

# Enabling Equivariance for Arbitrary Lie Groups

Lachlan E. MacDonald\*, Sameera Ramasinghe, Simon Lucey  
 Australian Institute for Machine Learning  
 University of Adelaide, Australia

## Abstract

Although provably robust to translational perturbations, convolutional neural networks (CNNs) are known to suffer from extreme performance degradation when presented at test time with more general geometric transformations of inputs. Recently, this limitation has motivated a shift in focus from CNNs to Capsule Networks (CapsNets). However, CapsNets suffer from admitting relatively few theoretical guarantees of invariance. We introduce a rigorous mathematical framework to permit invariance to any Lie group of warps, exclusively using convolutions (over Lie groups), without the need for capsules. Previous work on group convolutions has been hampered by strong assumptions about the group, which precludes the application of such techniques to common warps in computer vision such as affine and homographic. Our framework enables the implementation of group convolutions over **any** finite-dimensional Lie group. We empirically validate our approach on the benchmark affine-invariant classification task, where we achieve  $\sim 30\%$  improvement in accuracy against conventional CNNs while outperforming most CapsNets. As further illustration of the generality of our framework, we train a homography-convolutional model which achieves superior robustness on a homography-perturbed dataset, where CapsNet results degrade.

## 1. Introduction

Symmetry (reversible, composable change) is ubiquitous in the visual reality perceived by humans. It is therefore no surprise that symmetry now occupies a fundamental and increasingly well-understood role in machine learning, particularly in computer vision [6, 5, 8, 33, 34, 31, 11, 22, 3, 32]. *Invariance* with respect to symmetries is particularly desirable in machine learning. Presently this is frequently achieved by training models on large volumes of symmetry-perturbed data. However, where there is a paucity of data or a need for explainability, architectural approaches such as

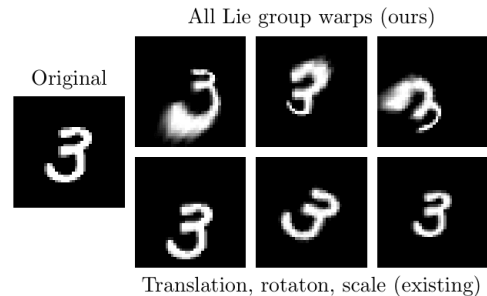


Figure 1: Domains of applicability of our framework and existing convolutional frameworks.

CapsNets may be preferable [16, 29, 13].

CapsNets admit very few theoretical guarantees of invariance [19]. However they are not the only means of achieving architectural invariance. Provable architectural invariance to *translational symmetries* is already pervasive in computer vision in the form of CNNs. The convolutions implemented in modern CNNs are, by definition, sums over responses between filter and feature, as the filter is translated across different positions. Crucially, these operations are *equivariant* in the sense that translating the input feature and then convolving is the same as first convolving and then translating the response. Carefully designed downsampling operations [37] turn this equivariance into *invariance*, so that translation of the input signal does not change the response. The respect of CNNs for the spatial locality inherent in natural images, together with their equi/in-variance, constitute the inductive biases that have seen CNNs become the industry standard in many computer vision tasks.

More recently still, in the seminal paper by Cohen et al. [6], it was observed that the translational convolutions of CNNs could be replaced by more general *group convolutions* that have for decades played an important role in mathematics, particularly in operator theory and noncommutative geometry [9, 17]. Given a group  $G$  of warps, a  $G$ -convolution is a sum over *warps* belonging to  $G$  of responses between a warped filter and feature; when  $G$  is the group of translations, one recovers ordinary convolution as

\*lachlan.macdonald@adelaide.edu.au

used in conventional CNNs. When built into neural networks, these group convolutions achieve better weight sharing and superior performance on image classification tasks than their conventional cousins, especially on datasets consisting of  $G$ -warped data [6, 5, 8, 33, 34, 31, 11, 22, 3, 32].

One aspect of group-convolutional neural networks which has received relatively little attention is their capacity for a mild form of out-of-distribution generalisation. Just as well-designed conventional CNNs perform well when tested on translationally perturbed data, a network with  $G$ -convolutional layers and appropriate downsampling will theoretically perform well on  $G$ -perturbed data. As far as the authors are aware, the only place this problem has been studied is in [7], where spherical CNNs (in which  $G$  is the special orthogonal group) are trained on data on a sphere and tested on rotated data with excellent results. One of the primary reasons for this scarcity of study appears to be the lack of a sufficiently general theoretical framework. While  $G$ -convolutions can be efficiently implemented for finite groups [6] and a small family of continuous (Lie) groups [7, 5, 11], current techniques do not allow for the implementation of  $G$ -convolutions for Lie groups that are common in natural images, such as the affine and homography groups. Should this hurdle be overcome,  $G$ -convolutions could constitute an effective tool for common computer vision problems which circumvent the need for data augmentation and huge numbers of parameters.

**Contributions:** First, we introduce a rigorous mathematical framework using tools from topology to enable well-defined convolutional networks over *arbitrary* finite-dimensional Lie groups. Up until now, convolutional networks could only be implemented over finite groups or Lie groups with surjective exponential map, ruling out many natural warps. Second, we introduce tools from differential geometry and Lie theory to enable Markov Chain Monte-Carlo (MCMC) sampling from Haar measure for arbitrary Lie groups. This sampling may be done in a scalable and easily parallelisable fashion, permitting fast approximation of group convolutions. Sampling from Haar measure had previously not been possible beyond a small family of Lie groups [21, 11]. Third and finally, we illustrate the validity of our mathematical framework by testing it on the benchmark affine-invariant image classification task, where it outperforms the SOTA  $E(2)$ -equivariant CNN [31], and all but one of the existing CapsNet benchmarks. Finally, we demonstrate the generality of our theory by establishing a new benchmark on homography-invariant image classification, which to the best of our knowledge has not previously been studied. On this benchmark, our method exhibits stable performance, while that of the CapsNets for which we could obtain code [29, 19] degrades<sup>1</sup>.

<sup>1</sup>The code we used for our experiments is available at <https://github.com/lemaconnald/equivariant-convolutions>

**Acknowledgements:** We thank the reviewers for their substantive critiques, which have helped to improve the paper.

## 2. Related works

Group equivariant convolutional networks were first introduced in [6], which considered finite groups only. The theory and methods presented therein were subsequently refined in [8, 18], and generalised to various subgroups of the Euclidean (rigid motion) groups in [7, 8, 33, 34, 31]. Additional equivariance to scaling transformations was demonstrated in [15, 30], and equivariant attention using convolutions introduced in [26]. In [12], a method is given for the construction of multilayer perceptrons that are equivariant to any matrix group. This method uses (non-convolutional) finite-dimensional representations rather than the infinite-dimensional regular representation, and as demonstrated in [8, Section 4], there are situations where the latter has advantages in accuracy over the former. Prior to the present paper, the most general method in the literature which uses the infinite-dimensional regular representation of a group is given by Finzi et. al. [11]. The approach of Finzi et. al. applies to all Lie groups with surjective exponential map, and via this assumption parameterises filters on a (non-Euclidean) group as functions on the (Euclidean) Lie algebra. Of all of these works, our own method is most closely related to the latter. However, where as [11] applies only to Lie groups with surjective exponential map, ruling out for instance the affine and homography groups, our method applies to all finite-dimensional Lie groups.

Capsules were introduced in [16] as a way of organising neurons so as to better deal with expected variations in input such as position, scale, lighting and orientation. In [29], CapsNets were introduced as a way of integrating capsules into CNN architectures. It was shown that using a dynamic routing procedure enabled CapsNets to achieve 79% test accuracy on the affNIST<sup>2</sup> (affine-perturbed MNIST) test set, after being trained on unperturbed MNIST. This is a significant improvement on the test accuracy of 66% achieved by conventional CNNs. The task of training on unperturbed MNIST and testing on affNIST has since become a benchmark for affine-invariant image classification [29, 23, 19, 14, 25]. The present state of the art on affNIST is one of the models given in [25], which introduces a variational routing procedure to give a test performance of 97.69% on affNIST after training on MNIST. Our method outperforms all CapsNet benchmarks bar this state of the art, including the smaller of the models given in [25]. Precisely how CapsNets achieve such remarkable results, without admitting any theoretical guarantees of out-of-distribution generalisation, is not presently understood.

<sup>2</sup><http://www.cs.toronto.edu/tijmen/affNIST/>

### 3. Preliminaries

Recall that a *group* is a set  $G$  equipped with an associative multiplication  $G \times G \ni (u_1, u_2) \mapsto u_1 u_2 \in G$ , a multiplicative identity  $e$  such that  $ue = eu = u$  for all  $u \in G$ , and an inversion  $G \ni u \mapsto u^{-1} \in G$  for which  $uu^{-1} = u^{-1}u = e$  and such that  $(u_1 u_2)^{-1} = u_2^{-1} u_1^{-1}$  for all  $u, u_1, u_2 \in G$ . A group  $G$  is said to *act* on a set  $X$  if it admits a multiplication  $G \times X \ni (u, x) \mapsto u \cdot x \in X$  such that  $(u_1 u_2) \cdot x = u_1 \cdot (u_2 \cdot x)$  for all  $u_1, u_2 \in G, x \in X$ . If  $G$  acts on sets  $X$  and  $Y$ , with actions denoted by  $\cdot_X$  and  $\cdot_Y$  respectively, then  $\phi : X \rightarrow Y$  is said to be *equivariant* if  $\phi(g \cdot_X x) = g \cdot_Y \phi(x)$  for all  $g \in G$  and  $x \in X$ . An important special case of equivariance is *invariance*, which is when  $G$  acts trivially on  $Y$ ; in this case, one has  $\phi(g \cdot x) = \phi(x)$  for all  $g \in G$  and  $x \in X$ . Building a network that is robust to  $G$ -perturbations of its input amounts to building a network which is *invariant* to  $G$  in the above sense. It is easy to see that if  $\phi_1$  is equivariant and  $\phi_2$  is invariant, then  $\phi_2 \circ \phi_1$  is invariant – in this way, equivariance in intermediate layers of a network facilitates invariance of the network as a whole.

We also recall that a group  $G$  is a *Lie group* if it is simultaneously a smooth manifold with respect to which the multiplication and inversion are smooth maps. If  $G$  is a Lie group, there is a vector space  $\mathfrak{g}$  called its *Lie algebra*, equipped with an antisymmetric multiplication  $[\cdot, \cdot] : \mathfrak{g} \times \mathfrak{g} \rightarrow \mathfrak{g}$  called the *Lie bracket*, and an *exponential map*  $\exp : \mathfrak{g} \rightarrow G$ , which is one-to-one from a neighbourhood of zero in  $\mathfrak{g}$  onto a neighbourhood of the identity in  $G$ . We will be particularly concerned with *matrix groups*, which are groups of invertible matrices. Any matrix group is a Lie group, with multiplication and inversion given by the usual matrix operations, and with Lie algebra given by some space of not-necessarily-invertible matrices on which the Lie bracket is defined by the matrix commutator:

$$[A, B] := AB - BA.$$

For matrix groups, the exponential map is given by the matrix exponential. In what remains of this section, the reader may restrict to matrix groups, although the theory is valid in full generality.

Convolutions are defined in terms of integrals over Lie groups  $G$ . Recall that any Lie group  $G$  admits the *left* (resp. *right*) *Haar measure*  $\mu_L$  (resp.  $\mu_R$ ) [35, Section 1.3], with respect to which integrals may be performed. Left Haar measure is *left-invariant* in the sense that  $\int_G f(vu) d\mu_L(u) = \int_G f(u) d\mu_L(u)$  for any  $v \in G$  and any integrable function  $f$ . Similarly, right Haar measure is *right-invariant* in the sense that  $\int_G f(uv) d\mu_R(u) = \int_G f(u) d\mu_R(u)$  for all  $v \in G$  and integrable functions  $f$ . These measures are related by  $d\mu_L(u^{-1}) = d\mu_R(u)$ . On the zero-dimensional Lie group  $\mathbb{Z}^2$  of integer translations,

for instance, Haar measure is simply counting measure – the Haar measure of any subset of  $\mathbb{Z}^2$  is simply the number of points in that subset.

We denote by  $C(G; \mathbb{R}^K)$  the set of continuous functions  $G \rightarrow \mathbb{R}^K$  on a Lie group  $G$ . A function  $\psi \in C(G; \mathbb{R}^K)$  is said to be *compactly supported* if, roughly speaking, it is zero outside a set of finite Haar measure<sup>3</sup>. We denote by  $C_c(G; \mathbb{R}^K) \subset C(G; \mathbb{R}^K)$  the set of compactly supported continuous functions. A function  $\psi$  being compactly supported means in particular that the integrals  $\int_G \psi(u) d\mu_L(u)$  and  $\int_G \psi(u) d\mu_R(u)$  are finite. As a consequence, letting  $\cdot$  denote matrix multiplication, the *convolution*  $f * \psi$  of  $f \in C_c(G; \mathbb{R}^K)$  and  $\psi \in C_c(G; \mathbb{R}^{K \times L})$  defined by

$$f * \psi(u) = \int_G f(v) \cdot \psi(v^{-1}u) d\mu_L(v) \quad (1)$$

$$= \int_G f(uv^{-1}) \cdot \psi(v) d\mu_R(v) \quad (2)$$

is itself a compactly supported and continuous function [24, Proposition 1.1]. The convolution formula of Kondor-Trivedi [18], who work with *compact* groups, is a special case of this, as is the convolution formula for conventional CNNs

$$f * \psi(m, n) = \sum_{(i,j) \in \mathbb{Z}^2} f(i, j) \cdot \psi(m - i, n - j)$$

when  $G = \mathbb{Z}^2$ . For  $v \in G$  the map  $L_v : C(G; \mathbb{R}^K) \rightarrow C(G; \mathbb{R}^K)$  defined by  $L_v(f)(u) := f(v^{-1}u)$  preserves the subset  $C_c(G; \mathbb{R}^K)$ , and satisfies  $L_v(f) * \psi = L_v(f * \psi)$  for all  $f \in C(G; \mathbb{R}^K)$  and  $\psi \in C_c(G; \mathbb{R}^{K \times L})$ . That is, convolutions are *equivariant* to warps of feature maps  $f$  by warps in  $G$ . It is this equivariance that accounts for the robust performance of  $G$ -CNNs [6]. As we will describe in the next section, when working with the affine and homography groups we do not have the luxury of assuming our feature maps to be compactly supported – in fact, our feature maps must be allowed to be *unbounded* in that they may blow up to infinity away from the identity. One of our theoretical contributions is showing that convolution is still well-defined in such a case.

An important theorem from Lie group theory is the Schur-Poincaré formula for the derivative of the exponential map. This formula is the key to our theorem that enables sampling from Haar measure for arbitrary Lie groups.

**Theorem 3.1.** [28, Theorem 5, Section 1.2] *Let  $\mathfrak{g}$  be the Lie algebra of a finite-dimensional Lie group  $G$ , and let  $t \mapsto \xi(t)$  be a curve in  $\mathfrak{g}$ . Then one has*

$$\frac{d}{dt} \Big|_{t=0} \exp(X(t)) = dL_{\exp(X(0))} \frac{1 - e^{-ad_{X(0)}}}{ad_{X(0)}} \frac{d}{dt} \Big|_{t=0} X(t),$$

<sup>3</sup>Strictly speaking, a  $\psi$  being “compactly supported” means that the topological closure of the set on which  $\psi$  is nonzero is topologically compact. Since Haar measures are Radon measures [35, Section 1.3], compact sets do have finite measure, but the converse is not true in general.

where for  $X \in \mathfrak{g}$ ,  $ad_X : \mathfrak{g} \rightarrow \mathfrak{g}$  is the linear map defined by  $ad_X(Y) := [X, Y]$ ,  $dL_{\exp(X)}$  denotes the derivative of left-multiplication by  $\exp(X) \in G$ , and

$$\frac{1 - e^{-ad_X}}{ad_X} = \sum_{k=0}^{\infty} \frac{(-1)^k}{(k+1)!} (ad_X)^k$$

is a power series in the linear map  $ad_X$ . □

The Lie groups of experimental interest in this paper are the *affine group*  $Aff$  and the *homography group*  $Hom$ , both of which are ubiquitous in computer vision. The affine group  $Aff$ , generated by all shears, scalings, rotations and translations, is the set of all invertible  $3 \times 3$  matrices of the form

$$\begin{pmatrix} u_1 & u_2 & u_3 \\ u_4 & u_5 & u_6 \\ 0 & 0 & 1 \end{pmatrix}, \quad (3)$$

The Lie algebra  $\mathfrak{aff}$  of  $Aff$  is the space of  $3 \times 3$  matrices with bottom row equal to zero. We choose as an ordered basis the  $3 \times 3$  matrices given by the outer products  $\{e_1e_1^T, e_1e_2^T, e_1e_3^T, e_2e_1^T, e_2e_2^T, e_2e_3^T\}$ , where  $\{e_i\}_{i=1}^3$  is the canonical basis for  $\mathbb{R}^3$ . Taken in isolation, the first and fifth of these basis elements determine scaling, the second and fourth determine shearing, and the third and sixth determine translation, although due to the noncommutativity of the affine group this description fails for nontrivial linear combinations of the basis elements. The affine group *acts* on the space  $\mathbb{R}^2$  by the formula

$$\begin{pmatrix} u_1 & u_2 & u_3 \\ u_4 & u_5 & u_6 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} := \begin{pmatrix} u_1x_1 + u_2x_2 + u_3 \\ u_4x_1 + u_5x_2 + u_6 \end{pmatrix}. \quad (4)$$

The six-dimensional affine group may be identified as a subgroup of the eight-dimensional *homography group*  $Hom$ , which is generated by the perspective projections together with all affine transformations and may be identified with the group of  $3 \times 3$  matrices with determinant one [20]. The Lie algebra  $\mathfrak{hom}$  of the homography group is then the space of  $3 \times 3$  matrices with zero trace, and as an ordered basis we choose the matrices  $\{e_1e_1^T - (1/3)I, e_1e_2^T, e_1e_3^T, e_2e_1^T, e_2e_2^T - (1/3)I, e_2e_3^T, e_3e_1^T, e_3e_2^T\}$ . The linear map  $\phi : \mathfrak{aff} \rightarrow \mathfrak{hom}$  defined by sending the aforementioned basis of  $\mathfrak{aff}$  to the first six basis elements of  $\mathfrak{hom}$  is an inclusion of Lie algebras, so that the first six basis elements of  $\mathfrak{hom}$  can be interpreted as scaling, shearing and translational warps just as for the affine group, while the final two basis elements determine perspective projections. The corresponding inclusion  $\Phi : Aff \rightarrow Hom$  of Lie groups is given by the formula  $\Phi(A) := \det(A)^{-\frac{1}{3}}A$ , and one has  $\exp \circ \phi = \Phi \circ \exp$ .

Having described the necessary fundamentals on Lie groups and convolutions, in the next section we will de-

scribe our theoretical contributions which allow convolutions over arbitrary Lie groups to be built into neural networks. In the following section, we will empirically validate our theoretical framework using the affine and homography groups.

## 4. Theoretical framework

In this section we lay out our theoretical contributions. We also describe how they may be implemented.

### 4.1. The first layer

One key problem that must be overcome in using group convolutions for image classification is that the feature maps one is typically given as data are functions on  $\mathbb{Z}^2$  (giving intensity values to each pixel coordinate) rather than functions on a group  $G$ . One must therefore “lift” functions on  $\mathbb{Z}^2$  to functions on  $G$ . An elegant solution proposed by Cohen et. al. [6] is to use a modified convolutional layer. Suppose that  $G$  is a finite group acting on  $\mathbb{Z}^2$ , with the action of  $u \in G$  on  $(n, m) \in \mathbb{Z}^2$  denoted  $u(n, m)$ . Then given a feature map  $f \in C_c(\mathbb{Z}^2; \mathbb{R}^K)$  and filter  $\psi \in C_c(\mathbb{Z}^2; \mathbb{R}^{K \times L})$ , one defines  $f * \psi \in C_c(G; \mathbb{R}^L)$  by the formula

$$f * \psi(u) := \sum_{(n,m) \in \mathbb{Z}^2} f(n, m) \cdot \psi(u^{-1}(n, m)). \quad (5)$$

It is proved in [6, Equation 12] that Equation (5) is equivariant in the sense that  $L_v(f) * \psi = L_v(f * \psi)$ , where  $L_v(f)(n, m) := f(v^{-1}(n, m))$ . Thus one is able to convert feature maps on  $\mathbb{Z}^2$  into feature maps on  $G$  in an equivariant fashion, and these can then be fed into higher convolutional layers given by Equations (1) and (2).

Equation (5) was also identified by mathematicians at around the same time as a kind of generalised Hilbert space inner product [17]. Note that  $f * \psi$  is compactly supported by virtue of the fact that the action of  $\mathbb{Z}^2$  on itself is a proper<sup>4</sup> action [17]. The actions of the affine and homography groups on the plane are *not* proper, and as we will soon see this means our theory for higher hidden layers needs to be sufficiently flexible to allow for *unbounded* feature maps, which have not previously been considered in the literature.

Supposing now that  $G$  is a Lie group acting on  $\mathbb{R}^2$ , Equation (5) ceases to work for two primary reasons. Firstly,  $G$  need not map  $\mathbb{Z}^2$  to itself (consider scaling by positive real numbers). This can be overcome by replacing  $\mathbb{Z}^2$  by  $\mathbb{R}^2$ , and defining  $\psi$  by an MLP as in [36]. The sum of Equation (5) can then be thought of as an approximation to the integral

$$f * \psi(u) := \int_{\mathbb{R}^2} f(x) \cdot \psi(u^{-1}x) dx \quad (6)$$

<sup>4</sup>An action of a topological group  $G$  on a topological space  $X$  is *proper* if for any compact set  $K$  in  $X \times X$ , the preimage of  $K$  under the map  $G \times X \ni (u, x) \mapsto (u \cdot x, x) \in X \times X$  is compact

over  $\mathbb{R}^2$ . There is, however, a deeper problem: namely that Equation (6) is *not equivariant* under any transformations which distort the volume of Euclidean space.

**Theorem 4.1.** *For  $v \in G$  and  $x \in \mathbb{R}^2$ , let  $Dv(x)$  denote the Jacobian at  $x$  of the map  $\mathbb{R}^2 \rightarrow \mathbb{R}^2$  defined by  $v$ . Then one has  $L_v(f) * \psi = L_v(f * \psi)$  for all  $f \in C_c(\mathbb{R}^2; \mathbb{R}^K)$  and  $\psi \in C_c(\mathbb{R}^2; \mathbb{R}^{K \times L})$  if and only if  $\det(Dv(x)) = 1$  for all  $x \in \mathbb{R}^2$ .*

*Proof.* Fixing  $v, f$  and  $\psi$ , one computes

$$\begin{aligned} L_v(f) * \psi(u) &= \int_{\mathbb{R}^2} f(v^{-1}x) \cdot \psi(u^{-1}x) dx \\ &= \int_{\mathbb{R}^2} f(\tilde{x}) \cdot \psi(u^{-1}v\tilde{x}) d(v\tilde{x}) \\ &= \int_{\mathbb{R}^2} f(\tilde{x}) \cdot \psi((v^{-1}u)\tilde{x}) \det(Dv(\tilde{x})) d\tilde{x}, \end{aligned}$$

using the substitution  $x = v\tilde{x}$  for the second line and the usual change of coordinate formula  $d(\varphi(x)) = \det(D\varphi(x))dx$  for the third. The result follows.  $\square$

Since the affine and homographic transformations we wish to consider for applications to computer vision need not preserve the Euclidean volume element in general, Theorem 4.1 obstructs the naive use of Equation (6). We propose instead the following modification, which nonetheless reduces to Equation (6) for volume-preserving warps. Similar formulae appear in less general settings in [15, 30].

**Theorem 4.2.** *Define the  $G$ -convolution of  $f \in C_c(\mathbb{R}^2; \mathbb{R}^K)$  and  $\psi \in C_c(\mathbb{R}^2; \mathbb{R}^{K \times L})$  by*

$$f * \psi(u) := \int_{\mathbb{R}^2} f(x) \cdot \psi(u^{-1}x) \det(Du^{-1}(x)) dx. \quad (7)$$

*Then  $f * \psi \in C(G; \mathbb{R}^L)$ , and for all  $v \in G$  one has  $L_v(f) * \psi = L_v(f * \psi)$ .*

*Proof.* We defer a proof of the fact that  $f * \psi$  is a continuous function to the supplementary material, and here prove only equivariance. Fixing  $v, f$  and  $\psi$  we see that  $L_v(f) * \psi(u)$  is given by

$$\begin{aligned} &\int_{\mathbb{R}^2} f(v^{-1}x) \cdot \psi(u^{-1}x) \det(Du^{-1}(x)) dx \\ &= \int_{\mathbb{R}^2} f(\tilde{x}) \cdot \psi(u^{-1}v\tilde{x}) \det(Du^{-1}(v\tilde{x})) \det(Dv(\tilde{x})) d\tilde{x} \\ &= \int_{\mathbb{R}^2} f(\tilde{x}) \cdot \psi((v^{-1}u)^{-1}\tilde{x}) \det(D(u^{-1}v)(\tilde{x})) d\tilde{x} \\ &= \int_{\mathbb{R}^2} f(\tilde{x}) \cdot \psi((v^{-1}u)^{-1}\tilde{x}) \det(D(v^{-1}u)^{-1}(\tilde{x})) d\tilde{x} \\ &= L_v(f * \psi)(u), \end{aligned}$$

where for the second line we have made the substitution  $x = v\tilde{x}$ , and for the third we have applied the chain rule.  $\square$

In practice, we approximate Equation (7) by a sum over the discrete integer lattice in  $\mathbb{R}^2$ . Interpolation of the image  $f$ , as is required in [15], is therefore unnecessary. We note that the “lifting” of a feature pair  $(x, f(x))$  implicit in Equation (7) to a new pair  $(u, (f * \psi)(u))$  is achieved via an entirely different means in [11], where instead a pair  $(x, f(x))$  is lifted to a triplet  $(u, q, f(x))$ , with  $u$  a group element and  $q$  is an element of the orbit of  $G$  through  $x$ . Our lifting layer thus has the advantage of being not only trainable so that it may extract features, but also of being more memory-efficient, producing only a doublet rather than a triplet after its forward pass.

Finally, notice that in Theorem 4.2, even though  $f$  and  $\psi$  are compactly supported, their convolution need not be since the action of  $G$  need not be proper (see footnote 2). The presence of the  $\det(Du^{-1}(x))$  in Equation (7) moreover means that  $f * \psi$  need not even be bounded (consider for instance the group of scaling transformations). We show in the next section that these technicalities do not prevent group convolutions in higher layers.

## 4.2. Hidden convolutional layers

Having lifted an image to a continuous function on  $G$  using the first layer detailed in the previous subsection, we must now implement convolutions over  $G$ . However, as alluded to in the previous section, the feature maps produced by Theorem 4.2 may be unbounded, and the ordinary theory [6, 18, 24] therefore does not apply. Showing that convolutions are possible even in the unbounded case is our second theoretical contribution; we defer a proof to the supplementary material.

**Theorem 4.3.** *Define the convolution of a feature map  $f \in C(G; \mathbb{R}^K)$  with a filter map  $\psi \in C_c(G; \mathbb{R}^{K \times L})$  by the formula*

$$f * \psi(u) := \int_G f(v) \cdot \psi(v^{-1}u) d\mu_L(v) \quad (8)$$

$$= \int_G f(uv^{-1}) \cdot \psi(v) d\mu_R(v), \quad (9)$$

*where  $\cdot$  denotes matrix multiplication. Then  $f * \psi$  is a continuous function on  $G$ , and for all  $v \in G$  one has  $L_v(f) * \psi = L_v(f * \psi)$ .  $\square$*

The next task is to describe how convolutions may be implemented. In [6], this is achieved by assuming  $G$  to be a discrete group, so that the values of any feature map at each group element can be encoded in channels of a corresponding feature tensor. While efficient computationally, this approach is insufficient for a continuous Lie group such as the affine or homography group.

In [11], Equation (8) is used in approximating convolutions using a Monte-Carlo method. Importantly, the method adopted in [11] makes the following two assumptions.

- **Assumption 1 of [11]:** It is assumed that  $G$  has a surjective exponential map, so that filters can be parameterised by multi-layer perceptrons (MLPs), which take *vectors* (that is, Lie algebra elements) as input.
- **Assumption 2 of [11]:** It is assumed that Haar measure on  $G$  is easily reducible to known measures on known sets (such as uniform measures on Euclidean space or the unit quaternions). Such measures are easily sampled from so that a Monte-Carlo estimate of Equation (8) can be obtained.

For the affine and homography groups, neither Assumption 1 nor Assumption 2 holds. Our next theorem allows us to overcome these assumptions entirely.

**Theorem 4.4.** *If  $f$  is an integrable function on  $G$  which is zero outside of a sufficiently small neighbourhood of the identity, then Haar measure  $d\mu_R$  can always be chosen such that*

$$\int_G f(u) d\mu_R(u) = \int_{\mathfrak{g}} f(\exp(\xi)) \det\left(\frac{1 - e^{-ad_{-\xi}}}{ad_{-\xi}}\right) d\xi, \quad (10)$$

where  $d\xi$  denotes a Euclidean volume element in the vector space  $\mathfrak{g}$ , and  $ad_{\xi} : \mathfrak{g} \rightarrow \mathfrak{g}$  and  $(1 - e^{-ad_{\xi}})/ad_{\xi}$  are given in Theorem 3.1.

A rigorous proof of the theorem requires some differential geometry (differential forms, their integrals and their pullbacks), which we defer to the supplementary material. For now we just give an idea of the proof.

*Idea of proof.* The small support of  $f$  enables one to use a change of coordinates  $u = \exp(\xi)$ , reducing the integral over  $G$  to an integral over  $\mathfrak{g}$ . The change of coordinates is implemented by the exponential map, and therefore the determinant of its derivative (Theorem 3.1) appears as a multiplicative factor to keep track of the way volumes are distorted by this change of coordinates. This explains the presence of the power series  $(1 - e^{-ad_{-\xi}})/ad_{-\xi}$  in the new expression. That multiplication by  $\exp(\xi)$  does not appear in this new expression, despite its appearance in Theorem 3.1, follows from the invariance of Haar measure.  $\square$

We now combine Theorem 4.4 with Equation (9), and demonstrate how they allow us to circumvent the assumptions made in [11]. Fix a feature map  $f \in C(G; \mathbb{R}^K)$ , and compactly supported filter  $\psi \in C_c(G; \mathbb{R}^{K \times L})$ . Assume  $\psi$  to have support in a sufficiently small neighbourhood of the identity that  $\tilde{\psi} := \psi \circ \exp^{-1}$  is well-defined. Then the convolution  $f * \psi$  is given by

$$\int_{\mathfrak{g}} f(u \exp(-\xi)) \cdot \tilde{\psi}(\xi) \det\left(\frac{1 - e^{-ad_{-\xi}}}{ad_{-\xi}}\right) d\xi. \quad (11)$$

Let us now explain how Equation (11) allows us to overcome Assumptions 1 and 2.

- **Overcoming assumption 1 of [11]:** The replacement of  $\psi$  with  $\tilde{\psi}$  in Equation (11) is key for implementation, as it allows trainable filters to be parameterised by small neural networks which take vectors (Lie algebra elements) as input rather than group elements. Equation (11) holds for *any Lie group*. In contrast, Equation (8) is used for the convolution  $f * \psi$  in [11], meaning the filter  $\psi$  needs to take group elements  $v^{-1}u$  as input. Since these group elements may be arbitrarily far from the identity, surjectivity of the exponential map must be assumed in order to write  $\psi = \tilde{\psi} \circ \exp^{-1}$ .
- **Overcoming assumption 2 of [11]:** There is a large family of Markov chain Monte-Carlo (MCMC) methods for sampling from distributions (measures) on Euclidean spaces [1]. Such methods do not, however, immediately apply to non-Euclidean spaces such as Lie groups. In [11], sampling from Haar measure on a Lie group is instead achieved by identifying Haar measure with more well-known measures which are easily sampled from. In contrast, using Equation (11) allows us to use MCMC methods to sample from Haar measure for *any* Lie group. Indeed, the Lie algebra of any Lie group is a Euclidean space, and the function

$$\xi \mapsto \det\left(\frac{1 - e^{-ad_{-\xi}}}{ad_{-\xi}}\right)$$

thereon can then be thought of as (being proportional to) a density function. With this density function in hand, standard MCMC methods such as the Metropolis algorithm may be utilised in order to sample from Haar measure to produce a Monte-Carlo estimate

$$f * \psi(u) \approx \frac{1}{N} \sum_{\xi_i \sim \exp^* \mu_R} f(u \exp(-\xi_i)) \cdot \tilde{\psi}(\xi_i)$$

of the convolution  $f * \psi$ .

We remark that MCMC methods like the Metropolis algorithm are easily parallelised [27, 4]. For the fast estimation of convolutions, we parallelise the Metropolis algorithm to produce multiple samples from Haar measure at once. We achieve this by running multiple chains at once, using boolean matrices to keep track of which samples are accepted and which are rejected. This parallelisation results in an order of magnitude greater speed than is obtained via the conventional method of producing samples sequentially.

### 4.3. Nonlinearities and fully connected hidden layers

It was observed in [6], and is easily verified, that if  $f : G \rightarrow \mathbb{R}^K$  is any continuous function, then postcomposition with any function  $\phi : \mathbb{R}^K \rightarrow \mathbb{R}^L$  is an equivariant

operation:

$$\phi \circ L_v(f) = L_v(\phi \circ f), \quad v \in G.$$

Therefore conventional nonlinearities may be applied in the usual way after any convolutional layer without breaking equivariance, and fully connected layers can be stacked on top of convolutional layers while preserving equivariance.

#### 4.4. Final layer

The final layer is a pooling layer. Since our feature maps are not generally bounded, *global* pooling need not be well-defined. However, our feature maps are continuous, meaning their maxima over any compact set are well-defined. Assuming a  $C$ -class classification problem, given a continuous function  $f \in C(G; \mathbb{R}^K)$ , a matrix  $A \in \mathbb{R}^{K \times C}$  and a bias vector  $b \in \mathbb{R}^C$ , the componentwise operation

$$f \mapsto \max_{u \in K} (Af(u) + b)$$

is well-defined for any compact subset  $K$  of  $G$ . Our next and final theorem, proved in the supplementary material, gives a theoretical guarantee of invariance of this operation.

**Theorem 4.5.** *Let  $f : G \rightarrow \mathbb{R}$  be a (possibly unbounded) continuous function, and  $K \subset G$  a compact neighbourhood of the identity. If  $f$  takes its maximum value over  $K$  only at points in the interior of  $K$ , then there is an open neighbourhood  $V$  of the identity for which  $\max_{u \in K} L_v(f)(u) = \max_{u \in K} f(u)$  for all  $v \in V$ .  $\square$*

## 5. Experiments

Our framework enables  $G$ -invariant models, which we test by training on MNIST and testing them on  $G$ -perturbed MNIST. We take  $G$  to be the affine and homography groups. Testing on affine-perturbed MNIST (affNIST) after training on MNIST is the benchmark in affine-invariant image classification [29, 23, 19, 14], while the analogous test for homography-invariant image classification has not yet been studied. To test homography-invariance, we created a *homNIST*<sup>5</sup> test set by sampling 32 homographies from right Haar measure for each of the MNIST test images.

In both cases, our networks consist of a lifting convolutional layer (Subsection 4.1), a single  $G$ -convolutional layer (Subsection 4.2), two residual fully connected layers and finally a pooling layer (Subsection 4.4). In each forward pass, a new collection of pooling samples is drawn from  $G$  according to a Gaussian distribution centered at the identity, and a new collection of samples for estimating the convolutional integral is drawn from right Haar measure using the Metropolis MCMC algorithm. Models were trained for 150 epochs with a batch size of 60, using Adam with the

<sup>5</sup><https://www.kaggle.com/datasets/lachlanmacdonald/homnist>

Model	affNIST Test Acc.	Parameters
RU CapsNet [25]	97.69	>580K
<b>affConv (ours)</b>	<b>95.08(±0.31)</b>	<b>373K</b>
affine CapsNet [14]	93.21	
GE CapsNet [19]	89.10	235K
CapsNet [29]	79	8.1M
CNN [29]	66	35.4M
E2SFCNN [31]	57.10(± 0.71)	7.07M

Model	homNIST Test Acc.	Parameters
<b>homConv (ours)</b>	<b>95.71(±0.09)</b>	<b>376K</b>
GE CapsNet [19]	84.67	235K
E2SFCNN [31]	82.53(± 0.30)	7.07M
CapsNet [29]	74.95	8.1M

Figure 2: Test performance comparison on affNIST and homNIST after training on MNIST.

recommended default settings [2]. The tables in Figure 2 benchmark our models’ mean test performances, taken at the final epoch, over five training runs from different initialisations, with standard errors. We were unfortunately unable to locate code or parameter counts for the affine CapsNet of [14], and those of [25] have been estimated from a smaller model in [25] for which parameter counts are given. We predict that other CapsNets would exhibit similar performance drops on homNIST relative to affNIST as the other CapsNets. Curiously, the  $E(2)$ -equivariant CNN of [31] (which we trained using the same regime as the authors, albeit on MNIST only) achieves worse performance on homNIST than on affNIST: we expect this is due to homNIST exhibiting smaller affine perturbations than affNIST, which was necessary to rule out edge effects<sup>6</sup>.

We predict performance to degrade as our approximations to idealised convolutions and max pools get worse. To test this prediction, we train our models using the same initialisation and different numbers  $N$  of samples to approximate convolution and pooling, and evaluate test performance against  $N$ . Figure 3 shows degradation of all test performances as the number of samples is decreased from  $N = 100$  to  $N = 25$ , until catastrophic failure at  $N = 1$ .

We tested the equivariance error of our layers, measured in the same manner as in [30, Section 6.1]. The results for the first layer are shown in Figure 4. Interestingly, the equivariance error of higher convolutional layers is *always zero*, regardless of the number of samples used to approximate the convolutions. This can be seen mathematically: let  $V := \{v_i\}_{i=1}^N$  be any finite sample from Haar measure on a group  $G$ , let  $f \in C(G)$  be a feature map, with  $\psi \in C_c(G)$

<sup>6</sup>Quoted figures are means with standard errors over four training runs with different random seeds

Model	Data	Num. pool samples				
		100	75	50	25	1
affConv	MNIST	98.6	98.5	98.4	97.9	68.1
	affNIST	94.6	94.7	93.6	91.5	51.1
homConv	MNIST	98.7	98.7	98.4	98.5	78.6
	homNIST	96.0	95.1	95.4	94.3	66.2

(a) Test accuracy versus number of pooling samples

Model	Data	Num. conv. samples				
		100	75	50	25	1
affConv	MNIST	98.3	98.2	98.1	96.3	36.0
	affNIST	94.1	93.3	92.5	88.0	30.0
homConv	MNIST	98.9	98.7	98.4	97.2	41.0
	homNIST	95.7	95.7	94.7	92.2	36.3

(b) Test accuracy versus number of convolutional samples

Figure 3: Test accuracies are relatively stable for smaller numbers of samples until catastrophic failure with only one sample. This failure is much more pronounced when convolutional samples are decreased than when pooling samples are decreased. Each test value is from a single training run.

Model	Equivariance error
affConv	5.52%
homConv	6.86%

Figure 4: Equivariance errors  $\|L_u(f * \psi)(v) - L_u(f) * \psi(v)\|_2^2 / \|L_u(f * \psi)(v)\|_2^2$  of the first layer, averaged over 100 sample warps  $u$  and 100 group elements  $v$ .

a filter map. Define  $f *_{\mathcal{V}} \psi \in C(G)$  by the formula

$$f *_{\mathcal{V}} \psi(u) := \frac{1}{N} \sum_{i=1}^N f(uv_i^{-1})\psi(v_i).$$

Monte-Carlo theory tells us that as the size of the sample  $V$  grows, the error in approximating the true, equivariant convolution  $f * \psi$  by the Monte-Carlo estimate  $f *_{\mathcal{V}} \psi$  shrinks. However, for any  $w \in G$ , one has perfect equivariance

$$\begin{aligned} L_w(f) *_{\mathcal{V}} \psi(u) &= \sum_{i=1}^N L_w(f)(uv_i^{-1})\psi(v_i) \\ &= \sum_{i=1}^N f(w^{-1}uv_i^{-1})\psi(v_i) \\ &= L_w(f *_{\mathcal{V}} \psi)(u) \end{aligned}$$

regardless of the size of  $V$ . This analysis suggests the degradation in Figure 3 is a consequence of the higher variance of the forward pass arising from the paucity of samples, resulting in a lack of trainability, rather than being a result of a lack of equivariance.

## 6. Discussion

**On theoretical guarantees:** Despite their desirability for building trustworthy AI, theoretical guarantees of performance are still uncommon in modern machine learning. In addition to the greater generality of our proposed framework, one advantage of our approach to invariance over conventional CapsNets is the availability of rigorous mathematical guarantees of invariance of *idealised models*. In practice these guarantees do not apply exactly due to the approximate methods used in implementation. Providing rigorous mathematical guarantees for *practical* models, which take into account the stochastic methods used to approximate continuous integrals and max pools, is desirable.

**On depth:** One limitation of the framework we propose is that naively adding more convolutional layers results in a corresponding increase in memory usage beyond that required to store the additional weights. If nonlinearities are not used between convolutional layers, this increase in memory usage can be made *linear* in the number of layers using well-known path-integral methods [10], however this sacrifices expressivity. In a naive implementation, keeping nonlinearities between convolutional layers implies an exponential growth in memory usage with the number of convolutional layers – with  $L$  convolutional layers, and  $N$  samples used to approximate each convolution, one needs  $N^L$  function evaluations in total<sup>7</sup>. Practically, this means very large memory usage at the front end of the forward pass. However, recent work on nonlinearities in Monte-Carlo methods [10] together with our ablation (Figure 3) suggest that this memory usage can be made vastly less while retaining good performance. Moreover, our empirical and theoretical observation in the Experiments section that higher convolutional layers are always perfectly equivariant, regardless of convolutional sample size, holds promise for use in more scalable architectures.

## 7. Conclusion

We introduced a mathematical framework for convolutional neural networks over *any* Lie group  $G$ . We have given rigorous guarantees of equi/invariance of idealised models, and a new, parallelisable method for sampling from Haar measure for arbitrary Lie groups. We have empirically validated our framework by using to achieve competitive results on benchmark invariance tasks in image classification.

<sup>7</sup>Combinatorial explosion is also present in [11, Section 4.4], where for each point  $u \in G$ , a *new* family of samples  $\{v_i\} \in \text{nbhd}(u)$ , dependent on  $u$ , must be drawn from Haar measure to estimate the convolution  $f * \psi(u)$ . Thus for instance to estimate  $f * \psi_1 * \psi_2(u)$  naively, one requires samples  $\{v_i\} \in \text{nbhd}(u)$  for the outermost integral and, for each  $i$ , new samples  $\{w_{i,j}\} \in \text{nbhd}(v_i)$  dependent on  $v_i$  for the innermost integral. In practice, the authors of [11] avoid this combinatorial explosion by using the *same* samples for each layer, however they provide no guarantee that this sampling method yields an unbiased estimate of the true convolution.



## References

- [1] *Handbook of Markov chain Monte Carlo*. Chapman and Hall/CRC handbooks of modern statistical methods. CRC Press, 2011. 6
- [2] Adam: A Method for Stochastic Optimization. *Int. Conf. Learn. Represent.*, 2015. 7
- [3] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics and Gauges. *arXiv preprint: arXiv:2104.13478*, 2021. 1, 2
- [4] Ben Calderhead. A general construction for parallelizing Metropolis Hastings algorithms. *PNAS*, 111:17408–17403, 2014. 6
- [5] Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral CNN. *Int. Conf. Machine Learning*, 97:1321–1330, 2019. 1, 2
- [6] Taco Cohen and Max Welling. Group equivariant convolutional networks. *Int. Conf. Machine Learning*, 48:2990–2999, 2016. 1, 2, 3, 4, 5, 6
- [7] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *Int. Conf. Learn. Represent.*, 2018. 2
- [8] Taco S. Cohen and Max Welling. Steerable cnns. *Int. Conf. Learn. Represent.*, 2017. 1, 2
- [9] Alain Connes. *Noncommutative Geometry*. Academic Press, 1994. 1
- [10] Jérémie Dauchet, Jean-Jacques Beziau, Stéphane Blanco, Cyril Caliot, Julien Charon, Christophe Coustet, Mouna El Hafí, Vincent Eymet, Olivier Farges, Vincent Forest, Richard Fournier, Mathieu Galtier, Jacques Gautrais, Anaïs Khuong, Lionel Pelissier, Benjamin Piau, Maxime Roger, Guillaume Terrée, and Sebastian Weitz. Addressing nonlinearities in Monte Carlo. *Sci. Rep.*, 8(1), 2018. 8
- [11] Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. *Int. Conf. Machine Learning*, 119:3165–3176, 2020. 1, 2, 5, 6, 8
- [12] Marc Finzi, Max Welling, and Andrew Gordon Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. *ICML*, 2021. 2
- [13] Geoffrey E. Hinton and Sara Sarbour and Nicholas Frosst. Matrix capsules with EM routing. *Int. Conf. Learn. Represent.*, 2018. 1
- [14] Jindong Gu and Volker Tresp. Improving the robustness of capsule networks to image affine transformations. *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7283–7291, 2020. 2, 7
- [15] João F. Henriques and Andrea Vedaldi. Warped convolutions: Efficient invariance to spatial transformations. *Int. Conf. Machine Learning*, 70:1461–1469, 2017. 2, 5
- [16] Geoffrey E. Hinton, Alex Krizhevsky, and Sida D. Wang. Transforming auto-encoders. In *Artificial Neural Networks and Machine Learning - ICANN 2011*, volume 6791 of *Lecture Notes in Computer Science*, pages 44–51. Springer, 2011. 1, 2
- [17] Rohit Holkar. Topological construction of  $C^*$ -correspondences for groupoid  $C^*$ -algebras. *Journal of Operator Theory*, 77, 2017. 1, 4
- [18] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. *Int. Conf. Machine Learning*, 80:2752–2760, 2018. 2, 3, 5
- [19] Jan Eric Lenssen, Matthias Fey, and Pascal Libuschewski. Group equivariant capsule networks. *Adv. Neural Inform. Process. Syst.*, pages 8858–8867, 2018. 1, 2, 7
- [20] Ninad Manerikar, Minh-Duc Hua, and Tarek Hamel. Homography observer design on special linear group  $sl(3)$  with application to optical flow estimation. In *2018 European Control Conference (ECC)*, pages 1–5, 2018. 4
- [21] Francesco Mezzadri. How to generate random matrices from the classical compact groups. *Notices of the American Mathematical Society*, 54:592–604, 2007. 2
- [22] Sameera Ramasinghe, Salman Khan, Nick Barnes, and Stephen Gould. Representation Learning on Unit Ball with 3D Roto-Translational Equivariance. *Int. J. Comput. Vis.*, 128, 1612–1634. 1, 2
- [23] David Rawlinson, Abdelrahman Ahmed, and Gideon Kowadlo. Sparse unsupervised capsules generalize better. *arXiv preprint arXiv:1804.06094*, 2018. 2, 7
- [24] Jean Renault. *A Groupoid Approach to  $C^*$ -algebras*. Springer-Verlag, 1980. 3, 5
- [25] Fabio De Sousa Ribeiro, Georgios Leontidis, and Stefanos Kollias. Introducing routing uncertainty in capsule networks. *NeurIPS*, 2020. 2, 7
- [26] David Romero, Erik Bekkers, Jakub Tomczak, and Mark Hoogendoorn. Attentive group equivariant convolutional networks. *Int. Conf. Machine Learning*, 119:8188–8199, 2020. 2
- [27] Jeffrey S. Rosenthal. Parallel Computing and Monte Carlo Algorithms. *Far East Journal of Theoretical Statistics*, 4:207–236, 2000. 6
- [28] Wulf Rossmann. *Lie Groups: An Introduction Through Linear Groups*. Oxford University Press, 2002. 3
- [29] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. *Adv. Neural Inform. Process. Syst.*, pages 3856–3866, 2017. 1, 2, 7
- [30] Ivan Sosnovik, Michal Szmaja, and Arnold W. M. Smeulders. Scale-equivariant steerable networks. *ICLR*, 2020. 2, 5, 7
- [31] Maurice Weiler and Gabriele Cesa. General  $e(2)$ -equivariant steerable cnns. *Adv. Neural Inform. Process. Syst.*, pages 14334–14345, 2019. 1, 2, 7
- [32] Maurice Weiler, Patrick Forré, Erik Verlinde, and Max Welling. Coordinate Independent Convolutional Networks – Isometry and Gauge Equivariant Convolutions on Riemannian Manifolds. *arXiv preprint: arXiv:2106.06020*, 2021. 1, 2
- [33] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. *Adv. Neural Inform. Process. Syst.*, pages 10402–10413, 2018. 1, 2

- [34] Maurice Weiler, Fred A. Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant cnns. *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 849–858, 2018. [1](#), [2](#)
- [35] Dana Williams. *Crossed Products of  $C^*$ -Algebras*, volume 134 of *Mathematical Surveys and Monographs*. American Mathematical Society, 2007. [3](#)
- [36] Wenxuan Wu, Zhongang Qi, and Fuxin Li. Pointconv: Deep convolutional networks on 3d point clouds. *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9621–9630, 2019. [4](#)
- [37] Richard Zhang. Making convolutional networks shift-invariant again. *Int. Conf. Machine Learning*, 97:7324–7334, 2019. [1](#)