on graphs. In Sect. 3 we restrict to homogeneous spaces $\mathcal{M} = G/K$ and introduce homogeneous vector bundles. We show that when restricting to homogeneous spaces the general framework of Sect. 2 gives rise to group equivariant convolutional neural networks with respect to the global symmetry G. We also introduce intensity equivariance and investigate its compatibility with group equivariance. Still restricting to global symmetries, Sect. 4 explores the form of the convolutional integral and the kernel constraints in various cases. Here, the starting point are vector valued maps between arbitrary topological spaces. This allows us to investigate convolutions between non-scalar features as well as non-transitive group actions. As an application of this we consider semi-direct product groups which are relevant for steerable neural networks. In Sect. 5 we assemble the pieces and discuss how one can construct equivariant deep architectures using our framework. To this end we begin by discussing how nonlinearities can be included in an equivariant setting. We further illustrate the group equivariant formalism by analyzing deep neural networks for semantic segmentation and object detection tasks. In Sect. 6 we provide a detailed analysis of spherical convolutions. Convolutions on S^2 can be computed using Fourier analysis with the aid of spherical harmonics and Wigner matrices. The output of the network is characterized through certain tensor products which decompose into irreducible representations of SO(3). In the final Sect. 7 we offer some conclusions and suggestions for future work.

2 Gauge equivariant convolutional layers

In this section we present the structure needed to discuss local transformations and symmetries on general manifolds. We also discuss the gauge equivariant convolution in Cheng et al. (2019) for features defined on a smooth manifold $\mathcal{M}$ along with lifting this into the principal bundle formalism. We end this section by expanding on two applications of convolutions on manifolds via a discretization to a mesh and compare these to the convolution on the smooth manifold.

2.1 Global and local symmetries

In physics the concept of symmetry is a central aspect when constructing new theories. An object is symmetric to a transformation if applying that transformation leaves the object as it started.

In any theory, the relevant symmetry transformations on a space form a *group K*. When the symmetry transformations act on vectors via linear transformations, we have a *representation* of the group. This is needed since an abstract group has no canonical action on a vector space; to allow the action of a group K on a vector space V, one must specify a representation. Formally a representation is a map $\rho : K \to \mathrm{GL}(\dim(V), F)$ into the space of all invertible $\dim(V) \times \dim(V)$ matrices over a field F. Unless otherwise stated, we use complex representations ($F = \mathbb{C}$). In contrast, real representations use $F = \mathbb{R}$. The representation needs to be a group homomorphism, i.e.

$$\rho(kk') = \rho(k)\rho(k'), \tag{20}$$

for all $k, k' \in K$. In particular, $\rho(k^{-1}) = \rho(k)^{-1}$ and $\rho(e) = \mathrm{id}_V$ where $e \in K$ is the identity element.

Remark 2.1 There are several ways to denote a representation and in this paper we use V_ρ to represent the representation ρ acting on some vector space, hence V_ρ and V_η will be viewed as two (possibly) different vector spaces acted on by ρ and η respectively.

Returning to symmetries there exists, in brief, two types of symmetries: *global* and *local*. An explicit example of a global transformation of a field $\phi : \mathbb{R}^2 \to \mathbb{R}^3$ is a rotation $R \in \mathrm{SO}(2)$ of the domain as

$$\phi(x) \xrightarrow{R} \phi'(x) = \rho(R)\phi(\eta(R^{-1})x), \tag{21}$$

where ρ is a representation for how $\mathrm{SO}(2)$ acts on $\mathbb{R}^3$ and η is the standard representation for how $\mathrm{SO}(2)$ acts on $\mathbb{R}^2$.

Remark 2.2 Note that this transformation not only transforms the vector $\phi(x)$ at each point $x \in \mathbb{R}^2$, but also moves the point x itself.

Example 2.3 If we let $R \in \mathrm{SO}(2)$ be the action of rotating with an angle R, then the standard representation of R would be

$$\eta(R) = \begin{pmatrix} \cos(R) & \sin(R) \\ -\sin(R) & \cos(R) \end{pmatrix}. \tag{22}$$

Example 2.4 An example of a rotationally symmetric object is when ϕ is three scalar fields, i.e. $\rho_3(R) = 1 \oplus 1 \oplus 1$, where each scalar field only depends on the distance from the origin. This yields

$$\phi'(x) = \rho_3(R)\phi(R^{-1}x) = \phi(R^{-1}x) = \phi(x), \tag{23}$$

since rotation of the domain around the origin leaves distances to the origin unchanged.

With the global transformation above we act with the same transformation on every point; with local transformations we are allowed to transform the object at each point differently. We can construct a similar explicit example of a local symmetry: Given a field $\phi : \mathbb{R}^2 \to \mathbb{C}$ we can define a local transformation as

$$\phi(x) \to \phi'(x) = \exp(if(x))\phi(x), \tag{24}$$

where $f : \mathbb{R}^2 \rightarrow [0, 2\pi)$. This is local in the sense that at each point x the field ϕ is transformed by $\exp(if(x))$ where f is allowed to vary over $\mathbb{R}^2$. We will refer to a local transformation as a *gauge transformation*. If an object is invariant under local (gauge) transformations, it is called *gauge invariant* or that it has a *gauge symmetry*. The group consisting of all local transformations is called the *gauge group*.

Remark 2.5 The example of a local transformation presented in (24) does not move the base point but in general there are local transformations that move the base point; the main example of which is locally defined diffeomorphisms which are heavily used in general relativity. In this section we will only consider transformations that do not move the base point and gauge transformations falls in this category. For a more detailed discussion on this see Remark 5.3.10 in Hamilton (2017).

Example 2.6 A simple example of a gauge invariant object using (24) is the field

$$\overline{\phi(x)}\phi(x), \tag{25}$$

where the bar denotes complex conjugate. This works since multiplication of complex numbers is commutative. This is an example of a commutative gauge symmetry.

Note that in the above example the phase of ϕ at each point can be transformed arbitrarily without affecting the field in (25), hence we have a redundancy in the phase of ϕ. For any object with a gauge symmetry one can get rid of the redundancy by choosing a specific gauge.

Example 2.7 The phase redundancy in the above example can be remedied by choosing a phase for each point. For example this can be done by

$$\phi(x) \rightarrow \phi'(x) = |\phi(x)| = \exp\left(-i \arg\left(\phi(x)\right)\right)\phi(x). \tag{26}$$

Thus ϕ' only takes real values at each point and since (25) is invariant to this transformation we have an equivalent object with real fields.

To introduce equivariance, let K be a group acting on two vector spaces V_ρ, V_η through representations ρ, η and let $\Phi : V_\rho \rightarrow V_\eta$ be a map. We say that Φ is *equivariant* with respect to K if for all $k \in K$ and $v \in V_\rho$,

$$\Phi(\rho(k)v) = \eta(k)\Phi(v), \tag{27}$$

or equivalently, with the representations left implicit, expressed $\Phi \circ k = k \circ \Phi$.

2.2 Motivation and intuition for the general formulation

When a neural network receives some numerical input, unless specified, it does not know what basis the data is expressed in, be it local or global. The goal of the general formulation presented in Sect. 2.3 is thus that if two numerical inputs are related by a transformation between equivalent states, e.g. by a change of basis, then the output from a layer should be related by the same transformation: if $u = k \triangleright v$ then $\phi(u) = k \triangleright \phi(v)$, where $k \triangleright$ is a the action of the group element through some representation on the input data v; in words ϕ is an equivariant map. The intuition for this is that it ensures that there is no basis dependence in the way ϕ acts on data.

To construct such a map ϕ we will construct objects which are gauge invariant but contain components that transform under gauge transformations. We then define a map Φ on those using ϕ to act on one of the components.
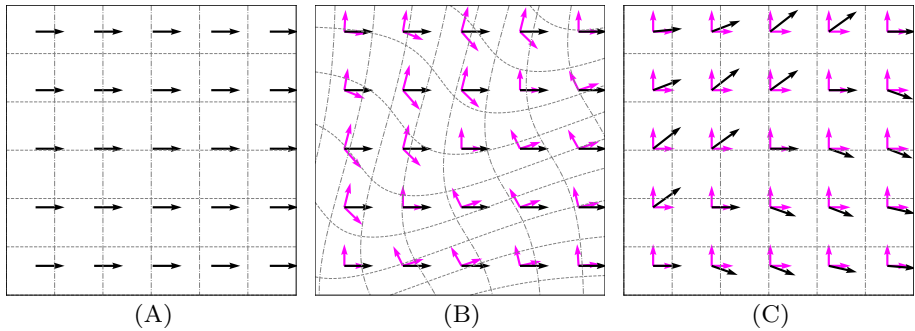
Fig. 3 A vector field on a base space $\mathcal{M}$ in different local coordinates. The local coordinates can be viewed either as induced by the coordinate system $u : \mathcal{M} \to \mathbb{R}^2$ or as a section $\omega : \mathcal{M} \to P$ of the frame bundle $P = \mathcal{L}\mathcal{M}$ that specifies a local basis at each point—see Sect. 2.3 for details. **A** The local basis for the vector field is at every point aligned with the standard basis in $\mathbb{R}^2$. **B** A new local basis induced by the coordinate system $u' : \mathcal{M} \to \mathbb{R}^2$, or equivalently as a section $\omega' : \mathcal{M} \to P$. The transition map is either $u' \circ u^{-1}$ or $\omega'(x) = \omega(x) \triangleleft \sigma(x)$ where $\sigma : \mathcal{M} \to K$ is a map from the base space to the gauge group K. **C** Here the vector field is at each point expressed in the new local coordinates. This illustrates that two vector fields can look very different when expressed in components, but when taking the local basis into account, they are the same

Our objects will be equivalence classes consisting of two elements: an element that specifies which gauge the numerical values are in and one element which are the numerical values. By construction these will transform in "opposite" ways and hence each equivalence class is gauge invariant. The intuition for the first element is that it will serve as book-keeping for the theoretical formulation.

The input to a neural network can then be interpreted as the numerical part of an equivalence class, and if two inputs are related by a gauge transformation the outputs from ϕ are also related by the same transformation. (Through whatever representation is chosen to act on the output space.)

Images can be thought of as compactly supported vector valued functions on $\mathbb{R}^2$ (or $\mathbb{Z}^2$ after discretization) but when going to a more general surface, a smooth d-dimensional manifold, the image needs instead to be interpreted as a section of a fiber bundle. In this case we cannot, in general, have a global transformation and we really need the local gauge viewpoint.

Example 2.8 If our image is an RGB image every pixel has a "color vector" and one could apply a random permutation of the color channels for each pixel. Applying a random SO(3) element would not quite work since each element of the "color vector" needs to lie in $[0, 255] \cap \mathbb{Z}$ and under a random SO(3) element a "color vector" could be rotated outside this allowable space.

Remark 2.9 As mentioned in the previous section there are local transformations that move base points locally, e.g. a diffeomorphism $\psi : \mathcal{M} \to \mathcal{M}$ such that $\psi(x) = x$ for some $x \in \mathcal{M}$. Note that this is not the same as a local change of coordinates at each point. In the neighborhood, and the tangent space, of x though one can view this as a local change of coordinates. Hence if one transforms a feature map at each point, the transformation of each feature vector can be viewed as a diffeomorphism centered at that point. See Fig. 4. A local coordinate chart on the manifold around a point would give the same result.
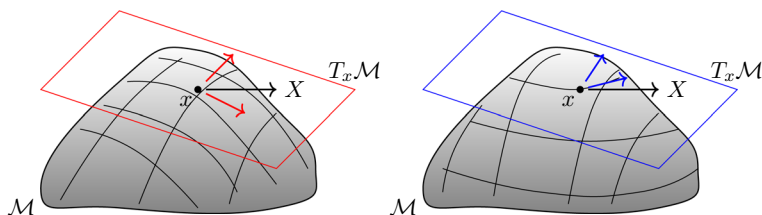
Fig. 4 The manifold $\mathcal{M}$ has a different choice of gauge—local coordinates on $\mathcal{M}$—at x in the left and the right figure leading to a different choice of basis in the corresponding tangent space $T_x\mathcal{M}$. This is the same process as in Fig. 3 where the difference is that in this case the base space is curved

2.3 General formulation

The construction presented here is based on the one used in Aronsson (2022). Since in the general case all transformations will be local, we need to formulate the theory in the language of fiber bundles. In short, a *bundle* $E \xrightarrow{\pi} \mathcal{M}$ is a triple $(E, \pi, \mathcal{M})$ consisting of a *total space* E and a surjective continuous projection map π onto a *base manifold* $\mathcal{M}$. Given a point $x \in \mathcal{M}$, the *fiber* over x is $\pi^{-1}(x) = \{v \in E : \pi(v) = x\}$ and is denoted E_x. If for every $x \in \mathcal{M}$ the fibers E_x are isomorphic the bundle is called a *fiber bundle*. Furthermore, a *section* of a bundle E is a map $\sigma : \mathcal{M} \to E$ such that $\pi \circ \sigma = \mathrm{id}_{\mathcal{M}}$ is the identity map on $\mathcal{M}$. We will use associated, vector, and principal bundles and will assume a passing familiarity with these, but will for completeness give a short overview. For more details see (Nakahara 2018; Kolář et al. 1993; Marsh 2019a).

In this section we follow the construction of Aronsson (2022) and begin by defining a principal bundle encoding some symmetry group K we want to incorporate into our network:

Definition 2.10 Let K be a Lie-group. A *principal K-bundle* over $\mathcal{M}$ is a fiber bundle $P \xrightarrow{\pi_P} \mathcal{M}$ with a fiber preserving, free and transitive right action of K on P,

$$\lhd : P \times K \to P, \quad \text{satisfying} \quad \pi_P(p \lhd k) = \pi_P(p), \tag{28}$$

and such that $p \lhd e = p$ for all $p \in P$ where e is the identity in K. As a consequence of the action being free and transitive we have that P_x and K are isomorphic as sets.

Remark 2.11 Note that even though P_x and K are isomorphic as sets we cannot view P_x as a group since there is no identity in P_x. To view the fiber P_x as isomorphic to the group we need to specify $p \in P_x$ as a reference-point. With this we can make the following map from K to P_x

$$k \mapsto p \lhd k, \tag{29}$$

which is an isomorphism for each fixed $p \in P_x$ since the group action on the fibers is free and transitive. We will refer to this choice of reference point as *choosing a gauge for the fiber P_x*.

Given a local patch $U \subseteq \mathcal{M}$, a local section $\omega : U \to P$ of the principal bundle P provides an element $\omega(x)$ which can be used as reference point in the fiber P_x, yielding a local trivialization

$$U \times K \to P, \quad (x, k) \mapsto \omega(x) \lhd k. \tag{30}$$

This is called choosing a *(local) gauge* on the patch $U \subseteq \mathcal{M}$.

Remark 2.12 For the case when $\mathcal{M}$ is a surface, i.e. is of (real) dimension 2, Melzi et al. (2019) presents a method for assigning a gauge (basis for the tangent space) to each point. This is done by letting one basis vector be the gradient of some suitable scalar function on $\mathcal{M}$ and the other being the cross product of the normal at that point with the gradient. This requires $\mathcal{M}$ to be embedded in $\mathbb{R}^3$.

If it is possible to choose a continuous global gauge ($U = \mathcal{M}$) then the principal bundle is trivial since (30) is then a global trivialization. On the other hand, if P is non-trivial and we allow $\omega : U \to P$ to be discontinuous we can always choose a reference-point independently for each fiber. Alternatively, we may choose a set of continuous sections $\{\omega_i\}$ whose domains U_i are open subsets covering $\mathcal{M}$. In the latter setting, if $x \in U_i \cap U_j$ then there is a unique element $k_x \in K$ relating the gauges ω_i and ω_j at x, such that $\omega_i(x) = \omega_j(x) \triangleleft k_x$.

Remark 2.13 As stated, a principal bundle is trivial if and only if there exists a global continuous gauge. Consequently, even if one has a covering $\{U_i\}$ of the manifold $\mathcal{M}$ and a local gauge ω_i for each U_i there is no way to combine these into a global continuous gauge unless the principal bundle is trivial. It is, however, possible to define a global section if one drops the continuity condition. In this paper will not implicitly assume that a chosen gauge is continuous unless specified.

Continuing in the same fashion we can view a local map $\sigma : U \subseteq \mathcal{M} \to K$ as changing the gauge at each point, or in other words a gauge transformation. With this we now define associated bundles:

Definition 2.14 Let $P \xrightarrow{\pi_P} \mathcal{M}$ be a principal K-bundle and let V be a vector space on which K acts from the left through some representation ρ

$$K \times V \to V, \quad k \triangleright v = \rho(k)v. \tag{31}$$

Now, consider the space $P \times V$ consisting of pairs (p, v) and define an equivalence relation on this space by

$$(p, v) \sim_\rho (p \triangleleft k, k^{-1} \triangleright v), \quad k \in K. \tag{32}$$

Denoting the resulting quotient space $P \times V / \sim_\rho$ by $E_\rho = P \times_\rho V_\rho$ and equipping it with a projection $\pi_\rho : E_\rho \to \mathcal{M}$ acting as $\pi_\rho([p, v]) = \pi_P(p)$ makes E_ρ a fiber bundle over $\mathcal{M}$ associated to P. Moreover, $\pi_\rho^{-1}(x)$ is isomorphic to V and thus $P \times_\rho V_\rho$ is a vector bundle over $\mathcal{M}$. The space of sections $\Gamma(E_\rho)$ is a vector space with respect to the point-wise addition and scalar multiplication of V.

Moving on we provide the definition of data points and feature maps in the setting of networks.

Definition 2.15 We define a *data point* $s \in \Gamma(E_\rho)$ as a continuous section of an associated bundle E_ρ, and a *feature map* $f \in C(P; \rho)$ as a continuous function $f : P \to V_\rho$ satisfying the following equivariance condition:

$$k \triangleright f(p) = f(p \triangleleft k^{-1}). \tag{33}$$

To connect the two notions we can use the following lemma.

Lemma 2.16 (Kolář et al. (1993)) *The linear map $\varphi_\rho : C(P; \rho) \to \Gamma(E_\rho)$, $f \mapsto s_f = [\pi_P^{-1}, f \circ \pi_P^{-1}]$, is a vector space isomorphism.*

Remark 2.17 With the notation $[\pi_P^{-1}, f \circ \pi_P^{-1}]$ we mean that when applying this to $x \in \mathcal{M}$ one first picks an element $p \in \pi_P^{-1}$ so that $s_f(x) = [p, f(p)]$. Note that this equality is well-defined since the equivalence class $[p, f(p)]$ does not depend on the choice of p in the fiber over x:

$$[p', f(p')] = [p \triangleleft k, f(p \triangleleft k)] = [p \triangleleft k, k^{-1} \triangleright f(p)] = [p, f(p)]. \qquad (34)$$

Remark 2.18 To present a quick argument for this note that with the equivalence class structure on the associated bundle every section $s_f \in \Gamma(E_\rho)$ is of the form $s_f(x) = [p, f(p)]$ where $p \in \pi_P^{-1}(x)$ since every element in the associated bundle consists of an element from the principal bundle and a vector from a vector space. Here we use f to specify which vector is used.

Before moving on to gauge equivariant layers we need to establish how a gauge transformation $\sigma : \mathcal{M} \to K$ acts on the data points.

Definition 2.19 Let $k \in K$ be an element of the gauge group and $[p, v]$ an element of the associated bundle $E_\rho = P \times_\rho V_\rho$. Then the action of k on $[p, v]$ is defined as

$$k \cdot [p, v] = [p \triangleleft k, v] = [p, k \triangleright v], \qquad (35)$$

which induces an action of $\sigma : U \subseteq \mathcal{M} \to K$ on $s_f = [\cdot, f]$ as

$$(\sigma \cdot s_f)(x) = \sigma(x) \cdot [p, f(p)] = [p, \sigma(x) \triangleright f(p)] = s_{\sigma \triangleright f}(x). \qquad (36)$$

We call σ a *gauge transformation*. If σ is constant on U it is called a *rigid gauge transformation*.

Remark 2.20 Note that (36) is really abuse of notation. To view σ as a map from region of the manifold $U \subseteq \mathcal{M}$ to the group K we first have to choose a gauge $\omega : U \to P$. This is identical to choosing a local trivialization of P and allows us to identify the fibers of P with K. For details on this see Hamilton (2017).

With this we now define layers as map between spaces of sections of associated vector bundles.

Definition 2.21 Let $E_\rho = P \times_\rho V_\rho$ and $E_\eta = P \times_\eta V_\eta$ be two associated bundles over a smooth manifold $\mathcal{M}$. Then a *layer* is a map $\Phi : \Gamma(E_\rho) \to \Gamma(E_\eta)$. Moreover, Φ is *gauge equivariant* if $\sigma \circ \Phi = \Phi \circ \sigma$ in the sense (36) for all gauge transformations σ.

Remark 2.22 Since the map $\varphi_\pi : C(P; \pi) \to \Gamma(E_\pi)$ is a vector space isomorphism for any representation π, any layer $\Phi : \Gamma(E_\rho) \to \Gamma(E_\eta)$ induces a unique map $\phi : C(P; \rho) \to C(P; \eta)$ by $\phi = \varphi_\eta^{-1} \circ \Phi \circ \varphi_\rho$, and vice versa; see the diagram below. We will refer to both Φ and ϕ as layers.

$$
\begin{array}{ccc}
\Gamma(E_\rho) & \xrightarrow{\ \Phi\ } & \Gamma(E_\eta) \\
{\scriptstyle \varphi_\rho} \Big\uparrow & & \Big\downarrow {\scriptstyle \varphi_\eta^{-1}} \\
C(P; \rho) & \dashrightarrow{\ \phi\ } & C(P; \eta)
\end{array}
$$

The equivariance property expressed explicitly in terms of ϕ, given a feature map f, is

$$k \triangleright (\phi f)(p) = \phi(k \triangleright f)(p), \qquad (37)$$

and since $\phi f \in C(P; \eta)$, the transformation property (33) yields the constraint

$$(\phi f)(p \triangleleft k^{-1}) = \phi(k \triangleright f)(p). \qquad (38)$$

2.4 Geometric symmetries on manifolds and the frame bundle

A special case of the general structure presented above is when the gauge symmetry is a freedom to choose a different coordinate patch $U \subset \mathcal{M}$ around each point in the manifold $\mathcal{M}$. This is also needed as soon as $\mathcal{M}$ is curved since it is impossible to impose a global consistent coordinate system on a general curved manifold. Because of this one is forced to work locally and use the fiber bundle approach.

In this section we will work within a coordinate patch $u : U \to \mathbb{R}^d$ around $x \in U$, and the i:th coordinate of x is denoted $y^i = u^i(x)$ being the i:th component of the vector $u \in \mathbb{R}^d$. A coordinate patch is sometimes denoted (u, U). This coordinate chart induces a basis for the tangent space $T_x\mathcal{M}$ as $\{\partial_1, \ldots, \partial_d\}$ such that any vector $v \in T_x\mathcal{M}$ can be written

$$v = \sum_{m=1}^{d} v^m \partial_m = v^m \partial_m, \tag{39}$$

where v^m are called the components of v and we are using the Einstein summation convention that indices repeated upstairs and downstairs are implicitly summed over. We will use this convention for the rest of this section.

Given a coordinate chart $u : U \to \mathbb{R}^d$ around $x \in U$ in which we denote coordinates y^i a change to a different coordinate chart $u' : U \to \mathbb{R}^d$ where we denote coordinates y'^i would be done through a map $u' \circ u^{-1} : \mathbb{R}^d \to \mathbb{R}^d$. This map can be expressed as a matrix $k^{-1} \in \mathrm{GL}(d)$ (we use k^{-1} for notational reasons) and in coordinates this transformation would be

$$y'^n = (k^{-1})^n{}_m y^m, \tag{40}$$

where $(k^{-1})^n{}_m$ should be interpreted as the element in the n:th row and m:th column. For a tangent vector $v \in T_x\mathcal{M}$ expressed in the first coordinate chart $v = v^m \partial_m$ the components transform as the coordinates

$$v'^m = k^m{}_n v^n, \tag{41}$$

whereas the basis transforms

$$\partial'_m = k^n{}_m \partial_n. \tag{42}$$

As a consequence the vector v is independent of the choice of coordinate chart:

$$v = v^m \partial_m = (k^{-1})^n{}_m v'^m k^j{}_n \partial'_j = v'^m k^j{}_n (k^{-1})^n{}_m \partial'_j = v'^m \delta^j{}_m \partial'_j = v'^m \partial'_m = v', \tag{43}$$

where $\delta^n{}_m = 1$ if $n = m$ else 0. The change of coordinates is hence a gauge symmetry as presented in Sect. 2.1 and because of this we have the freedom of choosing a local basis at each point in $\mathcal{M}$ without affecting the tangent vectors, although the components might change, so we need to track both the components and the gauge the vector is expressed in.

To express this in the formalism introduced in Sect. 2.3, let again $\mathcal{M}$ be a d-dimensional smooth manifold and consider the frame bundle $P = \mathcal{L}\mathcal{M}$ over $\mathcal{M}$. The frame bundle is defined by its fiber $\mathcal{L}_x\mathcal{M}$ over a point $x \in \mathcal{M}$ as[2]

$$\mathcal{L}_x\mathcal{M} = \{(e_1, \ldots, e_d) : (e_1, \ldots, e_d) \text{ is a basis for } T_x\mathcal{M}\} \simeq \mathrm{GL}(d, \mathbb{R}). \tag{44}$$

---

[2] Technically $\mathcal{L}_x\mathcal{M}$ and $\mathrm{GL}(d, \mathbb{R})$ are isomorphic as sets.

This is a principal K-bundle $\mathcal{LM} \xrightarrow{\pi} \mathcal{M}$ over $\mathcal{M}$ with $K = \mathrm{GL}(d, \mathbb{R})$; the projection is defined by mapping a given frame to the point in $\mathcal{M}$ where it is attached. In the fundamental representation, the group acts from the right on the frame bundle as

$$\lhd : \mathcal{LM} \times \mathrm{GL}(d, \mathbb{R}) \to \mathcal{LM}, \qquad (e_1, \ldots, e_d) \lhd k = (k^m{}_1 e_m, \ldots, k^m{}_d e_m). \tag{45}$$

Hence the action performs a change of basis in the tangent space as described above. If we also choose a (real) representation (ρ, V) of the general linear group, we obtain an associated bundle $\mathcal{LM} \times_\rho V$ with typical fiber V as constructed in Sect. 2.3. Its elements $[e, v]$ are equivalence classes of frames $e = (e_1, \ldots, e_d) \in \mathcal{LM}$ and vectors $v \in V$, subject to the equivalence relation introduced in (32):

$$[e, v] = [e \lhd k, k^{-1} \rhd v] = [e \lhd k, \rho(k^{-1})v]. \tag{46}$$

The bundle associated to $\mathcal{LM}$ with fiber $\mathbb{R}^d$ over a smooth d-dimensional manifold $\mathcal{M}$ is closely related to the tangent bundle $T\mathcal{M}$ by the following classical result, see (Nakahara 2018; Kolář et al. 1993) for more details:

Theorem 2.23 *Let ρ be the standard representation of $\mathrm{GL}(d, \mathbb{R})$ on $\mathbb{R}^d$. Then the associated bundle $\mathcal{LM} \times_\rho \mathbb{R}^d$ and the tangent bundle $T\mathcal{M}$ are isomorphic as bundles.*

Remark 2.24 One can use the same approach to create an associated bundle which is isomorphic to the (p, q)-tensor bundle $T^p_q \mathcal{M}$, for any p, q, where the tangent bundle is the special case $(p, q) = (1, 0)$.

Remark 2.25 It is not necessary to use the full general linear group as structure group. One can instead construct a frame bundle and associated bundles using a subgroup, such as $\mathrm{SO}(d)$. This represents the restriction to orthogonal frames on a Riemannian manifold.

2.5 Explicit gauge equivariant convolution

Having developed the framework for gauge equivariant neural network layers, we will now describe how this can be used to construct concrete gauge equivariant convolutions. These were first developed in Cheng et al. (2019) and we lift this construction to the principal bundle P.

To begin, let P be a principal K-bundle, E_ρ and E_η be two associated bundles as constructed in Sect. 2.3. Since the general layer $\Phi : \Gamma(E_\rho) \to \Gamma(E_\eta)$ induces a unique map $\phi : C(P; \rho) \to C(P; \eta)$ we can focus on presenting a construction of the map ϕ. When constructing such a map with $C(P; \eta)$ as its co-domain it needs to respect the equivariance of the feature maps (33). Hence we get the following condition on ϕ:

$$k \rhd (\phi f)(p) = (\phi f)(p \lhd k^{-1}). \tag{47}$$

For gauge equivariance, i.e. a gauge transformation $\sigma : U \subseteq \mathcal{M} \to K$ commuting with ϕ, we need ϕ to satisfy (37):

$$\sigma \rhd \phi f = \phi(\sigma \rhd f), \tag{48}$$

where $(\sigma \rhd f)(p) = \sigma(\pi_P(p)) \rhd f(p)$.

Equation (47) is a condition on how the feature map ϕf needs to transform when moving in the fiber. To move across the manifold we need the concept of geodesics and for that a connection on the tangent bundle $T\mathcal{M}$ is necessary. The intuition for a connection is that it designates how a tangent vector changes when moved along the manifold. The most common

choice, if we have a (pseudo) Riemannian metric $g_{\mathcal{M}} : T\mathcal{M} \times T\mathcal{M} \to \mathbb{R}$ on $\mathcal{M}$, is to induce a connection from the metric, called the *Levi-Civita connection*. This construction is the one we will assume here. For details see (Nakahara 2018).

Remark 2.26 The metric acts on two tangent vectors from the same tangent space. Hence writing $g_{\mathcal{M}}(X, X)$ for $X \in T_x\mathcal{M}$ is unique. If we want to reference the metric at some specific point we will denote this in the subscript.

To make clear how we move on the manifold we introduce the *exponential map* from differential geometry, which lets us move around by following geodesics. The exponential map,

$$\exp : \mathcal{M} \times T\mathcal{M} \to \mathcal{M}, \tag{49}$$

maps (x, X), where $x \in \mathcal{M}$ is a point and $X \in T_x\mathcal{M}$ is a tangent vector, to the point $\gamma(1)$ where γ is a geodesic such that $\gamma(0) = x$ and $\frac{d}{dt}\gamma(0) = X$. The exponential map is well-defined since every tangent vector $X \in T_x\mathcal{M}$ corresponds to a unique geodesic.

Remark 2.27 It is common to write the exponential map $\exp(x, X)$ as $\exp_x X$, which is then interpreted as a mapping $\exp_x : T_x\mathcal{M} \to \mathcal{M}$ for each fixed $x \in \mathcal{M}$.

Note that the exponential map defines a diffeomorphism between the subset $B_R = \{X \in T_x\mathcal{M} : \sqrt{g_{\mathcal{M}}(X, X)} < R\} \subset T_x\mathcal{M}$ and the open set $\{y \in \mathcal{M} : d_{g_{\mathcal{M}}}(x, y) < R\}$ where $d_{g_{\mathcal{M}}}$ is the geodesic induced distance function on $\mathcal{M}$. This will later be used as our integration domain.

In order to lift the convolution presented in Cheng et al. (2019) we need a notion of parallel transport in the principal bundle P. This will be necessary for the layer ϕ to actually output feature maps. In order to do this, we introduce vertical and horizontal directions with respect to the base manifold $\mathcal{M}$ in the tangent spaces T_pP for $p \in P$.

Definition 2.28 A *connection* on a principal bundle P is a smooth decomposition $T_pP = V_pP \oplus H_pP$, into a vertical and a horizontal subspace, which is equivariant with respect to the right action $\triangleleft$ of K on P.

In particular, the connection allows us to uniquely decompose each vector $X \in T_pP$ as $X = X^V + X^H$ where $X^V \in V_pP$ and $X^H \in H_pP$, and provides a notion of transporting a point $p \in P$ parallel to a curve in $\mathcal{M}$.

Definition 2.29 Let $\gamma : [0, 1] \to \mathcal{M}$ be a curve and let P be a principal bundle equipped with a connection. The *horizontal lift* of γ through $p \in \pi_P^{-1}(\gamma(0))$ is the unique curve $\gamma_p^{\uparrow} : [0, 1] \to P$ such that $\forall t \in [0, 1]$

i) $\pi_P \left(\gamma_p^{\uparrow}(t) \right) = \gamma(t)$

ii) $\frac{d}{dt}\gamma_p^{\uparrow}(t) \in H_{\gamma_p^{\uparrow}(t)}P$

Remark 2.30 The property $\frac{d}{dt}\gamma_p^{\uparrow}(t) \in H_{\gamma_p^{\uparrow}(t)}P$ implies that the tangent to the lift $\gamma_p^{\uparrow}$ has no vertical component at any point along the curve.

Given a curve γ on $\mathcal{M}$ connecting $\gamma(0) = x$ and $\gamma(1) = y$ we can now define the *parallel transport map* $\mathcal{T}_{\gamma}$ on P as

$$\mathcal{T}_{\gamma} : P_x \to P_y, \quad p \mapsto \gamma_p^{\uparrow}(1). \tag{50}$$

Moreover, the map $\mathcal{T}_\gamma$ is equivariant with respect to K (Theorem 11.6 Kolář et al. (1993)); that is

$$\mathcal{T}_\gamma(p \triangleleft k) = \mathcal{T}_\gamma(p) \triangleleft k, \quad \forall k \in K. \tag{51}$$

Since we will only be working with geodesics and there is a bijection between geodesics through $x \in \mathcal{M}$ and $T_x\mathcal{M}$ we will instead denote the parallel transport map as $\mathcal{T}_X$ where $X = \frac{d}{dt}\gamma(0)$.

The final parts we need to discuss before defining the equivariant convolutions are the properties of the integration region and measure.

Lemma 2.31 *Let (u, U) be a coordinate patch around $x \in \mathcal{M}$. The set $B_R = \{X \in T_x\mathcal{M} : \sqrt{g_\mathcal{M}(X, X)} < R\} \subset T_x\mathcal{M}$ and the integration measure $\sqrt{\det(g_{\mathcal{M},x})}dX$ are invariant under change of coordinates that preserve orientation.*

Remark 2.32 The above lemma abuses notation for X since X in $g_\mathcal{M}(X, X)$ is indeed a tangent vector in $T_x\mathcal{M}$ while $dX = dy^1 \wedge dy^2 \wedge \cdots \wedge dy^d$ where y^i are the coordinates obtained from the coordinate chart. See the previous section for more details on coordinate charts.

Proof If $X = X^n e_n$ and $Y = Y^n e_n$ are two tangent vectors in $T_x\mathcal{M}$ then the metric evaluated on these is

$$g_\mathcal{M}(X, Y) = X^n Y^m g_\mathcal{M}(e_n, e_m). \tag{52}$$

Expressing X in other coordinates $X' = (k^{-1})^n{}_m X^m k^i{}_n e_i$ and similar for Y we get that (52) transforms as

$$g_\mathcal{M}(X', Y') = (k^{-1})^n{}_j X^j (k^{-1})^m{}_i Y^i g_\mathcal{M}(k^\ell{}_n e_\ell, k^o{}_m e_o). \tag{53}$$

Since $g_{\mathcal{M},x}$ is bilinear at each $x \in \mathcal{M}$

$$g_\mathcal{M}(X', Y') = \delta^l{}_j X^j \delta^m{}_i Y^i g_\mathcal{M}(e_l, e_m) = g_\mathcal{M}(X, Y), \tag{54}$$

and we are done. The invariance of $\sqrt{\det(g_{\mathcal{M},x})}dX$ comes from noting $dX' = \det(k^{-1})dX$ and

$$\det(g'_{\mathcal{M},x}) = \det(g_\mathcal{M}(e'_m, e'_n)) = \det(k^i{}_m k^j{}_n g_\mathcal{M}(e_i, e_j)) = \det(k)^2 \det(g_{\mathcal{M},x}). \tag{55}$$

Hence,

$$\sqrt{\det(g'_{\mathcal{M},x})}dX' = \sqrt{\det(k)^2 \det(g_{\mathcal{M},x})} \det(k^{-1})dX = \sqrt{\det(g_{\mathcal{M},x})}dX, \tag{56}$$

since we have restricted $GL(d, \mathbb{R})$ to those elements with positive determinant, i.e. preserve orientation. $\square$

Remark 2.33 The integration measure $\sqrt{\det(g_\mathcal{M})}dX$ is expressed in local coordinates but is an intrinsic property of the (pseudo) Riemannian manifold. It is often written as $\mathrm{vol}_{T_x\mathcal{M}}$ to be explicitly coordinate independent.

We can now state the convolution defined in Cheng et al. (2019) as follows. Choose a local coordinate chart around $x \in \mathcal{M}$ and let $s \in \Gamma(E_\rho)$ be a section of an associated bundle (a data point in our terminology in Sect. 2.3). The gauge equivariant convolution is

$$(\Phi s)(x) = \int_{B_R} \kappa(x, X) s|_{\exp_x X_G}(x)\sqrt{\det(g_\mathcal{M})}dX, \tag{57}$$

given a convolution kernel function

$$\kappa : \mathcal{M} \times T\mathcal{M} \to \mathrm{Hom}(E_\rho, E_\eta), \tag{58}$$

and X_G is the geometric tangent vector that corresponds to the coordinate representation X. Here, $\det(g_\mathcal{M})$ is the determinant of the Riemannian metric at x, and $s|_{\exp_x X}(x)$ represents the parallel transport, with respect to a connection on the associated bundle, of the element $s(\exp_x X)$ back to x along a geodesic. The convolution (57) is gauge equivariant if κ satisfies the constraint (61).

Note that $s(\exp_x X)$ is parallel transported back to x along a geodesic since this results in a consistent construction. To see this, bear in mind that choosing different paths for different tangent vectors X could change the resulting value due to $s(\exp_x X)$ transforming differently along different paths when transported back to x on a curved manifold. Fixing the parallel transport to geodesics resolves this ambiguity in a diffeomorphism invariant way.

Using the principle bundle construction from above, we now present the lifted version of (57).

Definition 2.34 (Gauge equivariant convolution) Let $U \subseteq \mathcal{M}$ be such that $x = \pi_P(p) \in B_R \subseteq U$ and choose a gauge $\omega : U \to P_U = \pi_P^{-1}(U)$. Let $f \in C(P; \rho)$ be a feature map, then the gauge equivariant convolution is defined as

$$(\phi f)(p) = [\kappa \star f](p) = \int_{B_R} \kappa(x, X) f(\mathcal{T}_X p) \mathrm{vol}_{T_x\mathcal{M}}, \tag{59}$$

where

$$\kappa : \mathcal{M} \times T\mathcal{M} \to \mathrm{Hom}(V_\rho, V_\eta), \tag{60}$$

is the convolution kernel and $\mathrm{vol}_{T_x\mathcal{M}}$ is the volume form for $T_x\mathcal{M}$.

As we will see in Theorem 2.36, the convolution (59) is gauge equivariant if κ satisfies the constraint (61).

Remark 2.35 In an ordinary CNN a convolutional kernel has compact support on the image plane, $\kappa : \mathbb{Z}^2 \to \mathrm{Hom}(V_{\mathrm{in}}, V_{\mathrm{out}})$ and hence depends on its position but in this case the kernel depends on its position on $\mathcal{M}$ and a direction.

For ϕ to be gauge equivariant, the kernel κ must have the following properties.

Theorem 2.36 *Let (u, U) be a coordinate chart such that $U \subseteq \mathcal{M}$ is a neighborhood to $x = \pi_P(p)$ and let $\phi : C(P; \rho) \to C(P; \eta)$ be defined as in (59). Then ϕ satisfies the feature map condition (47), along with the gauge equivariance condition (48) for all rigid gauge transformations $\sigma : U \to K$, if*

$$\kappa(x, X') = \eta(k^{-1})\kappa(x, X)\rho(k), \tag{61}$$

where $X' = k \triangleright X$ is the transformation of tangent vectors under the gauge group.

Proof Choose a local coordinate chart (u, U) and let $U \subseteq \mathcal{M}$ be a neighborhood to $x = \pi_P(p)$. Let further $\phi : C(P; \rho) \to C(P; \eta)$ be defined in (59). We can then write (59) in the local chart as

$$(\phi f)(p) = \int_{B_R} \kappa(x, X') f(\mathcal{T}_{X'_G} p) \sqrt{\det(g_\mathcal{M})} \mathrm{d}X'. \tag{62}$$

Let $k \in K$ be such that $X' = k \triangleright X$ and let $\sigma : U \to K$, $\sigma(y) = k$ for all $y \in U$ be a rigid gauge transformation. Note that $X'_G = X_G$ since the geometric vector is independent of choice of coordinate representation. Then the left hand side of (48) can be written as

$$(\sigma \triangleright \phi f)(p) = \sigma(x) \triangleright (\phi f)(p) = k \triangleright (\phi f)(p)$$

$$= \eta(k) \int_{B_R} \kappa(x, X') f(\mathcal{T}_{X'_G} p) \sqrt{\det(g_{\mathcal{M}})} \mathrm{d}X'. \tag{63}$$

Using (61) and a change of variables we get

$$\int_{B_R} \kappa(x, X) \rho(k) f(\mathcal{T}_{X_G} p) \sqrt{\det(g_{\mathcal{M}})} \mathrm{d}X. \tag{64}$$

From here we first prove the feature map condition and follow up with the proof of the gauge equivariance.

Since f is a feature map we get $\rho(k) f(\mathcal{T}_{X_G} p) = f\left((\mathcal{T}_{X_G} p) \triangleleft k^{-1}\right)$ and using the equivariance of T we arrive at

$$f\left((\mathcal{T}_{X_G} p) \triangleleft k^{-1}\right) = f\left(\mathcal{T}_{X_G} (p \triangleleft k^{-1})\right). \tag{65}$$

Thus,

$$k \triangleright (\phi f)(p) = (\phi f)(p \triangleleft k^{-1}), \tag{66}$$

which gives the feature map condition.

For the gauge equivariance condition note that

$$\rho(k) f(\mathcal{T}_{X_G} p) = \sigma(x) \triangleright f(\mathcal{T}_{X_G} p) = \sigma\left(\pi_P(\mathcal{T}_{X_G} p)\right) \triangleright f(\mathcal{T}_{X_G} p) = (\sigma \triangleright f)(\mathcal{T}_{X_G} p), \tag{67}$$

using that σ is a rigid gauge transformation. Hence we arrive at

$$(\sigma \triangleright \phi f)(p) = \phi(\sigma \triangleright f)(p), \tag{68}$$

proving the gauge equivariance of (59) for kernels which satisfy (61).

Note that we use the equivariance of the parallel transport in P to get the feature map condition and the rigidity of σ to arrive at the gauge equivariance. □

Remark 2.37 Along the same lines, one can also prove that the convolution (57) is gauge equivariant if the kernel satisfies (61). The main difference is that in the general case the point $p \in P$, at which the feature map is evaluated, holds information about the gauge used. For (57) the choice of gauge is more implicit. For more intuition on this, see the discussion below.

To get an intuition of the difference between the gauge equivariant convolution presented in Cheng et al. (2019) and the lifted convolution in (59) we first note that both require an initial choice: for (57) we need to choose a coordinate chart u around the point $x \in U \subseteq \mathcal{M}$ and for the lifted version we need to choose a local gauge $\omega : U \to P$ around $x = \pi_P(p)$. When dealing with gauge transformations that are changes of local basis choosing a gauge and a local coordinate system is the same.

Continuing, we note that applying a gauge transformation to a feature map on the principal bundle is the same as moving the evaluation point of the feature map along a fiber in the principal bundle:

$$(\sigma \triangleright f)(p) = \sigma\left(\pi_P(p)\right) \triangleright f(p) = f\left(p \triangleleft \sigma\left(\pi_P(p)\right)\right). \tag{69}$$

Since the action of the gauge group is free and transitive on the fibers of P we can interpret this as the feature using its evaluation point as information about what gauge it is in. Hence this states that the gauge group action on f amounts to evaluating f in a different gauge over the same point in $\mathcal{M}$. Using this interpretation we get that to parallel transport a feature map $f(\mathcal{T}_X p)$ designates which gauge should be used at every point and hence how f transforms when moved along γ. Choosing a connection on an associated bundle does the same thing: it prescribes how a vector changes when moved between fibers. In this sense the transport of a feature map on P is the same as parallel transport a vector in an associated bundle. Since the integration measure used in (57) is just a local coordinate representation of the volume form $\mathrm{vol}_{T_x \mathcal{M}}$ we have now related all components of (57) to our lifted version.

2.6 Examples: gauge equivariant networks on graphs and meshes

In this section we will illustrate how gauge-equivariant CNNs can be implemented in practice. This will be done in the context of graph networks where gauge equivariance corresponds to changes of coordinate system in the tangent space at each point in the graph. The main references for this section are (Worrall et al. 2017; Haan et al. 2020; Wiersma et al. 2020). We begin by introducing some relevant background on harmonic networks.

2.6.1 Harmonic networks on surfaces

Let $\mathcal{M}$ be a smooth manifold of real dimension 2. For any point $x \in \mathcal{M}$ we have the tangent space $T_x \mathcal{M} \cong \mathbb{R}^2$. The main idea is to consider a convolution in $\mathbb{R}^2$ at the point x and lift it to $\mathcal{M}$ using the Riemann exponential mapping. As we identify the tangent space $T_x \mathcal{M}$ with $\mathbb{R}^2$ we have a rotational ambiguity depending on the choice of coordinate system. This is morally the same as a "local Lorentz frame" in general relativity. This implies that the information content of a feature map in the neighborhood of a point x can be arbitrarily rotated with respect to the local coordinate system of the tangent space at x. We would like to have a network that is independent of this choice of coordinate system. Moreover, we also have a path-dependence when transporting filters across the manifold $\mathcal{M}$. To this end we want to construct a network which is equivariant in the sense that the convolution at each point is equivariant with respect to an arbitrary choice of coordinate system in $\mathbb{R}^2$. Neural networks with these properties were constructed in Worrall et al. (2017); Wiersma et al. (2020) and called *harmonic networks*.

We begin by introducing standard polar coordinates in $\mathbb{R}^2$:

$$(r, \theta) \in \mathbb{R}_+ \times [0, 2\pi). \tag{70}$$

In order to ensure equivariance we will assume that the kernels in our network are given by the (complex valued) *circular harmonics*

$$\kappa_m(r, \theta; \beta) = R(r)e^{i(m\theta + \beta)}. \tag{71}$$

Here the function $R : \mathbb{R}_+ \to \mathbb{R}$ is the radial profile of the kernel and β is a free parameter. The degree of rotation is encoded in $m \in \mathbb{Z}$. The circular harmonic transforms by an overall phase with respect to rotations in SO(2):

$$\kappa_m(r, \theta - \phi; \beta) = e^{im\phi}\kappa_m(r, \theta; \beta). \tag{72}$$

Let $f : \mathcal{M} \to \mathbb{C}$ be a feature map. The convolution of f with κ_m at a point $x \in \mathcal{M}$ is defined by

$$(\kappa_m \star f)(x) = \iint_{D_x(\epsilon)} \kappa_m(r, \theta; \beta) f(r, \theta) \, r \, \mathrm{d}r \, \mathrm{d}\theta. \tag{73}$$

The dependence on the point x on the right hand side is implicit in the choice of integration domain

$$D_x(\epsilon) = \left\{ (r, \theta) \in T_x \mathcal{M} \, \middle| \, r \in [0, \epsilon], \theta \in [0, 2\pi) \right\}, \tag{74}$$

which is a disc inside the tangent space $T_x \mathcal{M}$.

The group SO(2) acts on feature maps by rotations $\varphi \in [0, 2\pi)$ according to the regular representation ρ:

$$(\rho(\varphi) f)(r, \theta) = f(r, \theta - \varphi). \tag{75}$$

Under such a rotation the convolution by κ_m is equivariant

$$(\kappa_m \star \rho(\varphi) f)(x) = e^{im\varphi} (\kappa_m \star f)(x), \tag{76}$$

as desired. Note that when $m = 0$ the convolution is *invariant* under rotations.

Now assume that we may approximate $\mathcal{M}$ by a triangular mesh. This is done in order to facilitate the computer implementation of the network. The feature map is represented by a vector f_i at the vertex i in the triangular mesh. This plays the role of choosing a point $x \in \mathcal{M}$. Suppose we have a deep network and that we are studying the feature vector at layer ℓ in the network. When needed we then decorate the feature vector with the layer ℓ as well as the degree m of rotation: $f_{i,m}^{\ell}$.

A feature vector is parallel transported from a vertex j to vertex i along a geodesic connecting them. Any such parallel transport can be implemented by a rotation in the tangent space:

$$P_{j \to i}(f_{j,m}^{\ell}) = e^{im\varphi_{ji}} f_{j,m}^{\ell}, \tag{77}$$

where φ_{ji} is the angle of rotation. We are now equipped to define the convolutional layers of the network. For any vertex i we denote by $\mathcal{N}_i$ the set of vertices that contribute to the convolution. In the continuous setting this corresponds to the *support* of the feature map f on $\mathcal{M}$. The convolution mapping features at layer ℓ to features at layer $\ell + 1$ at vertex i is now given by

$$f_{i,m+m'}^{\ell+1} = \sum_{j \in \mathcal{N}_i} w_j \, \kappa_m(r_{ij}, \theta_{ij}; \beta) \, P_{j \to i}(f_{j,m'}^{\ell}). \tag{78}$$

The coefficient w_j represents the approximation of the integral measure and is given by

$$w_j = \frac{1}{3} \sum_{jkl} A_{jkl}, \tag{79}$$

where A_{jkl} is the area of the triangle with vertices jkl. The radial function R and the phase $e^{i\beta}$ are learned parameters of the network. The coordinates (r_{ij}, θ_{ij}) represents the polar coordinates of the tangent space of every vertex j in $\mathcal{N}_i$.

Let us now verify that this is indeed the building blocks of a deep equivariant network. To this end we should prove that the following diagram commutes:

$$
\begin{array}{ccc}
f^{\ell}_{i,m} & \longmapsto & f'^{\ell}_{i,m} \\
\Big\downarrow & & \Big\downarrow \\
f^{\ell+1}_{i,m+m'} & \longmapsto & f'^{\ell+1}_{i,m+m'}
\end{array}
\tag{2.61}
$$

Here, $f' = \rho(-\varphi)f$. The commutativity of this diagram ensures that a coordinate change in the tangent space commutes with the convolution. Let us now check this.

If we rotate the coordinate system at vertex i by an angle $-\varphi$ the feature vector transforms according to

$$
f^{\ell}_{i,m} \to f'^{\ell}_{i,m} = e^{im\varphi} f^{\ell}_{i,m}.
\tag{62}
$$

This coordinate transformation will affect the coordinates (r_{ij}, θ_{ij}) according to

$$
(r'_{ij}, \theta'_{ij}) = (r_{ij}, \theta_{ij} + \varphi).
\tag{63}
$$

The parallel transport $P_{j \to i}$ of features from j to i further transforms as

$$
P'_{j \to i} = e^{im\varphi} P_{j \to i}.
\tag{64}
$$

Using the above observations, let us now write the convolution with respect to the rotated coordinate system

$$
f'^{\ell+1}_{i,m+m'} = \sum_{j \in \mathcal{N}_i} w_j \, \kappa_m(r'_{ij}, \theta'_{ij}; \beta) P'_{j \to i}(f'^{\ell}_{j,m'}) = e^{i(m+m')} f^{\ell+1}_{i,m+m'}.
\tag{65}
$$

Thus we conclude that the diagram commutes. Note that nonlinearities can also be implemented in an equivariant fashion by only acting on the radial profile $R(r_{ij})$ but leaving the angular phase $e^{i\theta}$ untouched.

The formula (78) can be viewed as a discretization of the general gauge equivariant convolution on an arbitrary n-dimensional manifold $\mathcal{M}$ given in equation (57). The disc $D_x(\epsilon)$ plays the role of the ball B_R in the general formula. The combination $\sqrt{\det(g_{\mathcal{M},x})}\mathrm{d}X$ is approximated by the weight coefficients w_j in (78), while the coordinates X in the ball B_R corresponds to (r_{ij}, θ_{ij}) here. The parallel transport of the input feature f is implemented by the exponential map $\exp_x X$, whose discretization is $P_{j \to i}$. Thus we conclude that the harmonic network discussed above may be viewed as a discretized two-dimensional version of gauge equivariant convolutional neural networks, a fact that appears to have gone previously unnoticed in the literature.

2.6.2 Gauge-equivariant mesh CNNs

A different approach to gauge equivariant networks on meshes was given in Haan et al. (2020). A mesh can be viewed as a discretization of a two-dimensional manifold. A mesh consists of a set of vertices, edges, and faces. One can represent the mesh by a graph, albeit at the expense of loosing information concerning the angle and ordering between incident edges. Gauge equivariant CNNs on meshes can then be modeled by graph CNNs with gauge equivariant kernels.

Let M be a mesh, considered as a discretization of a two-dimensional manifold $\mathcal{M}$. We can describe this by considering a set of vertices $\mathcal{V}$ in $\mathbb{R}^3$, together with a set of tuples $\mathcal{F}$

consisting of vertices at the corners of each face. The mesh M induces a graph $\mathcal{G}$ by ignoring the information about the coordinates of the vertices.

We first consider graph convolutional networks. At a vertex $x \in \mathcal{V}$ the convolution between a feature f and a kernel K is given by

$$(\kappa \star f)(x) = \kappa_{\text{self}} f(x) + \sum_{y \in \mathcal{N}_x} \kappa_{\text{nb}} f(x). \tag{66}$$

The sum runs over the set $\mathcal{N}_x$ of neighboring vertices to x. The maps $\kappa_{\text{self}}, \kappa_{\text{nb}} \in \mathbb{R}^{N_{\text{in}} \times N_{\text{out}}}$ are modeling the self-interactions and nearest neighbor interactions, respectively. Notice that κ_{nb} is independent of $y \in \mathcal{N}_x$ and so does not distinguish between neighboring vertices. Such kernels are said to be *isotropic* (see also Remark 6.1).

One can generalize this straightforwardly by allowing the kernel κ_{nb} to depend on the neighboring vertices. In order to obtain a general gauge equivariant network on M we must further allow for features to be parallel transported between neighboring vertices. To this end we introduce the matrix $\rho(k_{y \to x}) \in \mathbb{R}^{N_{in} \times N_{out}}$ which transports the feature vector at y to the vertex x. The notation here is chosen to indicate that this is a representation ρ of the group element $k_{y \to x}$ that rotates between neighboring vertices in M.

Putting the pieces together we arrive at the gauge equivariant convolution on the mesh M:

$$(\kappa \star f)(x) = \kappa_{\text{self}} f(x) + \sum_{y \in \mathcal{N}_x} \kappa_{\text{nb}}(\theta_{xy}) \big(\rho(g_{y \to x}) f \big)(x), \tag{67}$$

where θ_{xy} is the polar coordinate of y in the tangent space $T_x M$. Under a general rotation k_φ by an angle φ the kernels must transform according to

$$\kappa(\theta - \varphi) = \rho_{\text{out}}(k_{-\varphi}) \kappa(\theta) \rho_{\text{in}}(k_\varphi), \tag{68}$$

in order for the network to be equivariant:

$$(\kappa \star \rho_{\text{in}}(k_{-\varphi}) f)(x) = \rho_{\text{out}}(k_{-\varphi})(\kappa \star f)(x). \tag{69}$$

Let us compare this with the harmonic networks discussed in the previous section. If we omit the self-interaction term, the structure of the convolution is very similar. The sum is taken over neighboring vertices $\mathcal{N}_x$, which is analogous to (78). The kernel κ_{nb} is a function of the polar coordinate θ_{xy} in the tangent space at x. The corresponding radial coordinate r_{xy} is suppressed here, but obviously we can generalize to kernels of the form $\kappa_{\text{nb}}(r_{xy}, \theta_{xy})$. Note, however, that here we have not made any assumption on the angular dependence of $\kappa_{\text{nb}}(r_{xy}, \theta_{xy})$, in contrast to the circular harmonics in (71). Note also that the condition (68) is indeed a special case of the general kernel condition for gauge equivariance as given in Theorem 2.36.

3 Group equivariant layers for homogeneous spaces

In the previous section, we considered neural networks which process data and feature maps defined on general manifolds $\mathcal{M}$, and studied the equivariance of those networks with respect to local gauge transformations. A prominent example was the local transformations of frames induced by a change of coordinates on $\mathcal{M}$. The freedom to choose frames is a local (gauge) symmetry which exists in any vector bundle over the manifold $\mathcal{M}$, but if $\mathcal{M}$ has an additional global (translation) symmetry, this can be exploited with great performance gains. In this

section, we discuss *group equivariant convolutional neural networks (GCNNs)* (Cohen et al. 2019; Cohen and Welling 2016), which are equivariant with respect to global transformations of feature maps induced from a global symmetry on $\mathcal{M}$. This section is largely based on Aronsson (2022), with a few additions: We relate convolutional layers in GCNNs to the gauge equivariant convolutions defined in Sect. 2.5, and we also discuss equivariance with respect to *intensity* and analyze its compatibility with group equivariance.

3.1 Homogeneous spaces and bundles

3.1.1 Homogeneous spaces

A manifold $\mathcal{M}$ with a sufficiently high degree of symmetry gives rise to symmetry transformations which translate any point $x \in \mathcal{M}$ to any other point $y \in \mathcal{M}$. For instance on a sphere $\mathcal{M} = S^2$, any point x can be rotated into any other point y; similarly, any point y in Euclidean space $\mathcal{M} = \mathbb{R}^n$ can be reached by translation from any other point x. Intuitively, this means that all points on the manifold are equivalent. This idea is formalized in the notion of a homogeneous space.

Definition 3.1 Let G be a topological group. A topological space $\mathcal{M}$ is called a *homogeneous G-space*, or just a *homogeneous space*, if there is a continuous, transitive group action

$$G \times \mathcal{M} \to \mathcal{M}, \qquad (g, x) \mapsto g \cdot x. \tag{70}$$

In the special case of Lie groups G and smooth manifolds $\mathcal{M}$, all homogeneous G-spaces are diffeomorphic to a quotient space G/K, with $K \leq G$ a closed subgroup (Theorem 21.18 Lee 2012). We therefore restrict attention to such quotient spaces. The elements of a homogeneous space $\mathcal{M} = G/K$ are denoted sometimes as x and sometimes as gK, depending on the context.

Remark 3.2 For technical reasons, the Lie group G is assumed to be unimodular. This is not a strong assumption, since examples of such groups include all finite or (countable) discrete groups, all abelian or compact Lie groups, the Euclidean groups, and many others (Folland 2016; Führ 2005). We also assume the subgroup $K \leq G$ to be compact—a common assumption that includes most homogeneous spaces of practical interest.

Example 3.3 (1) Any group G is a homogeneous space over itself with respect to group multiplication. In this case, K is the trivial subgroup so that $\mathcal{M} = G/K = G$.
(2) In particular, the pixel lattice $\mathcal{M} = \mathbb{Z}^2$ used in ordinary CNNs is a homogeneous space with respect to translations: $G = \mathbb{Z}^2$.
(3) The n-sphere is a homogeneous space $S^n = \mathrm{SO}(n+1)/\mathrm{SO}(n)$ for all $n \geq 1$. The special case $n = 2$, $S^2 = \mathrm{SO}(3)/\mathrm{SO}(2)$, has been extensively studied in the context of geometric deep learning, cf. Sect. 6.
(4) Euclidean space $\mathbb{R}^n = \mathrm{E}(n)/\mathrm{O}(n)$ is homogeneous under rigid motions; combinations of translations and rotations which form the Euclidean group $G = \mathrm{E}(n) = \mathbb{R}^n \rtimes \mathrm{O}(n)$.

Homogeneous G-spaces are naturally equipped with a bundle structure (§7.5 Steenrod (1999)) since G is decomposed into orbits of the subgroup K, which form fibers over the base manifold G/K. The bundle projection is given by the quotient map

$$q : G \to G/K, \qquad g \mapsto gK, \tag{71}$$

which maps a group element to its equivalence class. The free right action $G \times K \to G$ defined by right multiplication, $(g, k) \mapsto gk$, preserves each fiber $q^{-1}(gK)$ and turns G into a principal bundle with structure group K. We can therefore view the homogeneous setting as a special case of the framework discussed in Sect. 2 with $P = G$ and $\mathcal{M} = G/K$.

3.1.2 Homogeneous vector bundles

Consider a rotation $g \in \mathrm{SO}(3)$ of the sphere, such that $x \in S^2$ is mapped to $gx \in S^2$. Intuitively, it seems reasonable that when the sphere rotates, its tangent spaces rotate with it and the corresponding transformation $T_x S^2 \to T_{gx} S^2$ should be linear, because all tangent vectors are rotated in the same way. Indeed, for any homogeneous space $\mathcal{M} = G/K$, the differential of the left-translation map L_g on $\mathcal{M}$ is a linear isomorphism $\mathrm{d}L_g : T_x \mathcal{M} \to T_{gx}\mathcal{M}$ for all $x \in \mathcal{M}$ and each $g \in G$. This means that the transitive G-action on the homogeneous space $\mathcal{M}$ induces a transitive G-action on the tangent bundle $T\mathcal{M}$ that is linear on each fiber. This idea is captured in the notion of a homogeneous vector bundle.

Definition 3.4 (§5.2.1 Wallach (1973)) Let $\mathcal{M}$ be a homogeneous G-space and let $E \xrightarrow{\pi} \mathcal{M}$ be a smooth vector bundle with fibers E_x. We say that E is *homogeneous* if there is a smooth left action $G \times E \to E$ satisfying

$$g \cdot E_x = E_{gx}, \tag{72}$$

and such that the induced map $L_{g,x} : E_x \to E_{gx}$ is linear, for all $g \in G$, $x \in \mathcal{M}$.

Associated vector bundles $E_\rho = G \times_\rho V_\rho$ are homogeneous vector bundles with respect to the action $G \times E_\rho \to E_\rho$ defined by $g \cdot [g', v] = [gg', v]$. The induced linear maps

$$L_{g,x} : E_x \to E_{gx}, \qquad [g', v] \mapsto [gg', v], \tag{73}$$

leave the vector inside the fiber invariant and are thus linear. Any homogeneous vector bundle E is isomorphic to an associated vector bundle $G \times_\rho E_K$ where $K = eK \in G/K$ is the identity coset and where ρ is the representation defined by $\rho(k) = L_{k,K} : E_K \to E_K$ (§5.2.3 Wallach (1973)).

Remark 3.5 In this section, (ρ, V_ρ) and (η, V_η) are finite-dimensional unitary representations of the compact subgroup $K \leq G$. Unitarity is important for defining induced representations and, by extension, G-equivariant layers below. This is not a restriction of the theory. Indeed, since K is compact, unitarity of ρ (η) can be assumed without loss of generality by defining an appropriate inner product on V_ρ (V_η) (Lemma 7.1.1 Deitmar and Echterhoff (2014)).

Let $\langle \cdot, \cdot \rangle_\rho$ be an inner product that turns $V_\rho \simeq E_K$ into a Hilbert space and makes ρ unitary. This inner product then induces an inner product on each fiber E_x,

$$\langle [g, v], [g, w] \rangle_x = \langle v, w \rangle_\rho, \tag{74}$$

which is well-defined precisely because ρ is unitary. Further consider the unique G-invariant measure $\mathrm{d}x$ on G/K such that, for all integrable functions $f : G \to \mathbb{C}$ (Theorem 1.5.3 Deitmar and Echterhoff (2014)),

$$\int_G f(g) \, \mathrm{d}g = \int_{G/K} \int_K f(xk) \, \mathrm{d}k \mathrm{d}x. \tag{75}$$

We can then combine the measure $\mathrm{d}x$ with (74) to integrate the point-wise norm of a section.

Definition 3.6 Let (ρ, V_ρ) be a finite-dimensional unitary K-representation and consider the homogeneous vector bundle $E_\rho = G \times_\rho V_\rho$.

(1) The *induced representation* $\mathrm{ind}_K^G \rho$ of G is the representation

$$\left(\mathrm{ind}_K^G \rho(g)s\right)(x) = g \cdot s(g^{-1}x), \tag{76}$$

on the complex Hilbert space of square-integrable sections

$$L^2(E_\rho) = \left\{ s : G/K \to E_\rho \ \middle| \ \int_{G/K} \|s(x)\|_x^2 \, \mathrm{d}x < \infty \right\}. \tag{77}$$

(2) The *induced representation* $\mathrm{Ind}_K^G \rho$ of G is the representation

$$\left(\mathrm{Ind}_K^G \rho(g)f\right)(g') = f(g^{-1}g'), \tag{78}$$

on the complex Hilbert space of square-integrable feature maps

$$L^2(G; \rho) = \left\{ f : G \to V_\rho \ \middle| \ \int_G \|f(g)\|_\rho^2 \, \mathrm{d}g < \infty \right\}. \tag{79}$$

The linear isomorphism $\varphi_\rho : C(G; \rho) \to \Gamma(E_\rho)$, $f \mapsto s_f$ extends to a unitary isomorphism $L^2(G; \rho) \to L^2(E_\rho)$ that intertwines the induced representations $\mathrm{ind}_K^G \rho$ and $\mathrm{Ind}_K^G \rho$. That is, the induced representations are unitarily equivalent and we therefore choose to identify them.

3.2 Group equivariant layers

To summarize the previous subsection, global symmetry of the homogeneous space $\mathcal{M} = G/K$ gives rise to homogeneous vector bundles and an induced representation that lets us translate sections and feature maps. GCNNs are motivated by the idea that layers between feature maps should preserve the global translation symmetry of $\mathcal{M}$. They do so by intertwining induced representations.

Definition 3.7 A *G-equivariant layer* is a bounded linear map $\Phi : L^2(E_\rho) \to L^2(E_\eta)$ that intertwines the induced representations:

$$\Phi \circ \mathrm{ind}_K^G \rho = \mathrm{ind}_K^G \eta \circ \Phi. \tag{80}$$

That is, G-equivariant layers are elements $\Phi \in \mathrm{Hom}_G(L^2(E_\rho), L^2(E_\eta))$.

The unitary equivalence between $\mathrm{ind}_K^G \rho$ and $\mathrm{Ind}_K^G \rho$ implies that any G-equivariant layer $\Phi : L^2(E_\rho) \to L^2(E_\eta)$ induces a unique bounded linear operator $\phi : L^2(G; \rho) \to L^2(G; \eta)$ such that $\Phi s_f = s_{\phi f}$, as in Sect. 2. This operator also intertwines the induced representations,

$$\phi \circ \mathrm{ind}_K^G \rho = \mathrm{ind}_K^G \eta \circ \phi, \tag{81}$$

hence $\phi \in \mathrm{Hom}_G(L^2(G; \rho), L^2(G; \eta))$. The operators ϕ are also called G-equivariant layers.

As the name suggests, GCNNs generalize convolutional neural networks ($G = \mathbb{Z}^2$, $K = \{0\}$) to other homogeneous spaces G/K. The next definition generalize convolutional layers (3) in this direction.

Definition 3.8 A *convolutional layer* $\phi : L^2(G; \rho) \to L^2(G; \eta)$ is a bounded operator

$$(\phi f)(g) = [\kappa \star f](g) = \int_G \kappa(g^{-1}g') f(g') \, \mathrm{d}g', \qquad f \in L^2(G; \rho), \tag{82}$$

with an operator-valued kernel $\kappa : G \to \mathrm{Hom}(V_\rho, V_\eta)$.

Given bases for V_ρ and V_η, we can think of the kernel κ as a matrix-valued function. Each row in this matrix is a function $\kappa_i : G \to V_\rho$, just like the feature maps, so we can interpret each row as a separate filter that we convolve with respect to. This is analogous to ordinary CNNs in which both data and filters have the same structure as images. Furthermore, $\dim V_\rho$ is the number of input channels and $\dim V_\eta$ the number of output channels, one for each filter κ_i. From this perspective, the full matrix κ is a stack of $\dim V_\eta$ filters and the convolutional layer ϕ computes all output-channels simultaneously.

Convolutional layers form the backbone of GCNNs, and implementations are often based on these layers. Note that the kernel in a convolutional layer cannot be chosen arbitrarily but must satisfy certain transformation properties, to make sure that $\kappa \star f$ transforms correctly. First of all, the requirement that $\kappa \star f \in L^2(G; \eta)$ implies that

$$\int_G \eta(k)\kappa(g) f(g) \, \mathrm{d}g = \eta(k) [\kappa \star f] (e) = [\kappa \star f] (k^{-1}) = \int_G \kappa(kg) f(g) \, \mathrm{d}g, \tag{83}$$

which is satisfied if $\kappa(kg) = \eta(k)\kappa(g)$. Moreover, unimodularity of G means that the left Haar measure on G is also right-invariant, so we can perform a change of variables $g \mapsto gk$,

$$\int_G \kappa(g) f(g) \, \mathrm{d}g = \int_G \kappa(gk) f(gk) \, \mathrm{d}g = \int_G \kappa(gk)\rho(k)^{-1} f(g) \, \mathrm{d}g, \tag{84}$$

which indicates that $\kappa(gk) = \kappa(g)\rho(k)$. These relations can be summarized in one equation,

$$\kappa(kgk') = \eta(k)\kappa(g)\rho(k'), \qquad g \in G, k, k' \in K. \tag{85}$$

This was previously discussed in Cohen et al. (2019). A consequence of this relation is that $\kappa(gk) f(gk) = \kappa(g) f(g)$ for all $k \in K$ and each $g \in G$, so the product κf only depends on the base point $x = q(g) \in G/K$. The quotient integral formula (75) then implies that

$$[\kappa \star f] (g) = \int_{G/K} \kappa(g^{-1}x) f(x) \, \mathrm{d}x, \tag{86}$$

see (Corollary 3.24 Aronsson (2022)) for a formal proof.

The fact that convolutional layers can be computed by integrating over the homogeneous space G/K, rather than integrating over the group G, can greatly improve computational efficiency when G is large.

Remark 3.9 The integral (86) is closely related to the gauge equivariant convolution (59). First of all, homogeneous spaces $\mathcal{M} = G/K$ always admit Riemannian metrics $g_\mathcal{M}$ that are invariant under translations (§2.3 Howard (1994)), see also (Santaló and Kac 2004). The Riemannian volume form $\mathrm{vol}_\mathcal{M}$ is also invariant, and the corresponding Riemannian volume measure is thus an invariant measure on $\mathcal{M}$. By the quotient integral formula (Theorem 1.5.3 Deitmar and Echterhoff (2014)), the Riemannian volume measure is related to $\mathrm{d}y$ by a positive scaling factor $c > 0$, so that

$$\int_\mathcal{M} \kappa(g^{-1}y) f(y) \, \mathrm{d}y = \int_\mathcal{M} L_g \kappa f \, \mathrm{vol}_\mathcal{M}. \tag{87}$$

The sliding kernel $\kappa(g^{-1}y)$ in (86) can be viewed as a special case of the explicitly base point-dependent kernel $\kappa(x, X)$ in (59), after taking into account the diffeomorphism between B_R and $\{y \in \mathcal{M} : d_{g_{\mathcal{M}}}(x, y) < R\}$. We can also interpret the domain of integration as the kernel support. The parallel transport map $\mathcal{T}_X$ need not be invoked here as the integrand in (86) is already defined on $x \in G/K$, without lifting to G. Finally, the relation $\mathrm{vol}_{\mathcal{M}}|_x = \mathrm{vol}_{T_x \mathcal{M}}$ lets us rewrite (86), with some abuse of notation and ignoring the constant c, as

$$[\kappa \star f](g) = \int_{B_R} L_g \kappa f \ \mathrm{vol}_{T_x \mathcal{M}}, \tag{88}$$

which is similar to (59).

Boundedness of (82) is guaranteed if the kernel matrix elements $\kappa_{ij} : G \to \mathbb{C}$ are integrable functions, for some choices of bases in V_ρ, V_η Aronsson (2022).

Theorem 3.10 *(Aronsson (2022)) Let $\phi : L^2(G; \rho) \to L^2(G; \eta)$ be a bounded linear map.*

(1) If ϕ is a convolutional layer, then ϕ is a G-equivariant layer.
(2) If ϕ is a G-equivariant layer such that $\Im(\phi)$ is a space of bandlimited functions, then ϕ is a convolutional layer.

The bandlimit criteria is automatically satisfied for all finite groups and for discrete abelian groups such as $G = \mathbb{Z}^2$ (Corollaries 20–21 Aronsson (2022)). It follows that all G-equivariant layers can be expressed as convolutional layers for these groups.

3.3 Equivariance with respect to intensity

In image processing tasks, a neural network may treat an image differently depending on the level of saturation. One way to avoid this is to design saturation-equivariant neural networks, or *intensity*-equivariant neural networks if we generalize beyond image data. In this section, we define this notion of equivariance and investigate the question of when a G-equivariant layer is also intensity-equivariant. This part is based on the concept of induced systems of imprimitivity in (§3.2 Kaniuth and Taylor (2012)).

Mathematically, one changes the intensity of a data point $s \in L^2(E_\rho)$ by scaling the vector at each point: $(\psi s)(x) = \psi(x)s(x)$ where $\psi : G/K \to \mathbb{C}$ is a continuous function. Equivalently, we can scale the feature maps instead, via the mapping

$$S(\psi) : L^2(G; \rho) \to L^2(G; \rho), \qquad \big(S(\psi)f\big)(g) = \psi(gK)f(g). \tag{89}$$

For technical reasons we assume that ψ vanishes at infinity: $\psi \in C_0(G/K)$.

Definition 3.11 A bounded linear map $\phi : L^2(G; \rho) \to L^2(G; \eta)$ is *equivariant with respect to intensity*, or *intensity-equivariant*, if

$$S(\psi) \circ \phi = \phi \circ S(\psi), \tag{90}$$

for all $\psi \in C_0(G/K)$.

Remark 3.12 The mapping (89) is a $*$-representation of $C_0(G/K)$ on the space $L^2(G; \rho)$, and intensity-equivariant maps (90) are intertwiners of two such representations.

Example 3.13 Let $T : V_\rho \to V_\eta$ be a bounded linear map that intertwines ρ and η, that is, $T \in \mathrm{Hom}_K(V_\rho, V_\eta)$. By letting T act point-wise on feature maps $f \in L^2(G; \rho)$, we obtain a bounded linear map

$$\phi_T : L^2(G; \rho) \to L^2(G; \eta), \qquad (\phi_T f)(g) = T\big(f(g)\big), \tag{91}$$

and we observe that ϕ_T is equivariant with respect to intensity. This is because ϕ_T performs point-wise transformations of vectors and S performs point-wise multiplication by scalar, so we need only employ the linearity of T:

$$\big(S(\psi)\phi_T f\big)(g) = \psi(gK)T\big(f(g)\big) = T\big(\psi(gK)f(g)\big) = \big(\phi_T S(\psi)f\big)(g). \tag{92}$$

Note that (91) is also G-equivariant since its action on $f(g) \in V_\eta$ does not depend on $g \in G$. Indeed,

$$\big(\phi_T \circ \mathrm{Ind}_K^G \rho(g)f\big)(g') = T\big(f(g^{-1}g')\big) = (\phi_T f)(g^{-1}g') = \big(\mathrm{Ind}_K^G \eta(g) \circ \phi_T f\big)(g). \tag{93}$$

It turns out that (91) are the only possible transformations $\phi : L^2(G; \rho) \to L^2(G; \eta)$ that are both G-equivariant and intensity-equivariant.

Theorem 3.14 (Theorem 3.16 Kaniuth and Taylor (2012)) *Let $\phi : L^2(G; \rho) \to L^2(G; \eta)$ be a G-equivariant layer. Then ϕ is intensity-equivariant if and only if $\phi = \phi_T$ for some $T \in \mathrm{Hom}_K(V_\rho, V_\eta)$.*

For some groups, this theorem exclude convolutional layers from being intensity-equivariant, as the following example illustrates. This means that intensity-equivariant and convolutional layers are two separate classes of layers for these groups.

Example 3.15 Consider the special case $G = \mathbb{R}$ and let $K = \{0\}$ be the trivial subgroup. Then ρ, η are trivial representations, and assume for simplicity that $\dim V_\rho = \dim V_\eta = 1$ so that $L^2(\mathbb{R}; \rho) = L^2(\mathbb{R}; \eta) = L^2(\mathbb{R})$. If we now let $\phi : L^2(\mathbb{R}) \to L^2(\mathbb{R})$ be a convolutional layer with some kernel $\kappa : \mathbb{R} \to \mathbb{C}$,

$$(\phi f)(x) = [\kappa \star f](x) = \int_{-\infty}^{\infty} \kappa(y - x) f(y) \, \mathrm{d}y, \qquad f \in L^2(\mathbb{R}), \tag{94}$$

and consider an arbitrary element $\psi \in C_0(\mathbb{R})$, then

$$\phi(S(\psi)f)(x) = \int_{-\infty}^{\infty} \kappa(y - x)\psi(y)f(y) \, \mathrm{d}y. \tag{95}$$

Note that the function ψ is part of the integrand, which is not the case for $S(\psi)\phi f$. This is essentially what prevents convolutional layers (94) from being intensity-equivariant.

To see this, fix $\epsilon > 0$ and consider the bump function $\psi_\epsilon : \mathbb{R} \to [0, 1]$ defined by

$$\psi_\epsilon(x) = \begin{cases} \exp\left(1 - \dfrac{1}{1 - (x/\epsilon)^2}\right), & x \in (-\epsilon, \epsilon) \\ 0, & \text{otherwise} \end{cases}, \tag{96}$$

which is supported on the compact interval $[-\epsilon, \epsilon]$ and satisfies $\psi_\epsilon(0) = 1$. Then $\psi_\epsilon \in C_0(\mathbb{R})$ and it is clear that

$$\big(S(\psi_\epsilon)\phi f\big)(0) = \psi_\epsilon(0)\phi f(0) = \phi f(0). \tag{97}$$

Comparing (95) and (97), we see that the convolutional layer ϕ is intensity-equivariant only if

$$\int_{-\infty}^{\infty} \kappa(y) f(y) \, \mathrm{d}y = \phi f(0) = \int_{-\infty}^{\infty} \kappa(y) \psi_\epsilon(y) f(y) \, \mathrm{d}y, \tag{98}$$

for each $f \in L^2(\mathbb{R})$. Hölder's inequality yields the bound

$$|\phi f(0)| \leq \int_{-\infty}^{\infty} |\kappa(y) \psi_\epsilon(y) f(y)| \, \mathrm{d}y = \|\kappa \psi_\epsilon f\|_1 \leq \|\kappa \psi_\epsilon\|_2 \|f\|_2. \tag{99}$$

However, because $\psi_\epsilon \leq 1$ everywhere and vanishes outside $[-\epsilon, \epsilon]$, we have

$$\|\kappa \psi_\epsilon\|_2^2 \leq \int_{-\epsilon}^{\epsilon} |\kappa(x)|^2 \, \mathrm{d}x, \tag{100}$$

which vanishes as $\epsilon \to 0$. It follows that $\phi f(0) = 0$ for all $f \in L^2(\mathbb{R})$. This argument can be adapted to show that $\phi f(x) = 0$ for all $x \in \mathbb{R}$, so we conclude that ϕ must vanish identically. In other words, there does not exist a non-zero, intensity-equivariant convolutional layer in the case $G = \mathbb{R}$, $K = \{0\}$. This is consistent with Theorem 3.14 because if a convolutional layer (94) had been intensity-equivariant, then Theorem 3.14 would imply that $\phi = \phi_T$ acts point-wise, i.e. that $\phi f(x) \in \mathbb{C}$ only depends on $f(x) \in \mathbb{C}$. This would require the kernel κ to behave like a Dirac delta function, which is not a mathematically well-defined function.

The conclusion would have been different, had (100) not vanished in the limit $\epsilon \to 0$. This would have required the singleton $\{0\}$ to have non-zero Haar measure, which is only possible if the Haar measure is the counting measure, that is, if G is discrete (Proposition 1.4.4 Deitmar and Echterhoff (2014)). In that case, one could define a convolution kernel $\kappa_T : G \to \text{Hom}(V_\rho, V_\eta)$ in terms of a Kronecker delta, $\kappa_T(g) = \delta_{g,e} T$ for some linear operator $T \in \text{Hom}_K(V_\rho, V_\eta)$. The convolutional layer

$$(\phi f)(g) = [\kappa_T \star f](g) = \int_G \kappa_T(g^{-1} g') f(g') \, \mathrm{d}g' = T\big(f(g)\big), \tag{101}$$

would then coincide with ϕ_T and intensity equivariance would be achieved.

4 General group equivariant convolutions

In the previous sections, we have developed a mathematical framework for the construction of equivariant convolutional neural networks on homogeneous spaces, starting from an abstract notion of principal bundles, leading to the convolutional integral (82). Here, we continue this discussion and consider various generalizations of (82) which could be implemented in a CNN. While doing so, we will reproduce various expressions for convolutions found in the literature. A similar overview of convolutions over feature maps and kernels defined on homogeneous spaces can be found in Kondor and Trivedi (2018), however our discussion also takes non-trivial input- and output representations into account (see also (Cohen et al. 2019, 2018a)).

4.1 Structure of symmetric feature maps

Let $\mathcal{X}$, $\mathcal{Y}$ be topological spaces and V_1, V_2 vector spaces. The input features are maps $f_1 : \mathcal{X} \to V_1$, the output features are maps $f_2 : \mathcal{Y} \to V_2$. Assume that a group G acts on all four

spaces by[3]

$$\sigma_1(g)x \,, \ \forall x \in \mathcal{X} \,, \qquad\qquad \rho_1(g)v_1 \,, \ \forall v_1 \in V_1 \,, \qquad (102)$$

$$\sigma_2(g)y \,, \ \forall y \in \mathcal{Y} \,, \qquad\qquad \rho_2(g)v_2 \,, \ \forall v_2 \in V_2 \,, \qquad (103)$$

for all $g \in G$. These actions can be combined to define a group action on the feature maps:

$$[\pi_1(g)f_1](x) = \rho_1(g)f_1(\sigma_1^{-1}(g)x) \,, \qquad (104)$$

$$[\pi_2(g)f_2](y) = \rho_2(g)f_2(\sigma_2^{-1}(g)y) \,. \qquad (105)$$

In the case of non-trivial group actions ρ_1 or ρ_2, the resulting networks are called *steerable*.

Example 4.1 (GCNNs) In the simplest case of non-steerable GCNNs discussed above, we have $\rho_1 = \rho_2 = \mathrm{id}$ and the input to the first layer is defined on the homogeneous space $\mathcal{X} = G/K$ of G with respect to some subgroup K. The output of the first layer then has $\mathcal{Y} = G$. Subsequent layers have $\mathcal{X} = \mathcal{Y} = G$. The group G acts on itself and on G/K by group multiplication: $\sigma_{1,2}(g')g = g'g$ for $g' \in G$ and $g \in G$ or $g \in G/K$.

On top of this, Kondor and Trivedi (2018) discusses also the cases of convolutions from a homogeneous space into itself, $\mathcal{X} = \mathcal{Y} = G/K$ and of a double coset space into a homogeneous space $\mathcal{X} = H\backslash G/K$, $\mathcal{Y} = G/K$. In all these cases, $\sigma_{1,2}$ are given by group multiplication.[4]

The representations σ_i, ρ_i arising in applications are dictated by the problem at hand. In the following example, we discuss some typical cases arising in computer vision.

Example 4.2 (Typical computer vision problems)

Consider the case that the input data of the network consists of flat images. Then, for the first layer of the network, we have $\mathcal{X} = \mathbb{R}^2$ and $V_1 = \mathbb{R}^N$ if the image has N color channels and the input features are functions $f_1 : \mathbb{R}^2 \to \mathbb{R}^N$. Typical symmetry groups G of the data are rotations (SO(2)), translations ($\mathbb{R}^2$) or both (SE(2)) or finite subgroups of these. For these groups, the representation σ_1 is the fundamental representation $\mathbf{2}_{\mathrm{SO}(2)}$ which acts by matrix multiplication, i.e. for $G = \mathrm{SO}(2)$, we have

$$\sigma_1(\phi) = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix} \,, \qquad (106)$$

and similarly for the other groups. Since the color channels of images do not contain directional information, the input representation ρ_1 is the trivial representation: $\rho_1 = \mathrm{id}_N$.

The structure of the output layer depends on the problem considered. For image classification, the output should be a probability distribution $P(\Omega)$ over classes Ω which is invariant under actions of G. In the language of this section, this would correspond to the domain $\mathcal{Y}$ of the last layer and the representations σ_2 and ρ_2 being trivial and $V_2 = P(\Omega)$.

The cases of semantic segmentation and object detection are discussed in detail in Sects. 5.2 and 5.3 below.

---

[3] Note that as opposed to previous sections, here, we are considering representations of the entire group G, not just of a subgroup.

[4] Note, however, that there appear to be some mathematical inconsistencies in Kondor and Trivedi (2018). For example, in Case I of Sect. 4.1, the symmetry group acts from the right on a left quotient space, which is ill-defined. Furthermore, the choice of coset representative denoted by $\bar{x}$ is not equivariant with respect to the group action, even from the left. Similar problems arise in Case III.

4.2 The kernel constraint

The integral mapping f_1 to f_2 is defined in terms of the kernel κ which is a bounded, linear-operator valued, continuous mapping

$$\kappa : \mathcal{X} \times \mathcal{Y} \rightarrow \text{Hom}(V_1, V_2). \tag{107}$$

That is, $\kappa(x, y) : V_1 \rightarrow V_2$ is a homomorphism for each pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$. In neural networks, κ additionally has local support, but we do not make this assumption here as the following discussion does not need it.

In order to have an integral which is compatible with the group actions, we require $\mathcal{X}$ to have a Borel measure which is invariant under σ_1, i.e.

$$\int_{\mathcal{X}} f_1(\sigma_1(g)x)\mathrm{d}x = \int_{\mathcal{X}} f_1(x)\mathrm{d}x, \tag{108}$$

for every integrable function $f_1 : \mathcal{X}_1 \rightarrow V_1$ and all $g \in G$. Now, we can write the output features as an integral over the kernel κ and the input features,

$$f_2(y) = [\kappa \cdot f_1](y) = \int_{\mathcal{X}} \kappa(x, y) f_1(x)\mathrm{d}x, \tag{109}$$

where the matrix multiplication in the integrand is left implicit.

We now require the map from input to output features to be equivariant with respect to the group actions (104) and (105), i.e. for any input function f_1, we require

$$[\kappa \cdot \pi_1(g) f_1] = \pi_2(g)[\kappa \cdot f_1] \qquad \forall g \in G. \tag{110}$$

This leads to the following

Lemma 4.3 *The transformation (109) is equivariant with respect to π_1, π_2 if the kernel satisfies the constraint*

$$\kappa(\sigma_1^{-1}(g)x, \sigma_2^{-1}(g)y) = \rho_2^{-1}(g)\kappa(x, y)\rho_1(g), \qquad \forall x \in \mathcal{X}, \ \forall y \in \mathcal{Y}, \ \forall g \in G. \tag{111}$$

Proof The constraint (111) is equivalent to

$$\kappa(\sigma_1(g)x, y)\rho_1(g) = \rho_2(g)\kappa(x, \sigma_2^{-1}(g)y). \tag{112}$$

Integrating against f_1 leads to

$$\int_{\mathcal{X}} \kappa(\sigma_1(g)x, y)\rho_1(g) f_1(x)\mathrm{d}x = \int_{\mathcal{X}} \rho_2(g)\kappa(x, \sigma_2^{-1}(g)y) f_1(x)\mathrm{d}x. \tag{113}$$

Using (108) on the left-hand side shows that this is equivalent to (110). $\qquad \square$

4.3 Transitive group actions

In the formulation above, the output feature map is computed as a scalar product of the input feature map with a kernel which satisfies the constraint (111). In this section, we discuss how the two can be combined into one expression, which is the familiar convolutional integral, if the group acts *transitively* by σ_1 on the space $\mathcal{X}$. I.e. we assume that there exists a base point $x_0 \in \mathcal{X}$ such that for any $x \in \mathcal{X}$, there is a $g_x \in G$ with

$$x = \sigma_1(g_x)x_0. \tag{114}$$

Defining

$$\kappa(y) = \kappa(x_0, y), \tag{115}$$

we obtain from (111)

$$\kappa(x, y) = \rho_2(g_x)\kappa\left(\sigma_2^{-1}(g_x)y\right)\rho_1^{-1}(g_x). \tag{116}$$

Plugging this into (109) and using (114) yields a convolution as summarized in the following proposition.

Proposition 4.4 *If G acts transitively on $\mathcal{X}$, the map $\kappa \cdot f_1$ defined in (109) subject to the constraint (111) can be realized as the convolution*

$$[\kappa \star f_1](y) = \int_{\mathcal{X}} \rho_2(g_x)\,\kappa\left(\sigma_2^{-1}(g_x)y\right)\rho_1^{-1}(g_x)\,f_1(\sigma_1(g_x)x_0)\,\mathrm{d}x. \tag{117}$$

Since in (117), the integrand only depends on x through g_x, we can replace the integral over $\mathcal{X}$ by an integral over G if the group element g_x is unique for all $x \in \mathcal{X}$ (we can then identify $\mathcal{X}$ with G by $x \mapsto g_x$). In this is the case, the group action of G on $\mathcal{X}$ is called *regular* (i.e. it is transitive and *free*), leading to

Proposition 4.5 *If G acts regularly on $\mathcal{X}$, the map $\kappa \cdot f_1$ defined in (109) subject to the constraint (111) can be realized as the convolution*

$$[\kappa \star f_1](y) = \int_G \rho_2(g)\,\kappa\left(\sigma_2^{-1}(g)y\right)\rho_1^{-1}(g)\,f_1(\sigma_1(g)x_0)\,\mathrm{d}g, \tag{118}$$

where we use the Haar measure to integrate on G. Furthermore, for a regular group action, the group element g_x in (116) is unique and hence the kernel $\kappa(y)$ in (118) is unconstrained.

Remark 4.6 If there is a subgroup K of G which stabilizes x_0, i.e. $\sigma_1(h)x_0 = x_0 \;\forall h \in K$, $\mathcal{X}$ can be identified with the homogeneous space G/K. Proposition 4.5 corresponds to K being trivial. As will be spelled out in detail in Sect. 4.4, the integral in (118) effectively averages the kernel κ over K before it is combined with f_1, leading to significantly less expressive effective kernels. In the case of spherical convolutions, this was pointed out in Makadia et al. (2007); Cohen et al. (2018). Nevertheless, constructions of this form are used in the literature, cf. e.g. Esteves et al. (2018).

To illustrate (118), we start by considering GCNNs as discussed in Example 4.1.

Example 4.7 (GCNNs with non-scalar features) Consider a GCNN as discussed in Example 4.1 above. On G, a natural reference point is the unit element e, so we set $x_0 = e$ and hence $g_x = x$. Since σ_1 is now a regular group action, we can use (118) which simplifies to

$$[\kappa \star f](y) = \int_G \rho_2(g)\kappa(\sigma_2^{-1}(g)y)\rho_1^{-1}(g)f(g)\mathrm{d}g, \tag{119}$$

where $\kappa(y)$ is unconstrained. This is the convolution used for GCNNs if the input- and output features are not scalars.

Another important special case for (118) is given by spherical convolutions which are widely studied in the literature, cf. Sects. 1.6 and 6.

Example 4.8 (Spherical convolutions) Consider an equivariant network layer with $\mathcal{X} = \mathcal{Y} = G = \mathrm{SO}(3)$ as used in spherical CNNs Cohen et al. (2018). G acts on itself transitively by left-multiplication, $\sigma_i(Q)R = QR$ for $Q, R \in \mathrm{SO}(3)$ with base point $x_0 = e$. Therefore, according to Proposition (118), the transformation (109) can be written as[5]

$$[\kappa \star f_1](S) = \int_{\mathrm{SO}(3)} \rho_2(R) \kappa(R^{-1} S) \rho_1^{-1}(R) f_1(R) \, \mathrm{d}R \,, \tag{120}$$

where the Haar measure on $\mathrm{SO}(3)$ is given in terms of the Euler angles α, β, γ by

$$\int_{\mathrm{SO}(3)} \mathrm{d}R = \int_0^{2\pi} \mathrm{d}\alpha \int_0^\pi \mathrm{d}\beta \sin\beta \int_0^{2\pi} \mathrm{d}\gamma \,. \tag{121}$$

Instead of assuming a transitive group action on $\mathcal{X}$ and using this to solve the kernel constraint, one can also assume a transitive group action on $\mathcal{Y}$ and use this to solve the kernel constraint. Specifically, if there is a $y_0 \in \mathcal{Y}$ such that for any $y \in \mathcal{Y}$, we have a $g_y \in G$ satisfying

$$y = g_y y_0 \,, \tag{122}$$

we can define

$$\kappa(x) = \kappa(x, y_0) \,. \tag{123}$$

Then, according to (111), the two-argument kernel is given by

$$\kappa(x, y) = \rho_2(g_y) \kappa(\sigma_1^{-1}(g_y) x) \rho_1^{-1}(g_y) \,, \tag{124}$$

yielding the following

Proposition 4.9 *If G acts transitively on $\mathcal{Y}$, the map $\kappa \cdot f_1$ defined in (109) subject to the constraint (111) can be realized as the convolution*

$$[\kappa \star f_1](y) = \int_{\mathcal{X}} \rho_2(g_y) \kappa(\sigma_1^{-1}(g_y) x) \rho_1^{-1}(g_y) f_1(x) \mathrm{d}x \,. \tag{125}$$

However, since the group element now depends on y instead of on the integration variable x, we cannot replace the integral over $\mathcal{X}$ by an integral over G as we did above in (118) and have to compute the group element g_y for each y.[6]

4.4 Semi-direct product groups

In the previous section, we considered different convolutional integrals for the case that G acts transitively on $\mathcal{X}$. In particular, we recovered the familiar integral over G in the case that G acts regularly on $\mathcal{X}$. In practice however, the action of G on $\mathcal{X}$ is often transitive, but not regular. To study this case, consider a group G which is a semi-direct product group $G = N \rtimes K$ of a normal subgroup $N \subset G$ and a subgroup $K \subset G$. We require that N acts

---

[5] In Cohen et al. (2018), the spherical convolution is defined as $[\kappa \star f_1](S) = \int_{\mathrm{SO}(3)} \kappa(S^{-1}R) f_1(R) \, \mathrm{d}R$, which arises from our expression by $\rho_{1,2} \to \mathrm{id}$, $\kappa(R) \to \kappa(R^{-1})$. In the language developed below, the reference fixes a point in $\mathcal{Y}$ instead of $\mathcal{X}$. We use a different convention here to make the relation to the general case more transparent. For applications, the distinction is irrelevant.

[6] In the spherical case of fixing a point in $\mathcal{Y}$ to solve the kernel constraint, mentioned in footnote 5, this problem does not arise: We have $\mathcal{X} = \mathcal{Y} = G$ and therefore integrate over G from the beginning. Furthermore, we can choose $y_0 = e$, so $y = g_y$.

regularly on $\mathcal{X}$ and one can choose a base point $x_0 \in \mathcal{X}$ which is stabilized by K. In this case, we again define

$$\kappa(y) = \kappa(x_0, y), \tag{126}$$

obtaining

$$\kappa(x, y) = \rho_2(n_x)\kappa(\sigma_2^{-1}(n_x)y)\rho_1^{-1}(n_x), \tag{127}$$

where $n_x \in N$ is the unique group element which satisfies $x = \sigma_1(n_x)x_0$. The convolutional integral then becomes

$$[\kappa \star f_1](y) = \int_{\mathcal{X}} \rho_2(n_x)\kappa(\sigma_2^{-1}(n_x)y)\rho_1^{-1}(n_x)f_1(\sigma_1(n_x)x_0)\,\mathrm{d}x. \tag{128}$$

Since N acts regularly on $\mathcal{X}$, the integral over $\mathcal{X}$ can be replaced by an integral over N. However, since (127) only fixes an element of N but not of K, the kernel is not unconstrained and we have the following

Proposition 4.10 *If $G = N \rtimes K$ is a semi-direct product group, with N and K as above, the map $\kappa \cdot f_1$ defined in (109) subject to the constraint (111) can be realized as the convolution*

$$[\kappa \star f_1](y) = \int_N \rho_2(n)\kappa(\sigma_2^{-1}(n)y)\rho_1^{-1}(n)f_1(\sigma_1(n)x_0)\mathrm{d}n, \tag{129}$$

where the kernel satisfies the constraint

$$\kappa(\sigma_2(h)y) = \rho_2(h)\kappa(y)\rho_1^{-1}(h). \tag{130}$$

In practice, the constraint (130) restricts the expressivity of the network, as mentioned in Remark 4.6. To construct a network layer using (129) and (130), one identifies a basis of the space of solutions of (130), expands κ in this basis in (129) and trains only the coefficients of the expansion. A basis of solutions of (130) for compact groups K in terms of representation-theoretic quantities was given in Lang and Weiler (2020).

Example 4.11 (SE(n) equivariant CNNs) In the literature, the special case $G = \mathrm{SE}(n) = \mathbb{R}^n \rtimes \mathrm{SO}(n)$ and $\mathcal{X} = \mathcal{Y} = \mathbb{R}^n$ has received considerable attention due to its relevance to applications. Our treatment follows Weiler et al. (2018), for a brief overview of the relevant literature, see Sect. 1.6.

In this case, $\mathbb{R}^n$ acts by vector addition on itself, for $t \in \mathbb{R}^n$, $\sigma_1(t)x = \sigma_2(t)x = x + t$ and SO(n) acts by matrix multiplication, for $R \in \mathrm{SO}(n)$, $\sigma_1(R)x = \sigma_2(R)x = Rx$. Moreover, $\mathbb{R}^n$ acts trivially on V_1 and V_2, i.e. $\rho_1(t) = \rho_2(t) = \mathrm{id}$. The base point x_0 is in this case the origin of $\mathbb{R}^n$, which is left invariant by rotations. With this setup, (129) simplifies to

$$[\kappa \star f_1](y) = \int_{\mathbb{R}^n} \kappa(y - t)f_1(t)\mathrm{d}t. \tag{131}$$

The kernel constraint (130) becomes

$$\kappa(Ry) = \rho_2(R)\kappa(y)\rho_1^{-1}(R) \quad \forall R \in \mathrm{SO}(n). \tag{132}$$

4.5 Non-transitive group actions

If the group action of G on $\mathcal{X}$ is not transitive, we cannot simply replace the integral over $\mathcal{X}$ by an integral over G as in (118). However, we can split the space $\mathcal{X}$ into equivalence classes under the group action by defining

$$\mathcal{X}/G = \{[x_0] : x_0 \in \mathcal{X}\} \quad \text{where} \quad x_0 \in \mathcal{X} \sim \tilde{x}_0 \in \mathcal{X} \iff \exists\, g \in G \text{ s.t. } \sigma_1(g)x_0 = \tilde{x}_0 \,. \tag{133}$$

Within each equivalence class $[x_0]$, G acts transitively by definition. For each class we select an arbitrary representative as base point and define a one-argument kernel by

$$\kappa_{x_0}(y) = \kappa(x_0, y) \,. \tag{134}$$

Using this kernel, we can write the integral (109) as

$$[\kappa \star f_1](y) = \int_{\mathcal{X}/G} \int_G \rho_2(g)\kappa_{x_0}(\sigma_2^{-1}(g)y)\rho_1^{-1}(g) f_1(\sigma_1(g)x_0) \,\mathrm{d}g\mathrm{d}x_0 \,. \tag{135}$$

Example 4.12 (SO(3) acting on $\mathbb{R}^3$) Consider $\mathcal{X} = \mathbb{R}^3$ and $G = \mathrm{SO}(3)$. In this case, G does not act transitively on $\mathcal{X}$, since SO(3) conserves the norm on $\mathbb{R}^3$ and integrating over G is not enough to cover all of $\mathcal{X}$. Hence, $\mathcal{X}/G$ can be identified with the space $\mathbb{R}^+$ of norms in $\mathbb{R}^3$. The split of $\mathcal{X}$ into $\mathcal{X}/G$ and G therefore corresponds to the usual split in spherical coordinates, in which the integral over the radius is separate from the integral over the solid angle. In Fox et al. (2021), a similar split is used, where the integral over G is realized as a graph convolution with isotropic kernels.

5 Equivariant deep network architectures for machine learning

After having defined various general convolution operators in the previous section, we now want to illustrate how to assemble these into an equivariant neural network architecture using discretized versions of the integral operators that appeared above. To this end, we will first discuss the crucial equivariant nonlinearities that enable the network to learn non-linear functions. Then, we will use two important tasks from computer vision, namely semantic segmentation on S^2 and object detection on $\mathbb{Z}^2$, to show in detail what the entire equivariant network architecture looks like.

5.1 Nonlinearities and equivariance

So far, we have only discussed the linear transformation in the equivariant network layers. However, in order to approximate nonlinear functions, it is crucial to include nonlinearities into the network architecture. Of course, for the entire network to be equivariant with respect to group transformations, the nonlinearities must also be equivariant. Various equivariant nonlinearities have been discussed in the literature which we will review here briefly. An overview and experimental comparison of different nonlinearities for E(2) equivariant networks was given in Weiler and Cesa (2019).

A nonlinear activation function of a neural network is a nonlinear function η which maps feature maps to feature maps. An equivariant nonlinearity additionally satisfies the constraint

$$\eta[\pi_1(g)f] = \pi_2(g)\eta(f) \,, \tag{136}$$

where we used the notation introduced in (104) and (105).

Remark 5.1 (Biases) Often, the feature map to which the nonlinearity is applied is not directly the output of a convolutional layer, but instead a learnable bias is added first, i.e. we compute $\eta(f + b)$, where $b : \mathcal{X} \to V_1$ is constant.

The most important class of nonlinearities used in group equivariant networks act point-wise on the input, i.e.

$$(\eta(f))(x) = \bar{\eta}(f(x)), \tag{137}$$

for some nonlinear function $\bar{\eta} : V_1 \to V_2$. If f transforms as a scalar (i.e. $\rho_{1,2} = \text{id}$ in (104) and (105)), then any function $\bar{\eta}$ can be used to construct an equivariant nonlinearity according to (137). In this case, a rectified linear unit is the most popular choice for $\bar{\eta}$ and was used e.g. in Worrall et al. (2017); Weiler et al. (2018); Cohen et al. (2018); Esteves et al. (2020), but other popular activation functions appear as well Thomas et al. (2018).

If however the input- and output feature maps transform in non-trivial representations of G, $\bar{\eta}$ needs to satisfy

$$\bar{\eta} \circ \rho_1(g) = \rho_2(g) \circ \bar{\eta}, \qquad \forall g \in G. \tag{138}$$

Remark 5.2 If the convolution is computed in Fourier space with limited bandwidth (as discussed for the spherical case in Sect. 6.2), point-wise nonlinearities in position space as (137) break strict equivariance since they violate the bandlimit. The resulting networks are then only approximately group equivariant.

In the special case that the domain of f is a finite group G and $\rho_{1,2}$ are trivial, the point-wise nonlinearity (137) is called *regular* Cohen and Welling (2016). Regular nonlinearities haven been used widely in the literature, e.g. Cohen et al. (2019, 2018); Weiler et al. (2018); Dieleman et al. (2016); Hoogeboom et al. (2018); Bekkers et al. (2018). If the domain of the feature map is a quotient space G/K, Weiler and Cesa (2019) calls (137) a *quotient nonlinearity*. Similarly, given a non-linear function which satisfies (138) for representations $\rho_{1,2}$ of the subgroup K, a nonlinearity which is equivariant with respect to the induced representation Ind_K^G can be constructed by point-wise action on G/K Weiler and Cesa (2019). If f is defined on a semi-direct product group $N \rtimes G$, all these constructions can be extended by acting point-wise on N.

A nonlinearity which is equivariant with respect to a subgroup K of a semi-direct product group $G = K \ltimes N$ is given by the *vector field nonlinearity* defined in Weiler and Cesa (2019) after a construction in Marcos et al. (2017). In the reference, it is constructed for the cyclic rotation group $K = C_N < \text{SO}(2)$ and the group of translations $N = \mathbb{R}^2$ in two dimensions, but we generalize it here to arbitrary semi-direct product groups. The vector field nonlinearity maps a function on G to a function on N and is equivariant with respect to representations π_{reg} and π_2 defined by

$$\left[\pi_{\text{reg}}(\tilde{k}) f_1 \right](kn) = f_1(\tilde{k}^{-1}kn), \tag{139}$$

$$\left[\pi_2(\tilde{k}) f_2 \right](n) = \rho_2(\tilde{k}) f_2(n), \tag{140}$$

for some representation ρ_2 of K. It reduces the domain of the feature map by taking the maximum over orbits of K, akin to a subgroup maxpooling operation. However, in order to retain some of the information contained in the K-dependence of the feature map, it multiplies the maximum with an argmax over orbits of K which is used to construct a feature vector

transforming in the representation ρ_2 of K. Its equivariance is shown in the proof of the following

Proposition 5.3 *(Vector field nonlinearity) The nonlinearity $\eta \; : \; L^2(K \ltimes N, V_1) \; \to \; L^2(N, V_2)$ defined by*

$$[\eta(f)](n) = \max_{k \in K}(f(kn))\rho_2(\underset{k \in K}{\operatorname{argmax}}(f(kn)))v_0 \,, \tag{141}$$

where max (argmax) of a vector-valued function is defined as the max (argmax) of the norm and $v_0 \in V_2$ is some reference point, satisfies the equivariance property

$$\eta(\pi_{\text{reg}}(k)f) = \pi_2(k)[\eta(f)] \,, \qquad \forall k \in K \,. \tag{142}$$

Proof To verify the equivariance, we act with the π_{reg} on f, yielding

$$\begin{aligned}
\left[\eta(\pi_{\text{reg}}(\tilde{k})f)\right](n) &= \max_{k \in K}(f(\tilde{k}^{-1}kn))\rho_2(\underset{k \in K}{\operatorname{argmax}}(f(\tilde{k}^{-1}kn)))v_0 \\
&= \max_{k \in K}(f(kn))\rho_2(\tilde{k}\,\underset{k \in K}{\operatorname{argmax}}(f(kn)))v_0 \\
&= \rho_2(\tilde{k})[\eta(f)](n) \,,
\end{aligned} \tag{143}$$

where we used that the maximum is invariant and the argmax equivariant with respect to shifts.[7] $\square$

Example 5.4 (Two-dimensional roto-translations) Consider the special case $G = C_N \ltimes \mathbb{R}^2$, $v_0 = (1, 0)$ and ρ_2 the fundamental representation of C_N in $\mathbb{R}^2$ which was discussed in Weiler and Cesa (2019), where the vector field nonlinearity was first developed. In this case, we will denote a feature map on $C_N \ltimes \mathbb{R}^2$ by a function $f_\theta(x)$ where $\theta \in C_N$ and $x \in \mathbb{R}^2$. Then, the vector field nonlinearity is given by

$$[\eta(f)](x) = \max_{\theta \in C_N}(f_\theta(x)) \begin{pmatrix} \cos(\operatorname{argmax}_{\theta \in C_N} f_\theta(x)) \\ \sin(\operatorname{argmax}_{\theta \in C_N} f_\theta(x)) \end{pmatrix} \,, \tag{144}$$

illustrating the origin of its name.

If the input- and output features transform in the same unitary transformation, i.e. $\rho_1 = \rho_2 = \rho$ and $\rho(g)\rho^\dagger(g) = \text{id}$ for all $g \in G$, then *norm nonlinearities* are a widely used special case of (137). These satisfy (138) and are defined as

$$\bar{\eta}(f(x)) = \alpha(||f(x)||) f(x) \,, \tag{145}$$

for any nonlinear function $\alpha : \mathbb{R} \to \mathbb{R}$. Examples for α used in the literature include sigmoid Weiler et al. (2018), relu (Worrall et al. 2017; Esteves et al. 2020), shifted soft plus (Thomas et al. 2018) and swish Müller et al. (2021). In Favoni et al. (2022), norm nonlinearities are used for matrix valued feature maps with $|| \cdot || = \Re(\text{tr}(\cdot))$ and $\alpha = \text{relu}$. A further variation are *gated nonlinearities* (Weiler et al. 2018; Finzi et al. 2021) which are of the form $\bar{\eta}(f(x)) = \sigma(s(x)) f(x)$ with σ the sigmoid function and $s(x)$ an additional scalar feature from the previous layer.

Instead of point-wise nonlinearities in position space, nonlinearities in Fourier space have also been used. These circumvent the problem mentioned in Remark 5.2 and the resulting

---

[7] Note that the equivariance of the argmax function relies on the uniqueness of the maximum. If the global maximum is not unique, there is an ambiguity which cannot be resolved.

networks are therefore equivariant within numerical precision. E.g. (Thomas et al. 2018) uses norm nonlinearities of the form (145) in Fourier space. However, the most important nonlinearity in this class are *tensor product nonlinearities* (Kondor 2018; Kondor et al. 2018) which compute the tensor product of two Fourier components of the input feature map. They yield a feature map transforming in the tensor product representation which is then decomposed into irreducible representations. To eschew the large tensors in this process, Cobb et al. (2020); McEwen et al. (2021) introduce various refinements of this basic idea.

Remark 5.5 (Universality) The Fourier-space analogue of a point-wise nonlinearity in position space is a nonlinearity which does not mix the different Fourier components, i.e. which is of the form

$$[\eta(f)]^\ell = \bar{\eta}(f^\ell) \,. \tag{146}$$

This is the way that norm- and gated nonlinearities have been used in Thomas et al. (2018); Weiler et al. (2018); Finzi et al. (2021). However, as pointed out in Finzi et al. (2021), these nonlinearities can dramatically reduce the approximation capacity of the resulting equivariant networks. This problem does not exist for tensor product nonlinearities.

Subgroup pooling (Cohen and Welling 2016; Weiler et al. 2018; Bekkers et al. 2018; Worrall et al. 2017; Winkels and Cohen 2018) with respect to a subgroup K of G can be seen as a nonlinearity which does not act point-wise on f, but on orbits of K in the domain of f,

$$\eta(f)(gK) = \bar{\eta}(f(\{gk|k \in K\})) \,. \tag{147}$$

This breaks the symmetry group of the network from G to G/K and yields a feature map defined on G/K. The function $\bar{\eta}$ is typically an average[8] or maximum of the arguments.

5.2 Semantic segmentation on S^2

After having reviewed equivariant nonlinearities, we can now proceed to discuss concrete equivariant network architectures. For computer vision tasks such as semantic segmentation and object detection, the standard flat image space of $\mathbb{Z}^2$ is a homogeneous space under translation and hence falls in the class of convolutions discussed in Sect. 3. The standard convolution (3) is, by (3.10), the natural $\mathbb{Z}^2$ equivariant layer in this context. Let's now take the first concrete steps into a more complex example of how the equivariant structure on homogeneous spaces can be applied in a more interesting setup.

Moving to the sphere S^2 provides a simple example of a non-trivial homogeneous space as the input manifold. As detailed in (2), a semantic segmentation model can now be viewed as a map

$$\mathcal{N} : L^2(S^2, \mathbb{R}^3) \to L^2(S^2, P(\Omega)), \tag{148}$$

where $P(\Omega)$ is the space of probability distributions over the N classes Ω. The output features transform as scalars under the group action on the input manifold. In the notation of (104) and (105), this means that $\mathcal{X} = \mathcal{Y} = S^2$, $V_1 = \mathbb{R}^3$ and $V_2 = \mathbb{R}^N$. The symmetry group should act on the output space in the same way as on the input space, i.e. $\sigma_2 = \sigma_1$. Since the output class labels do not carry any directional information, we have $\rho_2 = \mathrm{id}_N$.

Viewed as a quotient space $S^2 = SO(3)/SO(2)$, the sphere is a homogeneous space where each point can be associated with a particular rotation. One possible parametrization would

---

[8] Of course, the average is a linear operation.
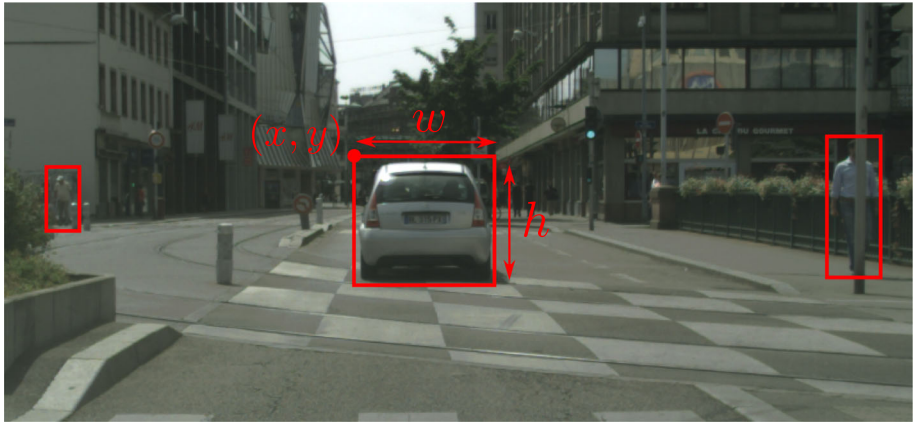
Fig. 5 Objects detected by bounding boxes. A bounding box is given by an anchor point (x, y) together with dimensions (w, h). Image from Cordts et al. (2016)

Fig. 6 Simple object detection model. In the first row the name of the first feature map (i.e. input image) is given by f_{in}, it maps the domain $\mathbb{Z}^2$ to RGB values. It can be represented by a rank 3 tensor $(f_{\text{in}})^c_{ij}$. The first convolution maps f_{in} to a new feature map f_2 with N_2 filters giving rise to another rank 3 tensor. The output is a feature map $\mathbb{Z}^2 \to \mathbb{R}^2 \oplus \mathbb{R}^3$, where an element in the co-domain takes the form (x, y, w, h, c). The first part in the co-domain $\mathbb{R}^2$, represents anchor point coordinates and transforms under translations. The second part $\mathbb{R}^3$ represents the dimensions of the bounding box together with a confidence score, both invariant under translations

be in terms of latitude (θ) and longitude (ϕ), i.e. the latitude specifies the angular distance from the north pole whereas the longitude specifies the angular distance from a meridian.

The corresponding convolution following from (82) can be formulated as an integral over S^2 using this parametrization. In practice there are much more efficient formulations using the spectral structure on the sphere, see Sect. 6 for a more detailed treatment of spherical convolutions.

5.3 Object detection on $\mathbb{Z}^2$

If we instead stay on $\mathbb{Z}^2$ as the input space but let the model output object detections we have a non-trivial example of where the output transforms under the group action and where equivariance of the full model becomes relevant.

Let us consider a single-stage object detection model that outputs a dense map $\mathbb{Z}^2 \to \mathbb{R}^2 \oplus \mathbb{R}^3$ of candidate detection in the form (x, y, w, h, c), where $(x, y) \in \mathbb{R}^2$ corresponds

to the anchor point of an axis aligned bounding box with dimensions (w, h). The confidence score c (or binary classifier) corresponds to the probability of the bounding box containing an object.[9] Fig. 6 illustrates how the anchor point together with the dimensions form a bounding box used to indicate a detection. The first subspace $\mathbb{R}^2$ in the co-domain, corresponding to anchor point coordinates, is identified as continuous coordinates on the input space $\mathbb{Z}^2$ and transforms under translations.

Starting from the same example architecture as the semantic segmentation model in Fig. 1, the object detection model in Fig. 5 ends with a feature map transforming in the fundamental representation of the translation group $\mathbb{Z}^2$. Note that since the detection problem is formulated as a regression task, which is usually implemented using models with floating point arithmetic, the output naturally takes values in $\mathbb{R}$ rather than $\mathbb{Z}$.

As introduced in (4), on $\mathbb{Z}^2$, the convolutions are discretized to

$$f_2(x, y) = \Phi_1(f_{\text{in}})(x, y) = [\kappa_1 \star f_{\text{in}}](x, y) = \sum_{(x', y') \in \mathbb{Z}^2} L_{(x,y)} \kappa_1(x', y') f_{\text{in}}(x', y'), \quad (149)$$

where we have specified the first layer as an illustration (cf. Fig. 5) and κ_1 is the kernel. Concretely on $[0, W] \times [0, H] \in \mathbb{Z}^2$ with the feature map and kernel represented as rank 3 tensors[10] this takes the form

$$(f_2)^c_{xy} = \sum_{(x', y') \in \mathbb{Z}^2} \sum_{c=0}^{2} \kappa^c_{(x'-x)(y'-y)} (f_{\text{in}})^c_{x'y'}. \quad (150)$$

The convolution in (149) is equivariant with respect to translations under $\mathbb{Z}^2$ as shown in (6). For the nonlinearity, we can choose any of the ones discussed in Sect. 5.1, e.g. a point-wise relu,

$$(f'_2)^c_{ij} = \text{relu}\left((f_2)^c_{ij}\right). \quad (151)$$

Thus the model is a G-equivariant network that respects the $\mathbb{Z}^2$ structure of the image plane. Note that in contrast to the case of semantic segmentation in Sect. 5.2 the output features here transforms under the group action. If the image is translated the corresponding anchor points for the detections should also be translated. This equivariance is built into the model rather than being learned from data, as would be the case in a standard object detection model.

In the notation of (104) and (105) the input and output spaces are $\mathcal{X} = \mathcal{Y} = \mathbb{Z}^2$ with output feature maps taking values in $V_2 = \mathbb{R}^2 \oplus \mathbb{R}^3$. The symmetry group $G = \mathbb{Z}^2$ acts by $\sigma_{(x',y')}(x, y) = (x + x', y + y')$ on the input and output space and by $\rho_{(x',y')}(x, y, w, h, c) = (x + x', y + y', w, h, c)$ on V_2.

The output of this model in terms of anchor points and corresponding bounding box dimensions is one of many possible ways to formulate the object detection task. For equivariance under translations this representation makes it clear that it is the position of the bounding box that transforms with the translation group.

If we instead are interested in a model that is equivariant with respect to rotations of the image plane it is instructive to regard a model that predicts bounding boxes that are not axis aligned. Let the output of the new model, as in Sect. 1, take values in $V_2 = \mathbb{R}^2 \oplus \mathbb{R}^2 \oplus \mathbb{R}^2$

---

[9] The dense map of candidate detections is typically filtered based on the confidence score using a method such as non-maximum suppression.

[10] In the sense of multidimensional arrays.

where an element (a, v_1, v_2) corresponds to a bounding box with one corner at a and spanned by the parallelogram of v_1 and v_2. All three output vector spaces transform in the fundamental representation of SO(2) so that $\rho_2 = \mathbf{2}_{SO(2)} \oplus \mathbf{2}_{SO(2)} \oplus \mathbf{2}_{SO(2)}$, cf. (106).

6 Spherical convolutions

In this section, we will investigate the spherical convolutions introduced in Example 4.8 in more detail. Mathematically, this case is particularly interesting, because on the sphere and on SO(3), we can leverage the power of the rich spectral theory on the sphere in terms of spherical harmonics and Wigner matrices to find explicit and compact expressions for the convolutions. Also for practical considerations, this case is of particular importance since data given on a sphere arises naturally in many applications, e.g. for fisheye cameras Coors et al. (2018), cosmological data Perraudin et al. (2019), weather data, molecular modeling Boomsma and Frellsen (2017) or diffusion MRI Elaldi et al. (2021). To faithfully represent this data, equivariant convolutions are essential.

There is a sizable literature on equivariant spherical convolutions. The approach presented here follows Cohen et al. (2018) and extends the results in the reference at some points. An overview of the existing literature in the field can be found in Sect. 1.6.

6.1 Preliminaries

For spherical CNNs, the input data has the form $f : S^2 \to \mathbb{R}^N$, i.e. the first layer of the network has $\mathcal{X} = S^2$. The networks discussed here are special cases of the GCNNs constructed in Sect. 3 for which the symmetry group G is the rotation group in three dimensions, SO(3) and the subgroup K is either SO(2) or trivial. In this framework, we identify S^2 with the homogeneous space $G/K = SO(3)/SO(2)$. The first layer of the network has trivial K in the output, i.e. $\mathcal{Y} = SO(3)$ and subsequent layers have trivial K also in the input, i.e. $\mathcal{X} = \mathcal{Y} = SO(3)$. The latter case was already discussed in Example 4.8, leading to (120).

For the first layer, SO(3) acts by the matrix–vector product on the input space, $\sigma_1(R)x = Rx$ and as usual by group multiplication on the output, $\sigma_2(R)Q = RQ$. The construction in Cohen et al. (2018) uses in this case the identity element in $\mathcal{Y} = SO(3)$ to solve the kernel constraint, leading to

$$(\kappa \star f)(R) = \int_{S^2} \rho_2(R)\kappa(R^{-1}x)\rho_1^{-1}(R)f(x)\,\mathrm{d}x\,, \tag{152}$$

cf. (125). The integration measure on the sphere is given by

$$\int_{S^2} \mathrm{d}x(\theta, \varphi) = \int_0^{2\pi} \mathrm{d}\varphi \int_0^{\pi} \mathrm{d}\theta \sin\theta\,. \tag{153}$$

Note that in (152) we perform the integral here over the input domain $\mathcal{X}$ and not the symmetry group G since we cannot replace the integral over $\mathcal{X}$ by an integral over G if we use a reference point in $\mathcal{Y}$, as mentioned above.

Remark 6.1 (Convolutions with $\mathcal{X} = \mathcal{Y} = S^2$) At first, it might seem counter-intuitive that the feature maps in a spherical CNN have domain SO(3) instead of S^2 after the first layer. Hence, it is instructive to study a convolutional integral with $\mathcal{X} = \mathcal{Y} = S^2$ constructed using the techniques of Sect. 4. The action of SO(3) on S^2 is as above by matrix–vector product. We next choose an arbitrary reference point x_0 on the sphere and denote by R_x the rotation

matrix which rotates x_0 into x: $R_x x_0 = x$. Following (117), we can then write the convolution as

$$[\kappa \star f](y) = \int_{\mathcal{X}} \rho_2(R_x)\kappa(R_x^{-1}y)\rho_1^{-1}(R_x)f(x)\,dx\,. \tag{154}$$

However, the element R_x is not unique: If Q rotates around x_0, $Qx_0 = x_0$, then $R_x Q$ also rotates x_0 into x. In fact, the symmetry group splits into a semi-direct product $SO(3) = N \rtimes H$ with $H = SO(2)$ the stabilizer of x_0 and $N = SO(3)/H$. This special case was considered in Sect. 4.4 and we can write (154) as

$$[\kappa \star f](y) = \int_{SO(3)/H} \rho_2(R)\kappa(R^{-1}y)\rho_1^{-1}(R)f(Rx_0)\,dR\,. \tag{155}$$

According to (130), the kernel κ is now not unconstrained anymore but satisfies

$$\kappa(Qy) = \rho_2(Q)\kappa(y)\rho_1^{-1}(Q)\,, \tag{156}$$

for $Q \in H$. In particular, if the input and output features transform like scalars, $\rho_1 = \rho_2 = \mathrm{id}$, (156) implies that the kernel is invariant under rotations around x_0, i.e. isotropic, as was noticed also in Makadia et al. (2007). In practice, this isotropy decreases the expressibility of the layer considerably.

6.2 Spherical convolutions and Fourier transforms

The Fourier decomposition on the sphere and on $SO(3)$ is well studied and can be used to find compact explicit expressions for the integrals defined in the previous section. To simplify some expressions, we will assume in this and the following section that $V_{1,2}$ are vector spaces over $\mathbb{R}$, so in particular $\rho_{1,2}$ are real representations.

A square-integrable function $f : S^2 \to \mathbb{R}^c$ can be decomposed into the spherical harmonics Y_m^ℓ via

$$f(x) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \hat{f}_m^\ell Y_m^\ell(x)\,, \tag{157}$$

with Fourier coefficients

$$\hat{f}_m^\ell = \int_{S^2} f(x)\overline{Y_m^\ell(x)}\,dx\,, \tag{158}$$

since the spherical harmonics form a complete orthogonal set,

$$\sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \overline{Y_m^\ell(x)}Y_m^\ell(y) = \delta(x - y)\,, \tag{159}$$

$$\int_{S^2} \overline{Y_{m_1}^{\ell_1}(x)}Y_{m_2}^{\ell_2}(x)dx = \delta_{\ell_1\ell_2}\delta_{m_1m_2}\,. \tag{160}$$

For later convenience, we also note the following property of spherical harmonics:

$$\overline{Y_m^\ell(x)} = (-1)^m Y_{-m}^\ell(x)\,. \tag{161}$$

In practice, one truncates the sum over ℓ at some finite L, the bandwidth, obtaining an approximation of f.

Similarly, a square-integrable function $f : \mathrm{SO}(3) \to \mathbb{R}^c$ can be decomposed into Wigner D-matrices $\mathcal{D}_{mn}^\ell(R)$ (for a comprehensive review, see Varshalovich et al. (1988)) via

$$f(R) = \sum_{\ell=0}^{\infty} \sum_{m,n=-\ell}^{\ell} \hat{f}_{mn}^\ell \mathcal{D}_{mn}^\ell(R), \tag{162}$$

with Fourier coefficients

$$\hat{f}_{mn}^\ell = \frac{2\ell+1}{8\pi^2} \int_{\mathrm{SO}(3)} f(R) \overline{\mathcal{D}_{mn}^\ell(R)} \, dR, \tag{163}$$

since the Wigner matrices satisfy the orthogonality and completeness relations

$$\int_{\mathrm{SO}(3)} \overline{\mathcal{D}_{m_1 n_1}^{\ell_1}(R)} \mathcal{D}_{m_2 n_2}^{\ell_2}(R) dR = \frac{8\pi^2}{2\ell_1+1} \delta_{\ell_1 \ell_2} \delta_{m_1 m_2} \delta_{n_1 n_2} \tag{164}$$

$$\sum_{\ell=0}^{\infty} \sum_{m,n=-\ell}^{\ell} \overline{\mathcal{D}_{mn}^\ell(Q)} \mathcal{D}_{mn}^\ell(R) = \frac{8\pi^2}{2\ell+1} \delta(R-Q). \tag{165}$$

Furthermore, the Wigner D-matrices form a unitary representation of SO(3) since they satisfy

$$\mathcal{D}_{mn}^\ell(QR) = \sum_{p=-\ell}^{\ell} \mathcal{D}_{mp}^\ell(Q) \mathcal{D}_{pn}^\ell(R), \tag{166}$$

$$\mathcal{D}_{mn}^\ell(R^{-1}) = (\mathcal{D}_{mn}^\ell(R))^{-1} = (\mathcal{D}_{mn}^\ell(R))^\dagger = \overline{\mathcal{D}_{nm}^\ell(R)}. \tag{167}$$

Note furthermore that

$$\overline{\mathcal{D}_{mn}^\ell(R)} = (-1)^{n-m} \mathcal{D}_{-m,-n}^\ell(R). \tag{168}$$

The regular representation of SO(3) on spherical harmonics of order ℓ is given by the corresponding Wigner matrices:

$$Y_m^\ell(Rx) = \sum_{n=-\ell}^{\ell} \overline{\mathcal{D}_{mn}^\ell(R)} Y_n^\ell(x). \tag{169}$$

A product of two Wigner matrices is given in terms of the Clebsch–Gordan coefficients $C_{\ell_1 m_1; \ell_2 m_2}^{JM}$ by

$$\mathcal{D}_{m_1 n_1}^{\ell_1}(R) \mathcal{D}_{m_2 n_2}^{\ell_2}(R) = \sum_{J=|\ell_1-\ell_2|}^{\ell_1+\ell_2} \sum_{M,N=-J}^{J} C_{\ell_1 m_1; \ell_2 m_2}^{JM} C_{\ell_1 n_1; \ell_2 n_2}^{JN} \mathcal{D}_{MN}^J(R). \tag{170}$$

We will now use these decompositions to write the convolutions (152) and (120) in the Fourier domain. To this end, we use Greek letters to index vectors in the spaces V_1 and V_2 in which the feature maps take values.

Proposition 6.2 *The Fourier transform of the spherical convolution (152) with $\mathcal{X} = S^2$ and $\mathcal{Y} = \mathrm{SO}(3)$ is given by*

$$\left[\widehat{(\kappa \star f)_\mu}\right]_{mn}^\ell = \sum_{\nu=1}^{\dim V_2} \sum_{\sigma,\tau=1}^{\dim V_1} \sum_{\substack{\ell_i=0 \\ i=1,2,3}}^{\infty} \sum_{\substack{m_i,n_i=-\ell_i \\ i=1,2,3}}^{\ell_i} \sum_{J=|\ell_2-\ell_1|}^{\ell_2+\ell_1} \sum_{M,N=-J}^{J} C_{\ell_2 m_2; \ell_1 n_1}^{JM} C_{\ell_2 n_2; \ell_1 m_1}^{JN}$$

$$C_{JM;\ell_3 m_3}^{\ell m} C_{JN;\ell_3 n_3}^{\ell n} \widehat{(\rho_{2,\mu\nu})}_{m_2 n_2}^{\ell_2} \widehat{(\kappa_{\nu\sigma})}_{m_1}^{\ell_1} \overline{\widehat{\rho_{1,\sigma\tau}}_{n_3 m_3}^{\ell_3}} \overline{\widehat{(f_\tau)}_{n_1}^{\ell_1}}. \tag{171}$$

For $\rho_1 = \rho_2 = \mathrm{id}$, (171) simplifies to

$$\left[\widehat{(\kappa \star f)}_\mu\right]^\ell_{mn} = \sum_{\nu=1}^{\dim V_1} (\widehat{\kappa}_{\mu\nu})^\ell_n \overline{(\widehat{f}_\nu)^\ell_m} \,. \tag{172}$$

A similar calculation can be performed for input features in SO(3) as detailed by the following proposition.

Proposition 6.3 *The Fourier transform of the spherical convolution (120) with $\mathcal{X} = \mathcal{Y} =$ SO(3) can be written in the Fourier domain as*

$$\left[\widehat{(\kappa \star f)}\right]^\ell_{mn} = \frac{8\pi^2}{2\ell+1} \sum_{\substack{p=-\ell \\ }} \sum_{\substack{\ell_i=0 \\ i=1,2,3}}^{\ell} \sum_{\substack{m_i,n_i=-\ell_i \\ i=1,2,3}}^{\ell_i} \sum_{J=|\ell_1-\ell_2|}^{\ell_1+\ell_2} \sum_{M,N=-J}^{J} C^{JM}_{\ell_1 m_1; \ell_2 m_2} C^{JN}_{\ell_1 n_1; \ell_2 n_2}$$
$$C^{\ell m}_{JM; \ell_3 m_3} C^{\ell p}_{JN; \ell_3 n_3} \widehat{\rho 2}^{\ell_1}_{m_1 n_1} \cdot \widehat{\kappa}^\ell_{pn} \cdot \overline{\widehat{\rho 1}^{\ell_2}_{n_2 m_2}} \cdot \widehat{f}^{\ell_3}_{m_3 n_3} \,. \tag{173}$$

Here, the dot $\cdot$ denotes matrix multiplication in $V_{1,2}$, as spelled out in (171).

For $\rho_1 = \rho_2 = \mathrm{id}$ (173) becomes

$$\left[\widehat{(\kappa \star f)}\right]^\ell_{mn} = \frac{8\pi^2}{2\ell+1} \sum_{p=-\ell}^{\ell} \widehat{\kappa}^\ell_{pn} \cdot \widehat{f}^\ell_{mp} \,. \tag{174}$$

Note that in all these expressions, the Fourier transform is done component wise with respect to the indices in $V_{1,2}$.

6.3 Decomposition into irreducible representations

An immediate simplification of (171) and (173) can be achieved by decomposing $\rho_{1,2}$ into irreps of SO(3) which are given by the Wigner matrices $\mathcal{D}^L$,

$$\rho(R) = \bigoplus_\lambda \bigoplus_\mu \mathcal{D}^\lambda(R) \,, \tag{175}$$

where μ counts the multiplicity of $\mathcal{D}^\lambda$ in ρ. Correspondingly, the spaces $V_{1,2}$ are decomposed according to $V_{1,2} = \bigoplus_\lambda \bigoplus_\mu V_{1,2}^{\lambda\mu}$, where $V_{1,2}^{\lambda\mu} = \mathbb{R}^{2\lambda+1}$. The feature maps then carry indices $f_\nu^{\lambda\mu}$ with $\lambda = 0, \ldots, \infty$, $\mu = 1, \ldots, \infty$, $\nu = -\lambda, \ldots, \lambda$ with only finitely many non-zero components. Using this decomposition of $V_{1,2}$ the convolution (152) is given by

$$(\kappa \star f)_\nu^{\lambda\mu}(R) = \sum_{\rho=-\lambda}^{\lambda} \sum_{\theta=0}^{\infty} \sum_{\sigma=1}^{\infty} \sum_{\tau,\pi=-\theta}^{\theta} \int_{S^2} \mathcal{D}^\lambda_{\nu\rho}(R) \kappa^{\lambda\mu;\theta\sigma}_{\rho;\tau}(R^{-1}x) \mathcal{D}^\theta_{\tau\pi}(R^{-1}) f_\pi^{\theta\sigma}(x) \mathrm{d}x \,. \tag{176}$$

By plugging these expressions for $\rho_{1,2}$, κ and f into (171), we obtain the following proposition.

Proposition 6.4 *The decomposition of ([171](#)) into representation spaces of irreducible representations of* SO(3) *is given by*

$$
\left[\widehat{(\kappa \star f)_\nu^{\lambda\mu}}\right]_{mn}^\ell = \sum_{\rho=-\lambda}^{\lambda} \sum_{\theta=0}^{\infty} \sum_{\sigma=1}^{\infty} \sum_{\tau,\pi=-\theta}^{\theta} \sum_{j=0}^{\infty} \sum_{q,r=-j}^{\ell_i} \sum_{J=|\ell_2-j|}^{\ell_2+j} \sum_{M,N=-J}^{J} C_{\lambda\nu;jr}^{JM} C_{\lambda\rho;jq}^{JN}
$$
$$
C_{JM;\theta\tau}^{\ell m} C_{JN;\theta\pi}^{\ell n} (\widehat{\kappa_{\rho;\tau}^{\lambda\mu;\theta\sigma}})_q^j \overline{(\widehat{f_\pi^{\theta\sigma}})_r^j}. \tag{177}
$$

Similarly, ([173](#)) decomposes according to

$$
(\widehat{(\kappa \star f)_\nu^{\lambda\mu}})_{mn}^\ell = \frac{8\pi^2}{2\ell+1} \sum_{\rho=-\lambda}^{\lambda} \sum_{\theta=0}^{\infty} \sum_{\sigma=1}^{\infty} \sum_{\tau,\pi=-\theta}^{\theta} \sum_{p=-\ell}^{\ell} \sum_{j=0}^{\infty} \sum_{q,r=-j}^{j} \sum_{J=|\lambda-\sigma|}^{\lambda+\sigma} \sum_{M,N=-J}^{J} C_{\lambda\nu;\theta\tau}^{JM} C_{\lambda\rho;\theta\pi}^{JN}
$$
$$
C_{JM;jq}^{\ell m} C_{JN;jr}^{\ell p} (\widehat{\kappa_{\rho;\tau}^{\lambda\mu;\theta\sigma}})_{pn}^\ell (\widehat{f_\pi^{\theta\sigma}})_{qr}^j. \tag{178}
$$

In these expressions, the Fourier transform of the convolution is given entirely in terms of the Fourier transforms of the kernel and the input feature map, as well as Clebsch–Gordan coefficients. In particular, Fourier transforms of the representation matrices $\rho_{1,2}$ are trivial in this decomposition of the spaces V_1 and V_2.

6.4 Output features in SE(3)

As an example of a possible application of the techniques presented in this section, consider the problem of 3D object detection in pictures taken by fisheye cameras. These cameras have a spherical image plane and therefore, the components of the input feature map f transform as scalars under the regular representation (cf. ([104](#))):

$$
[\pi_1(R)f](x) = f(R^{-1}x). \tag{179}
$$

Since this is the transformation property considered in the context of spherical CNNs, the entire network can be built using the layers discussed in this section.

As detailed in Sect. [5.3](#), for object detection, we want to identify the class, physical size, position and orientation for each object in the image. This can be realized by associating to each pixel in the output picture (which is often of lower resolution than the input) a class probability vector $p \in P(\Omega)$, a size vector $s \in \mathbb{R}^3$ containing height, width and depth of the object, a position vector $x \in \mathbb{R}^3$ and an orientation given by a matrix $Q \in$ SO(3). Here, $P(\Omega)$ is the space of probability distributions in N classes as in ([2](#)). In the framework outlined above, this means that the output feature map takes values in $P(\Omega) \oplus \mathbb{R}^3 \oplus \mathbb{R}^3 \oplus$ SO(3).

If the fisheye camera rotates by $R \in$ SO(3), the output features have to transform accordingly. In particular, since the classification and the size of the object do not depend on the rotation, p transforms as N scalars and s transforms as three scalars:

$$
\rho_2(R)p = p, \qquad \rho_2(R)s = s, \tag{180}
$$

where we used the notation introduced in ([105](#)). The position vector x on the other hand transforms in the fundamental representation of SO(3)

$$
\rho_2(R)x = R \cdot x. \tag{181}
$$

Finally, the rotation matrix Q transforms by a similarity transformation

$$
\rho_2(R)Q = R \cdot Q \cdot R^T. \tag{182}
$$

As described above for the general case, the transformation property (180)–(182) of the output feature map can be decomposed into a direct sum of irreducible representations, labeled by an integer ℓ. The scalar transformations of (180) are $N + 3$ copies of the $\ell = 0$ representation and the fundamental representation in (181) is the $\ell = 1$ representation. To decompose the similarity transformation in (182), we use the following

Proposition 6.5 *Let A, B, C be arbitrary matrices and* $\text{vec}(M)$ *denote the concatenation of the columns of the matrix M. Then,*

$$\text{vec}(A \cdot B \cdot C) = (A \otimes C^T) \cdot \text{vec}(B) \,, \tag{183}$$

where $\otimes$ denotes the Kronecker product.

Remark 6.6 To illustrate the matrix dimensions in (183), consider $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times k}$ and $C \in \mathbb{R}^{k \times \ell}$, hence $A \cdot B \cdot C \in \mathbb{R}^{m \times \ell}$ and $\text{vec}(A \cdot B \cdot C) \in \mathbb{R}^{m\ell}$. On the other hand, $A \otimes C^T \in \mathbb{R}^{m\ell \times nk}$ and $\text{vec}(B) \in \mathbb{R}^{nk}$ and hence the two can be multiplied by a matrix–vector product, yielding also a vector in $\mathbb{R}^{m\ell}$.

With this, we obtain for (182)

$$\rho_2(R) \, \text{vec}(Q) = (R \otimes R) \cdot \text{vec}(Q) \,, \tag{184}$$

i.e. Q transforms with the tensor product of two fundamental representations. According to (170), this tensor product decomposes into a direct sum of one $\ell = 0$, one $\ell = 1$ and one $\ell = 2$ representation. In total, the final layer of the network will therefore have (in the notation introduced below (175)) $\lambda = 0, 1, 2$, $\nu = -\lambda, \ldots, \lambda$ and $\mu = 1, \ldots, N + 4$ for $\lambda = 0$, $\mu = 1, 2$ for $\lambda = 1$ and $\mu = 1$ for $\lambda = 2$.

Note that the transformation properties (180)–(182) of the output feature map are independent of the transformation properties of the input feature map. We have restricted the discussion here to fisheye cameras since, as stated above, these can be realized by the spherical convolutions considered in this section. For pinhole cameras on the other hand, the representation π_1 acting on the input features will be a complicated non-linear transformation arising from the projection of the transformation in 3D space onto the flat image plane. Working out the details of this representation is an interesting direction for further research.

6.5 SE(3) equivariant networks

The same decomposition of $V_{1,2}$ into representation spaces of irreducible representations of SO(3), which was used in Sect. 6.3 can also be used to solve the kernel constraint (132) for equivariant networks of SE(3), as discussed in Weiler et al. (2018). In this section, we review this construction.

In the decomposition into irreps, the kernel constraint (132) reads

$$\kappa_{\rho;\tau}^{\lambda\mu;\theta\sigma}(Ry) = \sum_{\nu=-\lambda}^{\lambda} \sum_{\pi=-\theta}^{\theta} \mathcal{D}_{\rho\nu}^{\lambda}(R) \, \kappa_{\nu;\pi}^{\lambda\mu;\theta\sigma}(y) \, \mathcal{D}_{\pi\tau}^{\theta}(R^{-1}) \,. \tag{185}$$

In the following, we will use $\cdot$ to denote matrix multiplication in ρ, ν, π, τ and drop the multiplicity indices μ, σ since (185) is a component-wise equation with respect to the multiplicity.

On the right-hand side of (185), SO(3) acts in a tensor product representation on the kernel. To make this explicit, we use the vectorization (183) and the unitarity of the Wigner

matrices to rewrite (185) into[11]

$$\text{vec}(\kappa^{\lambda\theta}(Ry)) = (\mathcal{D}^\lambda(R) \otimes \overline{\mathcal{D}^\theta}(R)) \cdot \text{vec}(\kappa^{\lambda\theta}(y)). \tag{186}$$

Using (168) and (170), we can decompose the tensor product of the two Wigner matrices into a direct sum of Wigner matrices $\mathcal{D}^J$ with $J = |\lambda - \theta|, \ldots, \lambda + \theta$. Performing the corresponding change of basis for $\text{vec}(\kappa^{\lambda\theta})$ leads to components $\text{vec}(\kappa^{\lambda\theta;J})$ on which the constraint takes the form

$$\text{vec}(\kappa^{\lambda\theta;J}(Ry)) = \mathcal{D}^J(R) \cdot \text{vec}(\kappa^{\lambda\theta;J}(y)). \tag{187}$$

According to (161) and (169), the spherical harmonics Y_m^ℓ solve this constraint and they in fact span the space of solutions with respect to the angular dependence of $\kappa^{\lambda\theta,J}$. Therefore, a general solution of (187) has the form

$$\text{vec}(\kappa^{\lambda\theta;J}(y)) = \sum_k \sum_{m=-J}^{J} w_{k,m}^{\lambda\theta;J} \varphi^k(||y||) Y_m^J(y), \tag{188}$$

with radial basis functions φ^k and (trainable) coefficients w.

As an example of an application of group equivariant network architectures, we considered spherical CNNs in this section, which are of great practical importance. Spherical convolutions serve as a good example of the Fourier perspective on group equivariant convolutions since the spectral theory on the sphere and rotation group SO(3) is well-understood. Consequently, in Proposition 6.4, we could give explicit, yet completely general expressions for spherical convolutions for feature maps transforming in arbitrary representations of SO(3), given just in terms of Clebsch–Gordan coefficients.

In principle, such expressions could be derived for a large class of symmetry groups. The foundation of this generalization was laid in Lang and Weiler (2020), where it was shown how the kernel constraint for any compact symmetry group can be solved in terms of well-known representation theoretic quantities. In position space, algorithms already exist which can solve the symmetry constraint and generate equivariant architectures automatically Finzi et al. (2021).

7 Conclusions

In this paper we have reviewed the recent developments in geometric deep learning and presented a coherent mathematical framework for equivariant neural networks. In the process we have also developed the theory in various directions, in an attempt to make it more coherent and emphasize the geometric perspective on equivariance.

Throughout the paper we have used the examples of equivariant semantic segmentation and object detection networks to illustrate equivariant CNNs. In particular, in Sect. 6.4 we showed that in rotation-equivariant object detection using fisheye cameras, the input features transform as scalars with respect to the regular representation of $R \in \text{SO}(3)$, while the output features take values in SE(3). It would be very interesting to generalize this to object detection that is equivariant with respect to SE(3) instead of SO(3). In this case, we would add translations of the two-dimensional image plane $\mathbb{R}^2 \subset \mathbb{R}^3$ and so the regular representation of SE(3) needs to be projected onto the image plane. For instance, translations which change

---

[11] Note that there is a typo in the corresponding equation (13) in Weiler et al. (2018), where the complex conjugation is missing.

the distance of the object to the image plane will be projected to scalings. If this is extended to pinhole cameras, the resulting group action will be highly non-linear.

Yet another interesting open problem is the development of an unsupervised theory of deep learning on manifolds. This would require to develop a formalism of *group equivariant generative networks*. For example, one would like to construct group equivariant versions of variational autoencoders, deep Boltzmann machines and GANs (see e.g. Venkatesh et al. (2020)).

An interesting aspect of equivariant neural networks is their stability with respect to data perturbations and transformations which are close to, but not exact group actions. Since real world data is often noisy this is an important property which was studied in the context of wavelets Mallat (2012) and GENEOs Frosini and Jabłoński (2016). It would be interesting to extend these considerations to the gauge equivariant layers discussed in Sect. 2.

As we have emphasized in this work, the feature maps in gauge equivariant CNNs can be viewed as sections of principal (frame) bundles, which are generally called *fields* in theoretical physics. The basic building blocks of these theories are special sections corresponding to irreducible representations of the gauge group; these are the *elementary particles* of Nature. It is tantalizing to speculate that this notion could also play a key role in deep learning, in the sense that a neural network gradually learns more and more complex feature representations which are built from "elementary feature types" arising from irreducible representations of the equivariance group.

The concept of equivariance to symmetries has been a guiding design principle for theoretical physics throughout the past century. The standard model of particle physics and the general theory of relativity provide prime examples of this. In physics, the fundamental role of symmetries is related to the fact that to every (continuous) symmetry is associated through Noether's theorem a conserved physical quantity. For example, equivariance with respect to time translations corresponds to conservation of energy during the evolution of a physical system. It is interesting to speculate that equivariance to global and local symmetries may play similar key roles in building neural network architectures, and that the associated conserved quantities can be used to understand the dynamics of the evolution of the network during training. Steps in this direction have been taken to examine and interpret the symmetries and conserved quantities associated to different gradient methods and data augmentation during the training of neural networks Głuch and Urbanke (2021). In the application of neural networks to model physical systems, several authors have also constructed equivariant (or invariant) models by incorporating equations of motion—in either the Hamiltonian or Lagrangian formulation of classical mechanics—to accommodate the learning of system dynamics and conservation laws (Greydanus et al. 2019; Toth et al. 2020; Cranmer et al. 2020). Along these lines, it would be very interesting to look for the general analogue of Noether's theorem in equivariant neural networks, and understand the importance of the corresponding conserved quantities for the dynamics of machine learning.

We hope we have convinced the reader that geometric deep learning is an exciting research field with interesting connections to both mathematics and physics, as well as a host of promising applications in artificial intelligence, ranging from autonomous driving to biomedicine. Although a huge amount of progress has been made, it is fair to say that the field is still in its infancy. In particular, there is a need for a more foundational understanding of the underlying mathematical structures of neural networks in general, and equivariant neural networks in particular. It is our hope that this paper may serve as a bridge connecting mathematics with deep learning, and will provide seeds for fruitful interactions across the fields of machine learning, mathematics and theoretical physics.

References

Albin P (xxx) "Linear analysis on manifolds." https://faculty.math.illinois.edu/palbin/analysisonmfds/lecturenotes.pdf

Aronsson J (July, 2022), "Homogeneous vector bundles and G-equivariant convolutional neural networks,"Sampling Theory, Signal Processing, and Data Analysis 20, arXiv:2105.05400. https://doi.org/10.1007/s43670-022-00029-3

Bekkers EJ, Lafarge MW, Veta M, Eppenhof KAJ, Pluim JPW, Duits R (2018) "Roto-translation covariant convolutional networks for medical image analysis," in Medical Image Computing and Computer Assisted Intervention - MICCAI 2018, Frangi AF, Schnabel JA, Davatzikos C, Alberola-López C, Fichtinger G, eds., Lecture Notes in Computer Science, pp. 440–448. Springer International Publishing

Bergomi MG, Frosini P, Giorgi D, Quercioli N (2019) Towards a topological-geometrical theory of group equivariant non-expansive operators for data analysis and machine learning. Nat Mach Intell 1:423–433

Boomsma W, Frellsen J (2017) "Spherical convolutions and their application in molecular modelling," in Advances in Neural Information Processing Systems, Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R, eds., vol. 30, pp. 3433–3443. Curran Associates, Inc.,

Bronstein MM, Bruna J, Cohen T, Veličković P (2021) "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges," arXiv:2104.13478

Bronstein MM, Bruna J, LeCun Y, Szlam A, Vandergheynst P (2017) "Geometric deep learning: Going beyond euclidean data,". IEEE Signal Process. Mag. 34:18–42. https://doi.org/10.1109/msp.2017.2693418. arXiv:1611.08097

Chen MX, Firat O, Bapna A, Johnson M, Macherey W, Foster G, Jones L, Schuster M, Shazeer N, Parmar N, Vaswani A, Uszkoreit J, Kaiser L, Chen Z, Wu Y, Hughes M (2018) "The best of both worlds: Combining recent advances in neural machine translation," in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), NIPS'17, pp. 6000–6010. Association for Computational Linguistics, Red Hook, NY, USA. arXiv:1706.03762. https://doi.org/10.18653/v1/p18-1008

Cheng MCN, Anagiannis V, Weiler M, de Haan P, Cohen TS, Welling M (2019) "Covariance in physics and convolutional neural networks," arXiv:1906.02481

Cobb OJ, Wallis CGR, Mavor-Parker AN, Marignier A, Price MA, d'Avezac M, McEwen JD (2020) "Efficient generalized spherical cnns," arXiv:2010.11661

Cohen TS, Geiger M, Köhler J, Welling M (2018) "Spherical cnns," in International Conference on Learning Representations. arXiv:1801.10130

Cohen TS, Geiger M, Weiler M (2018a) "Intertwiners between induced representations (with applications to the theory of equivariant neural networks)," arXiv:1803.10743

Cohen TS, Weiler M, Kicanaoglu B, Welling M (2019) "Gauge equivariant convolutional networks and the icosahedral cnn," in Proceedings of the 36th International Conference on Machine Learning, Chaudhuri K, Salakhutdinov R, eds., vol. 97 of Proceedings of Machine Learning Research, pp. 1321–1330. PMLR. arXiv:1902.04615

Cohen TS, Welling M (2016) "Group equivariant convolutional networks," in Proceedings of The 33rd International Conference on Machine Learning, Balcan MF, Weinberger KQ, eds., vol. 48 of Proceedings of Machine Learning Research, pp. 2990–2999. PMLR, New York, New York, USA. arXiv:1602.07576

Cohen T, Geiger M, Weiler M (2019) "A general theory of equivariant cnns on homogeneous spaces," in Advances in Neural Information Processing Systems, Wallach H, Larochelle H, Beygelzimer A, F. d'Alché-Buc, E. Fox, and R. Garnett, eds., vol. 32. Curran Associates, Inc.,. arXiv:1811.02017

Conti P, Frosini D, Quercioli N (2022) On the construction of group equivari- ant non-expansive operators via permutants and symmetric functions. Frontiers in Artificial Intelligence 5:1–11

Coors B, Condurache AP, Geiger A (2018) Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In: Hebert M, Sminchisescu C, Weiss Y (eds) Computer Vision - ECCV 2018, vol Ferrari. Springer International Publishing, Cham, pp 525–541

Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S, Schiele B (2016) "The cityscapes dataset for semantic urban scene understanding," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr.2016.350

Cranmer M, Greydanus S, Hoyer S, Battaglia P, Spergel D, Ho S (Mar., 2020) "Lagrangian neural networks," in Proceedings of the International Conference on Learning Representations - ICLR 2020. arXiv:2003.04630

de Haan P, Weiler M, Cohen T, Welling M (2020) "Gauge equivariant mesh cnns: Anisotropic convolutions on geometric graphs," arXiv:2003.05425

Defferrard M, Milani M, Gusset F, Perraudin N (2020) "Deepsphere: A graph-based spherical cnn," in International Conference on Learning Representations. arXiv:2012.15000

Deitmar A, Echterhoff S (2014) Principles of Harmonic Analysis. Springer International Publishing. https://doi.org/10.1007/978-3-319-05792-7

Dieleman S, Fauw JD, Kavukcuoglu K (2016) "Exploiting cyclic symmetry in convolutional neural networks," in International Conference on Machine Learning, pp. 1889–1898. PMLR. arXiv:1602.02660

Elaldi A, Dey N, Kim H, Gerig G (2021) "Equivariant spherical deconvolution: Learning sparse orientation distribution functions from spherical data," arXiv:2102.09462

Elesedy B, Zaidi S (2021) "Provably strict generalisation benefit for equivariant models," arXiv:2102.10333

Esteves C (2020) "Theoretical aspects of group equivariant neural networks," arXiv:2004.05154

Esteves C, Allen-Blanchette C, Makadia A, Daniilidis K (2018) "Learning so(3) equivariant representations with spherical cnns," in Proceedings of the European Conference on Computer Vision (ECCV), pp. 52–68 arXiv:1711.06721

Esteves C, Makadia A, Daniilidis K (Oct., 2020) "Spin-weighted spherical cnns," in Advances in Neural Information Processing Systems, Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H, eds., vol. 33, pp. 8614–8625. Curran Associates, Inc., arXiv:2006.10731

Favoni M, Ipp A, Müller DI, Schuh D (Jan., 2022) "Lattice gauge equivariant convolutional neural networks,"Phys. Rev. Lett. 128 arXiv:2012.12901. https://doi.org/10.1103/physrevlett.128.032003

Finzi M, Welling M, Wilson AG (2021) "A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups," arXiv:2104.09459

Folland GB (Feb., 2016) A Course in Abstract Harmonic Analysis. Chapman and Hall/CRC, https://doi.org/10.1201/b19172

Fox J, Zhao B, Rajamanickam S, Ramprasad R, Song L (2021) "Concentric spherical gnn for 3d representation learning," arXiv:2103.10484

Frosini P, Jabłoński G (2016) "Combining persistent homology and invariance groups for shape comparison,". Discrete & Computational Geometry 55:373–409. https://doi.org/10.1007/s00454-016-9761-y

Führ H (2005) Abstract Harmonic Analysis of Continuous Wavelet Transforms. Springer, Berlin Heidelberg. https://doi.org/10.1007/b104912

Garcia Satorras V, Hoogeboom E, Fuchs F, Posner I, Welling M (2021) "E(n) equivariant normalizing flows," in Advances in Neural Information Processing Systems, Ranzato M, Beygelzimer A, Dauphin Y, Liang P, Vaughan JW, eds., vol. 34, pp. 4181–4192. Curran Associates, Inc

Gerken J, Carlsson O, Linander H, Ohlsson F, Petersson C, Persson D (2022) "Equivariance versus augmentation for spherical images," in Proceedings of The 39th International Conference on Machine Learning, Balcan MF, Weinberger KQ, eds., vol. 162 of Proceedings of Machine Learning Research, pp. 7404–7421. PMLR, New York, New York, USA. arXiv:2202.03990

Głuch G, Urbanke R (2021) "Noether: The more things change, the more stay the same," arXiv:2104.05508

Grafarend EW, Kühnel W (2011) A minimal atlas for the rotation group SO(3). GEM-International Journal on Geomathematics 2(1):113–122

Greydanus S, Dzamba M, Yosinski J (June, 2019) "Hamiltonian neural networks," in Proceedings of the Conference on Neural Information Processing Systems - NeurIPS 2019. arXiv:1906.01563

Hamilton MJ (2017) Mathematical Gauge Theory. Universitext. Springer International Publishing. https://doi.org/10.1007/978-3-319-68439-0

Hoogeboom E, Peters JW, Cohen TS, Welling M (2018) "Hexaconv," in International Conference on Learning Representations. arXiv:1803.02108

Howard R (1994) "Analysis on homogeneous spaces." https://people.math.sc.edu/howard/notes/harmonic.pdf. Accessed May 27, 2021

Jiang CM, Huang J, Kashinath K, Prabhat, Marcus P, Niessner M (Sept., 2018) "Spherical cnns on unstructured grids," in International Conference on Learning Representations

Kaniuth E, Taylor KF (2012) Induced Representations of Locally Compact Groups. Cambridge University Press. https://doi.org/10.1017/cbo9781139045391

Kolář I, Slovák J, Michor PW (1993) Natural Operations in Differential Geometry. Springer Berlin Heidelberg, 1993rd edition ed. https://doi.org/10.1007/978-3-662-02950-3

Kondor R (2018) "N-body networks: A covariant hierarchical neural network architecture for learning atomic potentials," arXiv:1803.01588

Kondor R, Lin Z, Trivedi S (Nov 2018) "Clebsch-Gordan nets: A fully fourier space spherical convolutional neural network," in Advances in Neural Information Processing Systems, Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, Garnett R, eds., vol. 31. Curran Associates, Inc., arXiv:1806.09231

Kondor R, Trivedi S (2018) "On the generalization of equivariance and convolution in neural networks to the action of compact groups," in Proceedings of the 35th International Conference on Machine Learning, Dy J, Krause A, eds., vol. 80 of Proceedings of Machine Learning Research, pp. 2747–2755. PMLR. arXiv:1802.03690

Lang L, Weiler M (2020) "A Wigner-Eckart Theorem for Group Equivariant Convolution Kernels," arXiv:2010.10952

Lee JM (2012) Introduction to Smooth Manifolds. Springer, New York. https://doi.org/10.1007/978-1-4419-9982-5

Luo D, Carleo G, Clark BK, Stokes J (Dec., 2021) "Gauge equivariant neural networks for quantum lattice gauge theories,"Phys. Rev. Lett. 127 arXiv:2012.05232. https://doi.org/10.1103/physrevlett.127.276402

Makadia A, Geyer C, Daniilidis K (2007) "Correspondence-free structure from motion,". Int. J. Comput. Vision 75:311–327. https://doi.org/10.1007/s11263-007-0035-2

Mallat S (2012) "Group invariant scattering,". Commun. Pure Appl. Math. 65:1331–1398. https://doi.org/10.1002/cpa.21413

Marcos D, Volpi M, Komodakis N, Tuia D (Oct., 2017) "Rotation equivariant vector field networks," in 2017 IEEE International Conference on Computer Vision (ICCV), pp. 5058–5067. IEEE. https://doi.org/10.1109/iccv.2017.540

Marsh A (2019a) "Gauge theories and fiber bundles: Definitions, pictures, and results," arXiv:1607.03089

McEwen JD, Wallis CGR, Mavor-Parker AN (2021) "Scattering networks on the sphere for scalable and rotationally equivariant spherical cnns," arXiv:2102.02828

Melzi S, Spezialetti R, Tombari F, Bronstein MM, Stefano LD, Rodola E (June, 2019) "Gframes: Gradient-based local reference frame for 3d shape matching," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4629–4638. IEEE. https://doi.org/10.1109/cvpr.2019.00476

Monroy R, Lutz S, Chalasani T, Smolic A (2018) "Salnet360: Saliency maps for omni-directional images with cnn,". Signal Process. Image Commun. 69:26–34. https://doi.org/10.1016/j.image.2018.05.005

Müller P, Golkov V, Tomassini V, Cremers D (2021) "Rotation-equivariant deep learning for diffusion mri," arXiv:2102.06942

Nakahara M (Oct., 2018) Geometry, Topology and Physics. CRC Press, Bristol ; Philadelphia, 2 edition ed.,. https://doi.org/10.1201/9781315275826

Perraudin N, Defferrard M, Kacprzak T, Sgier R (2019) "Deepsphere: Efficient spherical convolutional neural network with healpix sampling for cosmological applications,". Astron. Comput. 27:130–146. https://doi.org/10.1016/j.ascom.2019.03.004. arXiv:1810.12186

Romero DW, Bekkers EJ, Tomczak JM, Hoogendoorn M (2020) "Attentive group equivariant convolutional networks," in Proceedings of the 37th International Conference on Machine Learning, H. D. III and A. Singh, eds., vol. 119 of Proceedings of Machine Learning Research, pp. 8188–8199. PMLR. arXiv:2002.03830

Santaló LA, Kac M (2004) Integral Geometry and Geometric Probability. Cambridge University Press. https://doi.org/10.1017/cbo9780511617331

Shen Z, Shen T, Lin Z, Ma J (2021) PDO-eS2CNNs: Partial Differential Operator Based Equivariant Spherical CNNs. Proceedings of the AAAI Conference on Artificial Intelligence 35:9585–9593. https://doi.org/10.1609/aaai.v35i11.17154. arXiv:2104.03584

Steenrod N (1999) The topology of fibre bundles. Princeton University Press

Su Y-C, Grauman K (2017) "Learning spherical convolution for fast features from 360° imagery," in Advances in Neural Information Processing Systems, Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R, eds., vol. 30. Curran Associates, Inc., arXiv:1708.00919

Thomas N, Smidt T, Kearnes S, Yang L, Li L, Kohlhoff K, Riley P (2018) "Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds," arXiv:1802.08219

Toft C, Bökman G, Kahl F (2021) "Azimuthal rotational equivariance in spherical cnns."

Toth P, Rezende DJ, Jaegle A, Racanière S, Botev A, Higgins I (2020) "Hamiltonian generative networks," in International Conference on Learning Representations. arXiv:1909.13789

Varshalovich DA, Moskalev AN, Khersonskii VK (Oct., 1988) Quantum Theory of Angular Momentum. WORLD SCIENTIFIC. https://doi.org/10.1142/0270

Venkatesh B, Shaht T, Chen A, Ghafurian S (Apr., 2020) "Restoration of marker occluded hematoxylin and eosin stained whole slide histology images using generative adversarial networks," in 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI). IEEE. arXiv:2005.01683. https://doi.org/10.1109/isbi45749.2020.9098358

Wallach NR (1973) Harmonic Analysis on Homogeneous Spaces. American Mathematical Society. https://doi.org/10.1090/pspum/026

Weiler M, Cesa G (2019) "General e(2)-equivariant steerable cnns," in Advances in Neural Information Processing Systems, Wallach H, Larochelle H, Beygelzimer A, F. d'Alché-Buc, Fox E, Garnett R, eds., vol. 32. Curran Associates, Inc. arXiv:1911.08251

Weiler M, Hamprecht FA, Storath M (June, 2018) "Learning steerable filters for rotation equivariant cnns," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, NIPS'18, pp. 10402–10413. IEEE, Red Hook, NY, USA. arXiv:1807.02547. https://doi.org/10.1109/cvpr.2018.00095

Weiler M, Hamprecht FA, Storath M (June, 2018) "Learning steerable filters for rotation equivariant cnns," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 849–858. IEEE. arXiv:1711.07289. https://doi.org/10.1109/cvpr.2018.00095

Wiersma R, Eisemann E, Hildebrandt K (Aug., 2020) "Cnns on surfaces using rotation-equivariant features,"ACM Trans. Graphics 39, arXiv:2006.01570. https://doi.org/10.1145/3386569.3392437

Winkels M, Cohen TS (Apr., 2018) "3d g-CNNs for pulmonary nodule detection," in International conference on Medical Imaging with Deep Learning. arXiv:1804.04656

Worrall DE, Garbin SJ, Turmukhambetov D, Brostow GJ (July, 2017) "Harmonic networks: Deep translation and rotation equivariance," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 567–584. IEEE. arXiv:1804.04458. https://doi.org/10.1109/cvpr.2017.758

Worrall DE, Garbin SJ, Turmukhambetov D, Brostow GJ (July, 2017) "Harmonic networks: Deep translation and rotation equivariance," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7168–7177. IEEE. arXiv:1612.04642. https://doi.org/10.1109/cvpr.2017.758

Authors and Affiliations

Jan E. Gerken[1,2,3] ⓘ **· Jimmy Aronsson**[1] **· Oscar Carlsson**[1] ⓘ **· Hampus Linander**[4] **· Fredrik Ohlsson**[5] ⓘ **· Christoffer Petersson**[1,6] **· Daniel Persson**[1] ⓘ

Jan E. Gerken
gerken@chalmers.se

Jimmy Aronsson
jimmyar@chalmers.se

Oscar Carlsson
osccarls@chalmers.se

Hampus Linander
hampus.linander@gu.se

Fredrik Ohlsson
fredrik.ohlsson@umu.se

Christoffer Petersson
christoffer.petersson@chalmers.se

1 Department of Mathematical Sciences, Chalmers University of Technology & University of Gothenburg, 412 96 Gothenburg, Sweden

2 Machine Learning Group, Berlin Institute of Technology (TU Berlin), 10587 Berlin, Germany

3 Berlin Institute for the Foundations of Learning and Data (BIFOLD), Berlin, Germany

4 Department of Physics, Gothenburg University, 412 96 Gothenburg, Sweden

5 Department of Mathematics and Mathematical Statistics, Umeå University, 901 87 Umeå, Sweden

6 Zenseact, 417 56 Gothenburg, Sweden