LEARNING EQUIVARIANCES AND PARTIAL EQUIVARIANCES FROM DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Group equivariant Convolutional Neural Networks (G-CNNs) constrain features to respect the chosen symmetries, and lead to better generalization when these symmetries appear in the data. However, if the chosen symmetries are not present, group equivariant architectures lead to overly constrained models and worse performance. Frequently, the distribution of the data can be better represented by a subset of a group than by the group as a whole, e.g., rotations in $[-90°, 90°]$. In such cases, a model that respects equivariance *partially* is better suited to represent the data. Moreover, relevant symmetries may differ for low and high-level features, e.g., edge orientations in a face, and face poses relative to the camera. As a result, the optimal level of equivariance may differ per layer. In this work, we introduce *Partial G-CNNs*: a family of equivariant networks able to learn partial and full equivariances from data at every layer end-to-end. Partial G-CNNs retain full equivariance whenever beneficial, e.g., for rotated MNIST, but are able to restrict it whenever it becomes harmful, e.g., for 6 / 9 or natural image classification. Partial G-CNNs perform on par with G-CNNs when full equivariance is necessary, and outperform them otherwise. Our method is applicable to discrete groups, continuous groups and combinations thereof.

1 INTRODUCTION

The translation equivariance of Convolutional Neural Networks (CNNs) (LeCun et al., 1998) has proven to be an important inductive bias for good generalization on vision tasks. This is achieved by restricting learned representations to respect the translation symmetry observed in visual data, such that if an input is translated, its feature representation is also translated, but not modified in any other way. Group equivariant Convolutional Neural Networks (G-CNNs) (Cohen & Welling, 2016) extend equivariance to other symmetry groups. Analogously, they restrict the learned representations to respect the symmetries in the group considered such that if an input is transformed by an element in the group, e.g., a rotation, its feature representation is also transformed, e.g., rotated, but not modified.

However, the group to which G-CNNs are equivariant must be fixed prior to training, and imposing equivariance to a group larger than the symmetries present in the data leads to over-constraining and worse performance (Chen et al., 2020). The latter results from a difference in the distribution of the data, and the family of distributions the model can describe. Consequently, the group must be selected carefully, and it should correspond to the transformations that appear naturally in the data.

Interestingly, the distribution of the data can frequently be better represented by a subset of the group than by the group itself, e.g., rotations in $[-90°, 90°]$. For instance, natural images are much more likely to show an elephant standing straight or slightly rotated than an elephant upside-down. In some cases, group transformations can even change the desired model response. A typical example is the classification of the numbers 6 and 9, whose defining factor is their pose. In both examples, the distribution of the data is better represented by a model that respects rotation equivariance *partially*. That is, a model equivariant to some, but not all rotations.

In addition, the optimal family of full / partial equivariances may change per layer. This results from changes in pose likelihoods for low and high-level features in the data, e.g., the orientations of edges in an human face, and the pose of human faces relative to the camera. Whereas detecting a face may benefit from partial rotation equivariance, detecting edges benefits from full rotation equivariance.

These observations indicate that constructing a model able to define different ranges of partial / full equivariances at each layer may be advantageous. Yet, manually defining the level of partial / full equivariance a model must have at every layer is at best a difficult and time-consuming task.

In this work, we introduce *Partial Group equivariant Convolutional Neural Networks*: a new family of equivariant methods able to learn partial and full equivariances directly from data at every layer. This is achieved by defining a learnable probability distribution over group elements at each group convolutional layer in the network. Instead of sampling group elements uniformly from the group during the group convolution – as normally done –, Partial G-CNNs sample them from the learned distribution. This allows them to adjust their level of equivariance during training. We note that our partial equivariance framework allows for both equivariance forgetting, and full equivariance. The former is obtained by collapsing the distribution along a certain group dimension, e.g., the rotation axis, and the later results from keeping the learned distribution uniform over the entire group.

We evaluate Partial G-CNNs on illustrative toy tasks and on challenging benchmark datasets. We show that whenever full equivariance is beneficial, e.g., for rotated MNIST, Partial G-CNNs remain fully equivariant. However, whenever harmful, our models restrict equivariance to a subset of the group transformations, e.g., to make 6 / 9 classification possible or to improve on natural image classification. Our method improves upon conventional group equivariant networks when equivariance reductions are advantageous, and matches their performance whenever their design is optimal. To the best of our knowledge, this work is the first approach to explore partial group equivariances, and to provide a practical implementation to learn them directly from data in an end-to-end manner.

In summary, our **contributions** are:

- We present a novel architectural design for equivariant networks, with which group convolutional networks are able to learn layer-wise partial or full equivariances from data.
- We empirically show that partial equivariance allows our networks to improve upon conventional G-CNNs for tasks for which full equivariance is harmful. However, whenever beneficial, our models learn to stay fully equivariant and match the performance of G-CNNs.

2 Related Work

Equivariance learning. Learning equivariant mappings from data has been explored by (*i*) meta-learning as a means to iterate between the learning of free weights and a weight-tying matrix that encodes the symmetry equivariances of the network (Zhou et al., 2020), or by (*ii*) learning the Lie algebra generators of the group jointly with the parameters of the network (Dehmamy et al., 2021). However, both approaches utilize the same learned symmetries across layers. In addition, MSR (Zhou et al., 2020) is only applicable to (small) discrete groups, and requires longer training times. L-Conv (Dehmamy et al., 2021), on the other hand, is only applicable to continuous groups and is not fully compatible with current deep learning components, e.g., pooling, normalization.

Contrary to these approaches, our method learns different levels of full/partial equivariance at every layer, is fully compatible with current deep learning components, and is applicable for discrete groups, continuous groups and combinations thereof. It is important to mention, however, that Zhou et al. (2020); Dehmamy et al. (2021) aim to learn the structure of the group from scratch. Our method, on the other hand, starts from a (very) large group and allows layers in the network to constrain their equivariance levels whenever necessary in order to fit the distribution of the data better.

Invariance learning. Learning the right amount of invariance from data has been explored by (*i*) learning a probability distribution over data augmentations, e.g., rotation, scaling, and passing the responses of the network to the augmented inputs through a voting mechanism (Benton et al., 2020), or by (*ii*) minimizing the lower bound on the marginal likelihood of the problem to infer the right amount of rotation invariance in the weights of a network (van der Ouderaa & van der Wilk, 2021).

Contrary to these approaches, our method learns full / partial equivariant representations, and is able to modify the extent of equivariance at every layer. Most similar to our approach is Augerino (Benton et al., 2020). Nevertheless, Augerino aims for approximate invariance to *global input transformations*, and requires passing several augmented versions of an input through the network before to provide a response. To the best of our knowledge, our work is the first approach to explore partial group equivariances, and to provide a practical implementation to learn them from data end-to-end.

Group equivariant neural networks. The seminal work of Cohen & Welling (2016) has inspired several methods equivariant to many different groups. Existing convolutional methods show equivariance to planar rotations (Dieleman et al., 2016; Worrall et al., 2017; Weiler et al., 2018b), spherical rotations (Weiler et al., 2018a; Cohen et al., 2018; Esteves et al., 2019a;b; 2020), scaling (Worrall & Welling, 2019; Sosnovik et al., 2019; Romero et al., 2020b), and more general symmetry groups (Romero et al., 2020a; Bogatskiy et al., 2020; Finzi et al., 2021). Group equivariant self-attention has also been proposed (Fuchs et al., 2020; Romero & Cordonnier, 2020; Hutchinson et al., 2021).

Common to all these approaches is that they require fixing the symmetry group to which the model will be *fully* equivariant prior to training. In contrast, our method allows us to define layers approximately equivariant to subsets of the group, and to learn these subsets during training.

3 PREMILINARIES: GROUP EQUIVARIANCE AND THE GROUP CONVOLUTION

Group equivariance: Group equivariance is the property of a map to respect the transformations in a group. We say that a map is equivariant to a group if whenever the input is transformed by elements of the group, the output of the map is equally transformed but not modified. Formally, for a group $\mathcal{G}$ with elements $g \in \mathcal{G}$ acting on a set $\mathcal{X}$, and a map $\phi : \mathcal{X} \to \mathcal{X}$, we say that ϕ is equivariant to $\mathcal{G}$ if:

$$\phi(g \cdot x) = g \cdot \phi(x), \quad \forall x \in \mathcal{X}, \ \forall g \in \mathcal{G}. \tag{1}$$

For example, the convolution of a signal $f : \mathbb{R} \to \mathbb{R}$ and a kernel $\psi : \mathbb{R} \to \mathbb{R}$ is translation equivariant because $\mathcal{L}_t(\psi * f) = \psi * \mathcal{L}_t f$, where $\mathcal{L}_t$ translates the function by t: $\mathcal{L}_t f(x) = f(x - t)$. That is, if the input is translated, its numerical descriptors via the convolution are also translated but not modified.

The group convolution. In order to construct neural networks equivariant to a given group $\mathcal{G}$, it is crucial to define an operation that respects the symmetries in the group. The *group convolution* is such a mapping. It generalizes the convolution to be equivariant to general symmetry groups. Formally, the group convolution of a signal $f : \mathcal{G} \to \mathbb{R}$ and a kernel $\psi : \mathcal{G} \to \mathbb{R}$ is defined as:

$$(\psi * f)(u) = \int_{\mathcal{G}} \psi(v^{-1}u) f(v) \, \mathrm{d}\mu_{\mathcal{G}}(v), \tag{2}$$

where $\mu_{\mathcal{G}}(\cdot)$ is the (invariant) Haar measure of the group. The group convolution is $\mathcal{G}$-equivariant. That is, it holds that, for all $u, v, w \in \mathcal{G}$:

$$(\psi * \mathcal{L}_w f)(u) = \mathcal{L}_w(\psi * f)(u), \text{ with } \mathcal{L}_w f(u) = f(w^{-1}u)$$

The lifting convolution. Normally, input signals of a neural network are not readily defined on the group of interest. Consequently, we require an operation at the beginning of the network that *lifts* the input signal from its original space $\mathcal{X}$, to the space on which the group convolution is defined. This operation is called a *lifting convolution* and is defined as:

$$(\psi * f)(u) = \int_{\mathcal{X}} \psi(v^{-1}u) f(v) \, \mathrm{d}\mu_{\mathcal{G}}(v), \ v \in \mathcal{X}, u \in \mathcal{G}. \tag{3}$$

Practical implementation of the group convolution. The group convolution is defined on a continuous space, and thus it cannot be computed in finite time. Consequently, it must be approximated. Two main strategies exist to approximate the group convolution: (*i*) group discretization, e.g., Cohen & Welling (2016), and (*ii*) Monte Carlo approximations, e.g., Finzi et al. (2020).

Group discretization approximates the group convolution by providing a fixed discretization of the group, e.g., the rotation group as the set of rotations by multiple of 90 degrees. Unfortunately, the resulting operation is equivariant *only* to the transformations considered in the discretization, and not to the entire underlying continuous group.

A better solution is proposed by Finzi et al. (2020). Finzi et al. (2020) provide a Monte Carlo approximation of the group convolution by sampling transformations uniformly from the entire group during each forward pass. Formally, this approximation is given by:

$$(\psi \hat{*} f)(u_i) = \textstyle\sum_j \psi(v_j^{-1} u_i) f(v_j) \bar{\mu}_{\mathcal{G}}(v_j), \tag{4}$$

where $\{u_i\}$ and $\{v_j\}$ are the discretizations of the input and output space of the operation, respectively.

Importantly, this approximation requires the convolutional kernel ψ to be defined on the *continuous group* domain. As the domain cannot be enumerated, ψ cannot be parameterized by independent weights. Instead, Finzi et al. (2020) parameterize convolutional kernels via small neural networks. With this parameterization, all elements $v_j^{-1} u_i$ can be mapped to a well-defined kernel value.

4 PARTIAL GROUP EQUIVARIANT NETWORKS

In this section, we introduce our approach. First, we define the concept of partial group equivariance. Then, we generalize the group convolution to this case in a constructive manner. This allow us to illustrate the properties of the resulting mappings. Next, we discuss how to learn probability distributions on the group elements, and provide concrete examples for discrete groups, continuous groups, and combinations thereof. We conclude by presenting the architecture of Partial G-CNNs.

4.1 PARTIAL GROUP EQUIVARIANCE

Contrary to group equivariance, we say that a map ϕ is *partially equivariant* to $\mathcal{G}$, if the map is equivariant to transformations in a subset of the group $\mathcal{S} \subset \mathcal{G}$, but not to the whole group $\mathcal{G}$:

$$\phi(g \cdot x) = g \cdot \phi(x), \quad \forall x \in \mathcal{X}, \ \forall g \in \mathcal{S}. \tag{5}$$

Different from equivariance to a *subgroup* of $\mathcal{G}$: a subset of the group which also fulfill the group axioms, we do not restrict the subset $\mathcal{S}$ to be itself a group.

As will we show in Sec. 4.2, partial equivariance can only be obtained in an approximate manner. This is because the set $\mathcal{S}$ is not necessarily closed under group actions. Partial equivariance is related to *soft invariance* (van der Wilk et al., 2018): the property of a map to be approximately invariant. We opt for the word *partial* in the equivariant setting, to make clear that it defines a subset of the group.

4.2 FROM GROUP CONVOLUTIONS TO PARTIAL GROUP CONVOLUTIONS

In this section, we generalize the group convolution to describe partial equivariances. Vital to our analysis is the equivariance proof of the group convolution, which we provide here. In addition, we must distinguish between the ***input space*** and the ***output space*** of the group convolution. This distinction is important because they might differ for partial group convolutions.

Proposition 4.1. *The group convolution is $\mathcal{G}$-equivariant in the sense that for all $u, v, w \in \mathcal{G}$:*

$$(\psi * \mathcal{L}_w f)(u) = \mathcal{L}_w(\psi * f)(u), \ \textit{with} \ \mathcal{L}_w f(u) = f(w^{-1}u). \tag{6}$$

Proof.

$$\begin{aligned}(\psi * \mathcal{L}_w f)(u) &= \int_{\mathcal{G}} \psi(v^{-1}u) f(w^{-1}v) \, \mathrm{d}\mu_{\mathcal{G}}(v) = \int_{\mathcal{G}} \psi(\bar{v}^{-1}w^{-1}u) f(\bar{v}) \, \mathrm{d}\mu_{\mathcal{G}}(\bar{v}) \\ &= (\psi * f)(w^{-1}u) = \mathcal{L}_w(\psi * f)(u).\end{aligned}$$

From the first to the second equality, we use the change of variables $\bar{v} := w^{-1}v$. We can do this because the group convolution is a map from the group to itself, and thus if $w, v \in \mathcal{G}$, so does $w^{-1}v$. Since the Haar measure is an invariant measure on the group, we have that $\mu_{\mathcal{G}}(v) = \mu_{\mathcal{G}}(\bar{v})$, for all $v, \bar{v} \in \mathcal{G}$. □

Going from the group $\mathcal{G}$ to a subset $\mathcal{S}$. Crucial to the proof of Proposition 4.1 is the fact the group convolution is an operation from the group to itself. That is, $w^{-1}u$ is a member of the output space for any $w \in \mathcal{G}$ applied to the input space. Consequently, a group transformation applied on the input space can be exactly reflected by an equivalent transformation on the output space.

Now, consider the case in which the output space is not the group $\mathcal{G}$, but instead an arbitrary subset $\mathcal{S} \subset \mathcal{G}$, e.g., rotations in $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Following the proof of Proposition 4.1 with $u \in \mathcal{S}$, and $v \in \mathcal{G}$, we observe that the operation is equivariant to transformations $w \in \mathcal{G}$ as long as $w^{-1}u$ is a member of $\mathcal{S}$. However, if $w^{-1}u$ does not belong to the output space $\mathcal{S}$, the output of the operation cannot reflect an equivalent transformation to that of the input, and thus equivariance is not guaranteed.

Note, however, that equivariance is *only* obtained if Eq. 6 holds for *all* elements in the output space. That is, if $w^{-1}u$ is a member of $\mathcal{S}$, for all elements $u \in \mathcal{S}$. Unfortunately, this is not the case in general, as the output space is not necessarily closed. We emphasize, however, that this difference in responses is what allows the network to disrupt equivariance whenever necessary (Fig. 1).

Importantly, we can precisely quantify the equivariance error of the operation. Intuitively, the equivariance error is given by the difference in the parts of the feature representation that go in and out of $\mathcal{S}$ by the action of an input transformation (Fig. 1). Consequently, the more severe the transformation, the larger the equivariance error is. Fig. 1 illustrates this effect. The feature representation in $\mathcal{S}$ changes gradually for increasing transformations, and becomes entirely different for large transformations (180° rotations in Fig 1). This allows the network to break equivariance and distinguish among input transformations. A formal discussion and the derivation of the equivariance error are provided in Appx. B.1.

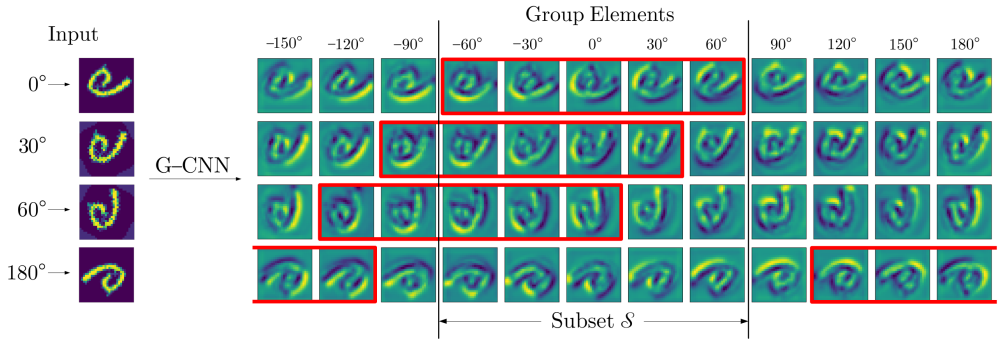
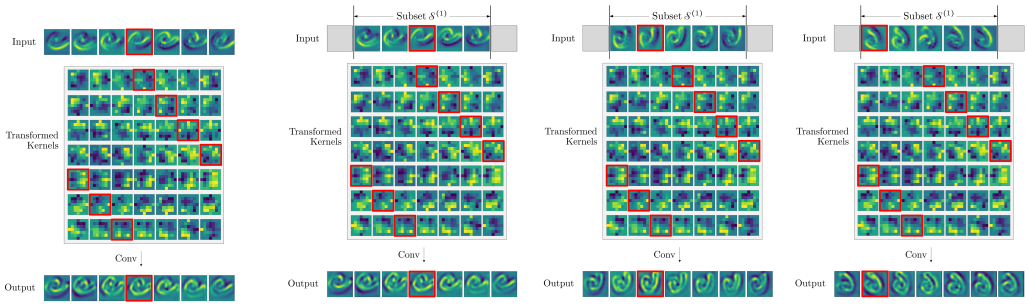


Figure 1: Partial group convolution. The output space of a group convolution is a closed space. As a result, all response components are part of the output space regardless of the transformation applied to the input. Partial group convolutions define a subset which is not necessarily closed ($\mathcal{S}$=$[-60°, 60°]$ here). As a result, the larger the transformation applied to the input, the more the responses within $\mathcal{S}$ change (see responses marked in red). This allows partial group convolutions to distinguish among input transformations. For large transformations (180° here), the responses within $\mathcal{S}$ change entirely.



(a) Group Conv.

(b) Partial Group Convolution on an input subset $\mathcal{S}^{(1)}$

Figure 2: Partial group convolution on a group subset. The group convolution (a) receives a feature on the group, and convolves it with group-transformed versions of the convolutional kernel. Since the input space is the group, i.e., a closed set, the convolutional kernels have access to the *entire* input space for any input transformation. If the input space is a group subset, however, the input values to which the convolutional kernel has access *change* for different input transformations (b). Consequently, the output shows variations in its response for different transformations of the input. This variation is determined by (*i*) the size of the subset $\mathcal{S}^{(1)}$, and (*ii*) the strength of the transformation.

Going from a subset $\mathcal{S}^{(1)}$ to another subset $\mathcal{S}^{(2)}$. Now, consider the case in which the input space $\mathcal{S}^{(1)}$, and the output space $\mathcal{S}^{(2)}$, are both subsets of the group, i.e., $v \in \mathcal{S}^{(1)}$ and $u \in \mathcal{S}^{(2)}$. Analogous to the previous case, equivariance to input transformations $w \in \mathcal{G}$ holds at positions $u \in \mathcal{S}^{(2)}$ for which $w^{-1}u$ are also members of $\mathcal{S}^{(2)}$. Now however, the input space is not restricted to be closed anymore. Consequently, the convolutional kernel ψ has no access to the the same values in the input space $\mathcal{S}$ for different transformed versions of the input $\mathcal{L}_w f$ during the calculation of the group convolution. This induces an additional equivariance error on the output response.

To see this, consider the partial group convolution illustrated in Fig. 2b. For different transformations of the input, the feature representations that fall within $\mathcal{S}^{(1)}$ change. As a result, the output of the group convolution differs for different input transformations, even if the output space of the operation is the entire group. Note that the difference in the output becomes larger for larger input transformations. As shown in Appx. B.2, the equivariance error is determined by (*i*) the size of the subset $\mathcal{S}^{(1)}$, and (*ii*) the strength of the input transformation.

In summary, considering a subset $\mathcal{S}^{(2)}$ in the output space induces an equivariance error in the feature response, as discussed in the previous case. In addition, considering a subset $\mathcal{S}^{(1)}$ in the input space introduces an additional equivariance error produced by changes in the input responses the convolutional kernel has access to, as an effect of input transformations. A formal treatment of this term as well as its derivation is provided Appx. B.2.

Going from a subset $\mathcal{S}$ to the group $\mathcal{G}$. Finally, we can also consider the case in which the input space is a subset $\mathcal{S}$ of the group and the output space is the group itself $\mathcal{G}$. Following the previous analysis, we can see that, this operation is equivariant to all input transformations $w \in \mathcal{G}$, as $w^{-1}u$ is always member of the output space for all values of $u \in \mathcal{G}$. However, not all values on the input space are known, and thus the same error illustrated in the previous case emerges here. This case can be seen as an special case of the previous section, where $\mathcal{S}^{(2)}$=$\mathcal{G}$, and thus the equivariance error induced by considering a subset in the output is zero.

Interestingly, the lifting convolution is an special case of this instance with $\mathcal{S}$=$\mathcal{X}$. Note however, that the lifting convolution is exactly equivariant. This is because the convolutional kernel is represented as a function on $\mathcal{X}$. Consequently, all positions $(x, y) \in \mathcal{X}$ are accessible for all input transformations.

4.3 Learning Group Subsets via Probability Distributions on Group Elements

So far, we have discussed the properties of the partial group convolution without providing details on how to learn subsets of group elements during training. In this section, we describe our approach and provide specific examples for discrete groups, continuous groups, and combinations thereof.

Vital to our approach is the Monte Carlo approximation to the group convolution presented in Sec. 3:

$$(\psi \hat{*} f)(u_i) = \sum_j \psi(v_j^{-1} u_i) f(v_j) \bar{\mu}_{\mathcal{G}}(v_j).$$

As shown in Appx. C, this approximation is equivariant to $\mathcal{G}$ in expectation if the elements in the input and output space, $\{u_i\}, \{v_j\}$ are uniformly sampled from the Haar measure, i.e., $u_i, v_j \sim \mu_{\mathcal{G}}(\cdot)$.[1]

Approach. Our main observation is that we can prioritize sampling specific group elements during the group convolution by learning a probability distribution $\mathrm{p}(g)$ over the elements of the group.

When group convolutions draw elements uniformly from the entire group, each group element is drawn with equal probability and thus, the resulting approximation is fully equivariant in expectation. However, we can also define a different probability distribution with which some samples are drawn with larger probability. For instance, we can sample from a predefined region, e.g., rotations in $[-\frac{\pi}{2}, \frac{\pi}{2}]$, by defining a probability distribution which is uniform in this range, but zero otherwise. In the limit, we can forget a certain equivariance by letting this distribution collapse to a single point, e.g., the identity, along the corresponding group dimension.

In other words, learning a proper probability distribution $\mathrm{p}(g)$ on the group elements can be used to effectively learn a subset of the group. Based on this intuition, we define a probability distribution $\mathrm{p}(g)$ on the output space of the operation, i.e., on $\{u_i\}$, in order to learn a subset of the group $\mathcal{S}^{(2)}$ upon which partial equivariance is defined. Note that we only need to learn a distribution on the output space for each layer. This is because group convolutional layers are applied sequentially, and thus the subset of the input space is learned by a distribution on the output space of the previous layer.

4.3.1 Defining $\mathrm{p}(g)$ for Discrete, Continuous and Multi-Dimensional Groups

We now specify how we learn the distribution $\mathrm{p}(g)$ for continuous, discrete groups and combinations thereof. Although better parameterizations for these distributions might exist, our proposed parameterization readily provides an easy way to define partial equivariances.

Probability distributions for one-dimensional continuous groups. We take inspiration from Augerino (Benton et al., 2020), an use the reparameterization trick (Kingma & Welling, 2013) to parameterize continuous distributions. In particular, we use the reparameterization trick on the Lie algebra of the group (Falorsi et al., 2019) to define a distribution which is uniform over a connected set of group elements $[u^{-1}, \ldots, e, \ldots, u]$, but zero otherwise.

To this end, we define a distribution $\mathfrak{u} \cdot [-1, 1]$ with learnable $\mathfrak{u}$ on the Lie algebra $\mathfrak{g}$, and map it to the group via the pushforward of the exponential map $\exp : \mathfrak{g} \to \mathcal{G}$. This give us a distribution which is uniform over a connected set of elements $[u^{-1}, \ldots, e, \ldots, u]$, but zero otherwise.

For the rotation group $\mathrm{SO}(2)$, we can learn a distribution uniform between $[-\theta, \theta]$ and zero otherwise by defining a probability distribution $\theta \cdot [-1, 1]$ with learnable θ on the Lie algebra, and mapping it to the group. If we parameterize group elements as scalars $g \in [-\pi, \pi)$, the exponential map is the

[1] Finzi et al. (2020) show a similar result where u_i and v_j are the same points and thus $v_j \sim \mu_{\mathcal{G}}(\cdot)$ suffices.

identity, and thus $\mathrm{p}(g)$=$\mathcal{U}(\theta \cdot [-1, 1))$. If we sample group elements from this distribution during the calculation of the group convolution, the output space will only contain elements in $[-\theta, \theta)$ and the output feature map will be partially equivariant.

On a more technical note, an $\exp$-pushforwarded local uniform distribution is locally equivalent to the Haar measure, and thus we can still use the Haar measurement for integration on our group subsets.

Probability distributions for one-dimensional discrete groups. To learn a distribution on a discrete group, we make use of the Gumbel-Softmax trick (Jang et al., 2016; Maddison et al., 2016). In particular, for a discrete group with elements $\{e, g_1, \ldots, g_n\}$, we learn n coefficients $\{p_i\}$, which describe the probability of sampling the element g_i, and use the Straight-Through Gumbel-Softmax to back-propagate through sampling. If all the probabilities $\{p_i{=}1\}$, the group convolution will be fully equivariant. However, whenever some probabilities start declining, group equivariance becomes partial. In the limit, if all probabilities $\{p_i{=}0\}$, the operation effectively forgets this equivariance.

Probability distributions for multi-dimensional groups. Multi-dimensional groups are important for the applications we consider. For instance, the orthogonal group $\mathrm{O}(2)$ is parameterized by rotations and mirroring, or the dilation-rotation group by scaling and rotations. In such cases, we construct the probability distribution over the entire group as a combination of *independent probability distributions along each of these axes*. For a group $\mathcal{G}$ with elements g decomposable along n dimensions g=$(d_1, ..., d_n)$, we decompose the probability distribution as: $\mathrm{p}(g)$= $\prod_{i=1}^{n} \mathrm{p}(d_i)$, where the probability $\mathrm{p}(d_i)$ is defined given the type of space, i.e., continuous or discrete. For instance, for the orthogonal group $\mathrm{O}(2)$ defined by rotations r and mirroring m, i.e., $g = (r, m)$, $r \in \mathrm{SO}(2), m \in \{\pm 1\}$, we define the probability distribution on the group as $\mathrm{p}(g)$=$\mathrm{p}(r) \cdot \mathrm{p}(m)$, where $\mathrm{p}(r)$ is a continuous distribution, and $\mathrm{p}(m)$ is a discrete one.

4.4 The Partial Group Convolution

To conclude, we provide the final definition of the partial group convolution. Let $\mathcal{S}^{(1)}, \mathcal{S}^{(2)}$ be subsets of a group $\mathcal{G}$. For $u \in \mathcal{S}^{(2)}$, $v \in \mathcal{S}^{(1)}$, we define the partial group convolution from $\mathcal{S}^{(1)}$ to $\mathcal{S}^{(2)}$ as:

$$(\psi * f)(u) = \int_{\mathcal{S}^{(1)}} \mathrm{p}(u)\psi(v^{-1}u)f(v)\,\mathrm{d}\mu_{\mathcal{G}}(v), \; u \in \mathcal{S}^{(2)}. \tag{7}$$

Other cases described in Sec. 4.2 can be written as special cases of Eq. 7. Any input values falling outside of $\mathcal{S}^{(1)}$ are assumed to be zero. In practice, we compute Eq. 7 with a Monte Carlo approximation as described in Algorithm 1 (Appx. D).

4.5 Partial Group Equivariant Networks

We built upon the work of Finzi et al. (2020) and construct base G-CNNs equivariant to continuous and discrete groups. This is achieved by combining kernel parameterizations on Lie groups, and actions of discrete groups on the group representation of the kernels. Moreover, we replace the isotropic lifting layer of Finzi et al. (2020) by a convolutional layer with lifting convolutions (Eq. 3). Inspired by Romero et al. (2021), we parameterize our group convolutional kernels as implicit neural representations with SIRENs (Sitzmann et al., 2020). The proposed parameterization shows large improvements in expressivity and convergence speed, and outperforms ReLU-, LeakyReLU and Swish-MLP parameterizations used so far for continuous G-CNNs, e.g., Schütt et al. (2017); Finzi et al. (2020), Tab. 4. Our network structure is shown in Fig. 3, and varies only in the number of blocks and channels.

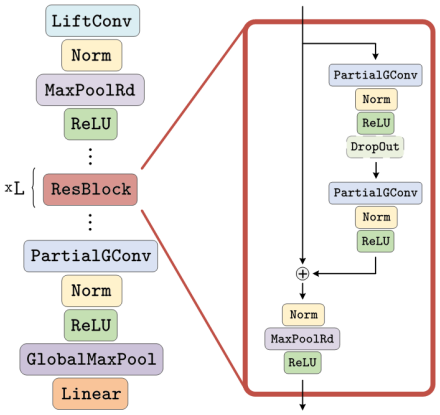


Figure 3: Partial G-CNN architecture.

5 Experiments

Experimental details. We parameterize all our convolutional kernels as a 3-layer SIREN (Sitzmann et al., 2020) with 32 hidden units. All our networks are constructed with two residual blocks of 32

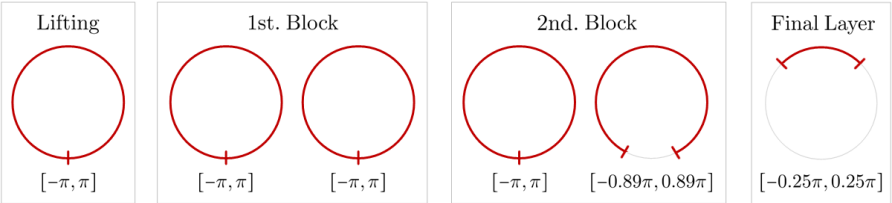


Figure 4: Learned full/partial equivariances for MNIST6-180 with a Partial $\mathrm{SE}(2)$-ResNet.

Table 1: Test accuracy on MNIST6-180 and MNIST6-M.

Model	Base Group	MNIST6-180 Acc. (%)	Base Group	MNIST6-M Acc. (%)	Base Group	MNIST6-M Acc. (%)
G-CNN	$\mathrm{SE}(2)$	50.0	Mirroring	50.0	$\mathrm{E}(2)$	50.0
Partial G-CNN		**100.0**		**100.0**		**100.0**

channels each, batch normalization (Ioffe & Szegedy, 2015) and follow the structure shown in Fig. 3. Our networks are intentionally selected to be simple to better assess the effect of partial equivariance. Importantly, we avoid learning probability distributions along the translation group, and assume all spatial positions to be sampled. This allows us to use fast `PyTorch` primitives in our implementation.[2]

Toy tasks: MNIST6-180 and MNIST6-M. First, we validate the ability of our approach to solve situations in which full equivariance impedes solving a task. To this end, we construct two datasets: *MNIST6-180*, and *MNIST6-M*, by extracting 6 numbers from the MNIST dataset (LeCun et al., 1998), and modifying them either by 180° rotations, or mirroring. The original numbers get the label "original", and the transformed data the label "transformed". The goal is to separate these two classes.

G-CNNs are unable to solve tasks for which discrimination among group transformations are required. As a result, $\mathrm{SE}(2)$-CNNs are unable to separate MNIST6-180 classes, and $\mathrm{E}(2)$-CNNs are unable to separate MNIST6-M classes (Tab 1). Partial G-CNNs with base groups $\mathrm{SE}(2)$, $\mathrm{E}(2)$, on the other hand, seamlessly solve the tasks. Interestingly, we observe that the original MNIST6 class already contains some mirrored numbers, e.g., the input used in Fig. 1, and thus some class overlap exists in MNIST6-M. Nevertheless, Partial G-CNNs are able to solve the task to perfection.

We can also visualize the learned full/partial equivariances at every layer of a network. Fig. 4 illustrates the learned equivariances for a Partial $\mathrm{SE}(2)$-CNN with 2 residual blocks trained on MNIST6-180. In order to solve the task, the network learns to progressively reduce its equivariance as a function of depth, up to a final range of $[-0.25\pi, 0.25\pi]$. Interestingly however, the network does not disrupt equivariance directly at the beginning of the network, but instead decides to stay fully equivariant in the first four layers. We hypothesize that this is because these layers extract low- and mid-level features, which might appear at arbitrary poses. As a consequence, it is advantageous to preserve full equivariance at this hierarchical level. The same behaviour is also present in $\mathrm{E}(2)$-CNNs. However, we observe that discrete distributions are more unstable to train, and other patterns also emerge that, for example, interrupt full equivariance at the second or third layer of the network.

Benchmark image datasets. Next, we validate our approach on the following benchmark image recognition datasets: rotated MNIST (Larochelle et al., 2007), CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009). Additional results on PatchCamelyon (Veeling et al., 2018) can be found in Appx. F. We construct Partial G-CNNs with base groups $\mathrm{SE}(2)$ and $\mathrm{E}(2)$, and varying number of elements used in the Monte Carlo approximation, and compare them to equivalent ResNets and G-CNNs.

Our results are shown in Table 2. Partial G-CNNs are competitive with G-CNNs where full-equivariance is needed, i.e., RotMNIST and PatchCamelyon. However, for tasks in which the data does not naturally exhibit full rotation equivariance, i.e., CIFAR-10 and CIFAR-100, Partial G-CNNs consistently outperform their full equivariant counterparts. Our results support the usage of Partial G-CNNs for settings with both partial and full equivariance.

In addition, we validate our claim that SIRENs are better suited to parameterize group convolutional kernels than existing alternatives. As shown in Tab. 4, $\mathrm{SE}(2)$-CNNs with SIREN kernels consis-

[2]Our code is publicly available at `link-hidden-due-to-double-blind-review`.

Table 2: Test accuracy on vision benchmark datasets.

MODEL	BASE GROUP	NO. ELEMENTS	PARTIAL EQUIV.	CLASSIFICATION ACCURACY (%)		
				ROTMNIST	CIFAR-10	CIFAR-100
ResNet	$\mathrm{T}(2)$	1	-	97.23	83.11	47.99
G-CNN	$\mathrm{SE}(2)$	4	✗	99.10	83.73	52.35
			✓	**99.13**	**86.15**	**53.91**
		8	✗	99.17	86.08	55.55
			✓	**99.23**	**88.59**	**57.26**
		16	✗	**99.24**	86.59	51.55
			✓	99.18	**87.45**	**57.31**
	$\mathrm{E}(2)$	8	✗	**98.14**	85.55	54.29
			✓	97.78	**89.00**	**55.22**
		16	✗	98.35	88.95	57.78
			✓	**98.58**	**90.12**	**61.46**

tently outperform $\mathrm{SE}(2)$-CNNs with kernel parameterizations via ReLU, $\mathrm{LeakyReLU}$ and Swish nonlinearities on all the image benchmarks considered. Our results support the usage of SIRENs as parameterization for continuous group convolutional kernels.

6 DISCUSSION

Memory consumption in partial equivariant settings. G-CNNs fix the number of samples with which the group convolution is approximated prior to training. G-CNNs use the same number of samples for all group convolutional layers regardless of the symmetries present in the data. Partial G-CNNs on the other hand, are able to *dynamically* adjust the number of samples used in order to approximate partial group convolutions. In particular, we fix a maximum number of samples prior to training, and restrict the amount of samples used at every layer based on the subset of the group learned. For instance, a Partial $\mathrm{SE}(2)$-CNN with a learned distribution $\mathrm{p}(u)$=$\mathcal{U}(\frac{\pi}{2}[-1, 1])$ only uses half of the elements used in the corresponding G-CNN. This efficiency in memory leads to important reductions in training time for Partial G-CNNs on partial equivariant settings.

Better kernel parameterization. Inspired by Romero et al. (2021), we replace the Swish kernel parameterization proposed by Finzi et al. (2020) by a SIREN Sitzmann et al. (2020). We observe that this parameterization consistently leads to better results and faster convergence.

Sampling per batch element. In our experiments, we sample once from the learned distribution $\mathrm{p}(u)$ at every layer, and use this sample for all elements in the batch. A better estimation of the probability distribution could be obtained by drawing a different sample for each element in the batch at every layer. Although this method may benefit the learned distributions, it comes at a large memory cost. Consequently, all our reported results are obtained using our original formulation.

Unstable training on discrete groups. Although discrete groups can be successfully modelled in partial equvariance settings, we observe that the discrete distribution used here is not very stable to train. Finding good parameterizations to learn discrete distributions is an active field of research (Hoogeboom et al., 2019; 2021), and advances in this field could lead to more stable training of partial equivariances on discrete groups.

7 CONCLUSION AND FUTURE WORK

We presented Partial Group equivariant Convolutional Neural Networks (Partial G-CNNs): a new family of equivariant methods able to learn partial and full equivariances directly from data at every layer. Partial G-CNNs match the performance of G-CNNs in full equivariant settings, and outperform G-CNNs in settings in which partial group equivariance better represents the data. By learning different levels of equivariance at every layer, Partial G-CNNs are able to maintain full group equivariance in early layers to efficiently learn low-level features appearing at arbitrary poses. However, they are able to restrict it whenever necessary to discriminate among high-level features.

Our method entirely relies on a good parameterization of the probability distribution defined over group elements. Finding better representations and learning methods for these distributions, specially for distributions on discrete variables, is a promising direction for future research.

REFERENCES

Gregory Benton, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. Learning invariances in neural networks. *arXiv preprint arXiv:2010.11882*, 2020.

Alexander Bogatskiy, Brandon Anderson, Jan Offermann, Marwah Roussi, David Miller, and Risi Kondor. Lorentz group equivariant neural network for particle physics. In *International Conference on Machine Learning*, pp. 992–1002. PMLR, 2020.

Shuxiao Chen, Edgar Dobriban, and Jane H Lee. A group-theoretic framework for data augmentation. *Journal of Machine Learning Research*, 21(245):1–71, 2020.

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

Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018.

Nima Dehmamy, Yanchen Liu, Robin Walters, and Rose Yu. Lie algebra convolutional neural networks with automatic symmetry extraction, 2021. URL https://openreview.net/forum?id=cTQnZPLIohy.

Sander Dieleman, Jeffrey De Fauw, and Koray Kavukcuoglu. Exploiting cyclic symmetry in convolutional neural networks. In *International conference on machine learning*, pp. 1889–1898. PMLR, 2016.

Carlos Esteves, Avneesh Sud, Zhengyi Luo, Kostas Daniilidis, and Ameesh Makadia. Cross-domain 3d equivariant image embeddings. In *International Conference on Machine Learning*, pp. 1812–1822. PMLR, 2019a.

Carlos Esteves, Yinshuang Xu, Christine Allen-Blanchette, and Kostas Daniilidis. Equivariant multi-view networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1568–1577, 2019b.

Carlos Esteves, Ameesh Makadia, and Kostas Daniilidis. Spin-weighted spherical cnns. *arXiv preprint arXiv:2006.10731*, 2020.

Luca Falorsi, Pim de Haan, Tim R Davidson, and Patrick Forré. Reparameterizing distributions on lie groups. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3244–3253. PMLR, 2019.

Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *International Conference on Machine Learning*, pp. 3165–3176. PMLR, 2020.

Marc Finzi, Max Welling, and Andrew Gordon Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. *arXiv preprint arXiv:2104.09459*, 2021.

Fabian B Fuchs, Daniel E Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *arXiv preprint arXiv:2006.10503*, 2020.

Emiel Hoogeboom, Jorn WT Peters, Rianne van den Berg, and Max Welling. Integer discrete flows and lossless compression. *arXiv preprint arXiv:1905.07376*, 2019.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Towards non-autoregressive language models. *arXiv preprint arXiv:2102.05379*, 2021.

Michael J Hutchinson, Charline Le Lan, Sheheryar Zaidi, Emilien Dupont, Yee Whye Teh, and Hyunjik Kim. Lietransformer: equivariant self-attention for lie groups. In *International Conference on Machine Learning*, pp. 4533–4543. PMLR, 2021.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

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

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, pp. 473–480, 2007.

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

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

David Romero, Erik Bekkers, Jakub Tomczak, and Mark Hoogendoorn. Attentive group equivariant convolutional networks. In *International Conference on Machine Learning*, pp. 8188–8199. PMLR, 2020a.

David W Romero and Jean-Baptiste Cordonnier. Group equivariant stand-alone self-attention for vision. *arXiv preprint arXiv:2010.00977*, 2020.

David W Romero, Erik J Bekkers, Jakub M Tomczak, and Mark Hoogendoorn. Wavelet networks: Scale equivariant learning from raw waveforms. *arXiv preprint arXiv:2006.05259*, 2020b.

David W Romero, Anna Kuzina, Erik J Bekkers, Jakub M Tomczak, and Mark Hoogendoorn. Ckconv: Continuous kernel convolution for sequential data. *arXiv preprint arXiv:2102.02611*, 2021.

Kristof T Schütt, Pieter-Jan Kindermans, Huziel E Sauceda, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *arXiv preprint arXiv:1706.08566*, 2017.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020.

Ivan Sosnovik, Michał Szmaja, and Arnold Smeulders. Scale-equivariant steerable networks. *arXiv preprint arXiv:1910.11093*, 2019.

Tycho F.A van der Ouderaa and Mark van der Wilk. Learning invariant weights in neural networks. *Workshop in Uncertainty & Robustness in Deep Learning, ICML*, 2021.

Mark van der Wilk, Matthias Bauer, ST John, and James Hensman. Learning invariances using the marginal likelihood. *arXiv preprint arXiv:1808.05563*, 2018.

Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 210–218. Springer, 2018.

Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. *arXiv preprint arXiv:1807.02547*, 2018a.

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

Daniel E Worrall and Max Welling. Deep scale-spaces: Equivariance over scale. *arXiv preprint arXiv:1905.11697*, 2019.

Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5028–5037, 2017.

Allan Zhou, Tom Knowles, and Chelsea Finn. Meta-learning symmetries by reparameterization. *arXiv preprint arXiv:2007.02933*, 2020.

APPENDIX

A GROUPS AND GROUP ACTION

Groups: Group theory is the mathematical language that describes symmetries. The core mathematical object is that of a *group*, and defines what it means for something to exhibit symmetries. Specifically, a group consists of a tuple $(\mathcal{G}, \cdot)$ – set of transformations $\mathcal{G}$, and a binary operation $\cdot$ with the following properties: (i) closure, i.e., $g_1 \times g_2 = g_3 \in \mathcal{G}, \forall g_1, g_2 \in \mathcal{G}$ (ii) associativity, i.e., $g_1 \cdot (g_2 \cdot g_3) = (g_1 \cdot g_2) \cdot g_3$ for all $g_1, g_2, g_3 \in \mathcal{G}$, (iii) the existence of an identity element $e \in \mathcal{G}$, such that $g \cdot e = e \cdot g = g$, and (iv) the existence of an inverse $g^{-1} \in \mathcal{G}$ for all $g \in \mathcal{G}$.

Group action: One can define the *action of the group* $\mathcal{G}$ on a set $\mathcal{X}$. This action describes how group elements $g \in \mathcal{G}$ modify the set $\mathcal{X}$ when the transformation is applied. For instance, the action of elements in the group of planar rotations $\theta \in \mathrm{SO}(2)$ on an image $x \in \mathcal{X}$ –written θx–, depicts how the image x changes when the rotation θ is applied.

B FORMAL TREATMENT OF THE EQUIVARIANCE ERROR IN PARTIAL GROUP CONVOLUTIONS

B.1 PARTIAL GROUP CONVOLUTIONS FROM THE GROUP $\mathcal{G}$ TO A SUBSET $\mathcal{S}$

The partial group convolution from $\mathcal{G}$ to a subset $\mathcal{S}$ can be interpreted as a group convolution in which group elements falling outside of $\mathcal{S}$ are set to zero. Consequently, we can calculate the equivariance error ϵ_{equiv} in the feature representation, by calculating the difference on the subset $\mathcal{S}$ of a group convolution with a group-transformed input $(\mathcal{L}_w f * \psi)$ and a group convolution with a canonical input proceeded by the same transformation on $\mathcal{S}$ $\mathcal{L}_w(f * \psi)$.

The equivariance error given by the effect of a subset in the output space $\epsilon_{\text{equiv}}^{\text{out}}$ is given by:

$$
\begin{aligned}
\epsilon_{\text{equiv}}^{\text{out}} &= \left\| \int_{\mathcal{S}} \mathcal{L}_w(\psi * f)(u)\, \mathrm{d}\mu_{\mathcal{G}}(u) - \int_{\mathcal{S}} (\psi * \mathcal{L}_w f)(u)\, \mathrm{d}\mu_{\mathcal{G}}(u) \right\|_2^2 \\
&= \left\| \int_{w^{-1}\mathcal{S}} (\psi * f)(w^{-1}u)\, \mathrm{d}\mu_{\mathcal{G}}(u) - \int_{\mathcal{S}} (\psi * f)(w^{-1}u)\, \mathrm{d}\mu_{\mathcal{G}}(u) \right\|_2^2 \\
&= \left\| \int_{\mathcal{S}} (\psi * f)(u)\, \mathrm{d}\mu_{\mathcal{G}}(u) - \int_{w\mathcal{S}} (\psi * f)(u)\, \mathrm{d}\mu_{\mathcal{G}}(u) \right\|_2^2 \\
&= \left\| \int_{s_{\min}}^{s_{\max}} (\psi * f)(u)\, \mathrm{d}\mu_{\mathcal{G}}(u) - \int_{ws_{\min}}^{ws_{\max}} (\psi * f)(u)\, \mathrm{d}\mu_{\mathcal{G}}(u) \right\|_2^2 \\
&= \left\| \left(\int_{ws_{\max}}^{s_{\max}} (\psi * f)(u)\, \mathrm{d}\mu_{\mathcal{G}}(u) + \int_{s_{\min}}^{ws_{\max}} (\psi * f)(u)\, \mathrm{d}\mu_{\mathcal{G}}(u) \right) - \right. \\
&\qquad \left. \left(\int_{s_{\min}}^{ws_{\max}} (\psi * f)(u)\, \mathrm{d}\mu_{\mathcal{G}}(u) + \int_{ws_{\min}}^{s_{\min}} (\psi * f)(u)\, \mathrm{d}\mu_{\mathcal{G}}(u) \right) \right\|_2^2 \\
&= \left\| \int_{ws_{\max}}^{s_{\max}} (\psi * f)(u)\, \mathrm{d}\mu_{\mathcal{G}}(u) - \int_{ws_{\min}}^{s_{\min}} (\psi * f)(u)\, \mathrm{d}\mu_{\mathcal{G}}(u) \right\|_2^2
\end{aligned}
$$

From the first line to the second we take advantage of the equivariance property of the group convolution: $(\mathcal{L}_w f * \psi)(u) = \mathcal{L}_w(f * \psi)(u)$, and account for the fact that only the region within $\mathcal{S}$ is visible at the output. We use the change of variables $u = w^{-1}u$ from the second to third line. We specify the boundaries of $\mathcal{S}$ from the third to the fourth line. In the fifth line we separate the integration over $\mathcal{S}$ as a sum of two integrals which depict the same range. In the last line, we cancel out the overlapping parts of the two integrals to come to the final result.

In conclusion, the equivariance error induced by a subset $\mathcal{S}^{(2)}$ on the output space $\epsilon_{\text{equiv}}^{\text{out}}$ is given by the difference between the part of the representation that leaves the subset $\mathcal{S}$, and the part that comes to replace it instead. This behaviour is illustrated in Figure 1.

B.2 PARTIAL GROUP CONVOLUTIONS FROM A SUBSET $\mathcal{S}^{(1)}$ TO A SUBSET $\mathcal{S}^{(2)}$

To isolate the effect of having a group subset as input space, we first consider the output space to be the group, i.e., $\mathcal{S}^{(2)}$=$\mathcal{G}$. The equivariance error is given by the difference across the entire output representation of the group convolution calculated on an input subset $\mathcal{S}^{(1)}$ with a canonical input f, and with a group transformed input $\mathcal{L}_w f$.

The equivariance error $\epsilon^{\text{in}}_{\text{equiv}}$ given by the effect of a subset in the input space is given by:

$$\begin{aligned}
\epsilon_{\text{equiv}} &= \left\| \int_{\mathcal{G}} \int_{\mathcal{S}} \psi(v^{-1}u) f(v) \, \mathrm{d}\mu_{\mathcal{G}}(v) \mathrm{d}\mu_{\mathcal{G}}(u) - \int_{\mathcal{G}} \int_{\mathcal{S}} \psi(v^{-1}u) f(w^{-1}v) \, \mathrm{d}\mu_{\mathcal{G}}(v) \mathrm{d}\mu_{\mathcal{G}}(u) \right\|_2^2 \\
&= \left\| \int_{\mathcal{G}} \left[\int_{\mathcal{S}} \psi(v^{-1}u) f(v) \, \mathrm{d}\mu_{\mathcal{G}}(v) - \int_{\mathcal{S}} \psi(v^{-1}u) f(w^{-1}v) \, \mathrm{d}\mu_{\mathcal{G}}(v) \right] \mathrm{d}\mu_{\mathcal{G}}(u) \right\|_2^2 \\
&= \left\| \int_{\mathcal{G}} \int_{\mathcal{S}} \psi(v^{-1}u) \left[f(v) - f(w^{-1}v) \right] \mathrm{d}\mu_{\mathcal{G}}(v) \mathrm{d}\mu_{\mathcal{G}}(u) \right\|_2^2
\end{aligned}$$

In other words, the equivariance error induces by a subset $\mathcal{S}^{(1)}$ on the input space $\epsilon^{\text{in}}_{\text{equiv}}$ is given by the difference in $\mathcal{S}^{(1)}$ the canonical input f, and the part that comes to replace it instead.

C EQUIVARIANCE PROPERTY OF MONTE-CARLO APPROXIMATIONS

Consider the Monte-Carlo approximation depicted in Eq. 4:

$$(\psi \hat{*} f)(u_i) = \textstyle\sum_j \psi(v_j^{-1} u_i) f(v_j) \bar{\mu}_{\mathcal{G}}(v_j).$$

For a transformed version of the $\mathcal{L}_w f$, we can show that the Monte-Carlo approximation of the group convolution is equivariant in expectation. The proof follows the same steps than Finzi et al. (2020). However, the last step of the proof has a different reason, resulting from the fact that input and output elements can be sampled from different distributions.

For a transformed version of the $\mathcal{L}_w f$, we have that:

$$\begin{aligned}
(\psi \hat{*} \mathcal{L}_w f)(u_i) &= \textstyle\sum_j \psi(v_j^{-1} u_i) f(w^{-1} v_j) \bar{\mu}_{\mathcal{G}}(v_j) \\
&= \textstyle\sum_j \psi(\tilde{v}_j^{-1} w^{-1} u_i) f(\tilde{v}_j) \bar{u}_{\mathcal{G}}(\tilde{v}_j) \\
&\overset{d}{=} (\psi \hat{*} f)(w^{-1} u_i) = \mathcal{L}_w (\psi \hat{*} f)(u_i)
\end{aligned}$$

From the first to the second line, we use the change of variables $\tilde{v}_j = w v_j$ and the fact that, group elements in the input space are sampled from the Haar measure for which it holds that $\bar{\mu}_{\mathcal{G}}(v_j)$=$\bar{\mu}_{\mathcal{G}}(\tilde{v}_j)$. However, from the second to the third line, we must also assume that this holds for the output space. That is, that the probability of drawing $w^{-1} u_j$ is equal to that of drawing u_j. This is of particular importance for the partial equivariance setting, in which this might not be the case.

D ALGORITHM FOR MONTE-CARLO APPROXIMATION OF THE PARTIAL GROUP CONVOLUTION

Algorithm 1: The Partial Group Convolution Layer

Inputs: position, function-value tuples on the group or a subset thereof $\{v_j, f(v_j)\}$.
Outputs: convolved position, function-value tuples on the output group subset $\{u_i, (f \hat{*} \psi)(u_i)\}$.
$\{u_i\} \sim \mathrm{p}(u)$; `/* Sample elements from p(u)`
for $u_i \in \{u_i\}$ **do**
  $h(u_i) = \sum_j \psi(v_j^{-1} u_i) f(v_j) \bar{\mu}_{\mathcal{G}}(v_j)$; `/* Compute group convolution (Eq. 4)`
end
return $\{u_i, h(u_i)\}$

Table 3: Image recognition accuracy on PatchCam dataset.

MODEL	BASE GROUP	NO. ELEMENTS	PARTIAL EQUIV.	CLASSIFICATION ACCURACY ON PATCHCAM (%)
ResNet	T(2)	1	-	67.59
G-CNN	SE(2)	8	✗	**89.87**
			✓	89.07
		16	✗	89.71
			✓	**<u>90.31</u>**
	E(2)	16	✗	**89.77**
			✓	88.13

E ARCHITECTURE AND HYPERPARAMETER DETAILS

We note that all the hyperparameters were chosen based on the best performance for the fully-equivariant networks and the same parameters were used for training the proposed Partial G-CNNs.

For all the experiments shown in this paper, we use residual networks (shown in the main paper in Figure 3) with an initial lifting convolutional layer followed by 2 ResBlocks with full/partial group convolutional layers (or regular convolutional layers for the non-equivariant ResNet baselines). For all the networks for all the datasets, we use 32 feature maps in the hidden layers. We employ Batch Normalization and ReLU non-linearities as shown in the figure. For MNIST6-M and MNIST6-180, max-pooling is performed after each of the ResBlock partial/full group convolutional layers. In the case of rotMNIST, max-pooling is performed after the lifting convolutional layer and the first group convolutional layers. For CIFAR-10 and CIFAR-100, we use max-pooling layers after each of the group convolutional layers. Finally, for PatchCamelyon, we apply the max-pooling operation after the lifting convolution as well as the both the group convolutional layers. At the end of the network, a global max-pooling layer is used to create the invariant before classification.

The networks for MNIST6-180, MNIST6-M, rotMNIST, CIFAR-10 and CIFAR-100 are trained for 300 epochs and the networks for PatchCamelyon are trained for 30 epochs. We use a batch size of 64 for all networks. We use an initial learning rate of $1e-3$ and Adam optimizer with cosine annealing with 5 epochs of linear warm-up. In the case of CIFAR-10, CIFAR-100 and PatchCamelyon datasets, we also use a weight decay of 0.0001. For roMNIST, CIFAR-10, CIFAR-100 and PatchCamelyon, we also use a different learning rate of $1e-4$ for only the parameters of the probability distributions over the group elements.

F ADDITIONAL EXPERIMENTAL RESULTS

Classification results on PatchCamelyon. Table 3 shows the results obtained for ResNet, G-CNNs and Partial G-CNNs on the PatchCamelyon dataset (Veeling et al., 2018). Partial G-CNNs match the performance of G-CNNs in this full equivariant setting, as expected.

We observe that the learned distributions over the group elements for rotMNIST and PatchCamelyon are exactly consistent with full-equivariance, and for CIFAR-10 and CIFAR-100, they clearly show partial equivariance

Evaluation of existing kernel parameterizations. We observe that SIREN kernels consistently lead to better accuracy than other existing kernel parameterizations. Our results are shown in Tab. 4.

Table 4: Comparison of kernel parameterizations.

Model	No. Elements	Kernel Type	Classification Accuracy (%)		
			RotMNIST	CIFAR-10	CIFAR-100
SE(2)-CNN	4	ReLU	96.49	59.95	28.01
		LeakyReLU	94.47	56.19	27.36
		Swish	94.41	66.12	34.20
		SIREN	**99.10**	**83.73**	**52.35**
	8	ReLU	97.73	68.29	37.81
		LeakyReLU	97.65	68.94	36.30
		Swish	97.72	69.20	34.10
		SIREN	**99.17**	**86.08**	**55.55**
	16	ReLU	98.49	66.84	37.72
		LeakyReLU	98.53	68.01	38.29
		Swish	98.55	65.99	37.72
		SIREN	**99.24**	**86.68**	**51.51**