On Fairly Comparing Group Equivariant Convolutional Networks

Lucas Roos [1] **Steve Kroon** [1 2]

Editors: *S. Vadgama, E.J. Bekkers, A. Pouplin, S.O. Kaba, H. Lawrence, R. Walters, T. Emerson, H. Kvinge, J.M. Tomczak, S. Jegelka*

Abstract

This paper investigates the flexibility of Group Equivariant Convolutional Neural Networks (G-CNNs), which specialize conventional neural networks by encoding equivariance to group transformations. Inspired by splines, we propose new metrics to assess the complexity of ReLU networks and use them to quantify and compare the flexibility of networks equivariant to different groups. Our analysis suggests that the current practice of comparing networks by fixing the number of trainable parameters unfairly affords models equivariant to larger groups additional expressivity. Instead, we advocate for comparisons based on a fixed computational budget—which we empirically show results in more similar levels of network flexibility. This approach allows one to better disentangle the impact of constraining networks to be equivariant from the increased expressivity they are typically granted in the literature, enabling one to obtain a more nuanced view of the impact of enforcing equivariance. Interestingly, our experiments indicate that enforcing equivariance results in *more* complex fitted functions even when controlling for compute, despite reducing network expressivity.

1. Introduction

Convolutional neural networks (CNNs) exploit the translational symmetry present in virtually all image data of interest to the deep learning community: features are semantically similar regardless of their location in the image. Instead of learning this symmetry from the training data, convolutional layers in CNNs are constrained a priori to be *equivariant* to translations. Group equivariant convolutional neural networks (G-CNNs) (Cohen & Welling, 2016) provide a framework for generalizing translational convolutions, allowing for the creation of architectures that are equivariant to other group transformations. G-CNNs have shown to be beneficial for a variety of tasks exhibiting symmetry, both theoretically (Kondor & Trivedi, 2018; Elesedy & Zaidi, 2021) and empirically (Walters et al., 2021; Dey et al., 2021; Lafarge et al., 2021; Bekkers et al., 2018; Zhu et al., 2022). G-CNNs are straightforward to implement for discrete groups and have also shown improvements over regular CNNs on more traditional image recognition tasks (Cohen & Welling, 2016; Weiler & Cesa, 2019; Hoogeboom et al., 2018). However, they have yet to enter mainstream use, with most practitioners still employing regular CNNs.

One drawback of incorporating group equivariance into architectures, potentially acting as a barrier to mainstream adoption, is the added computational expense: when the number of trainable parameters in a network is fixed, the computational cost of applying a group convolution scales linearly with the group size. In order to demonstrate the efficacy of their models, many works (Cohen & Welling, 2016; Knigge et al., 2022; Hoogeboom et al., 2018; Weiler & Cesa, 2019; Klee et al., 2023) compare "more-equivariant" networks—networks equivariant to larger groups—to their original, "less-equivariant", counterparts, by adjusting their layer widths to equate the number of trainable parameters. In this work, we argue that this practice is not well-motivated, as it does not consider how the symmetries in the data constrain the learning of models that do not directly enforce equivariance to these symmetries: if a less-equivariant network needs to learn to produce equivariant features in order to achieve a low training error—which is especially applicable in the presence of data augmentation—then the less-equivariant network's trainable parameters will be regularized towards equivariant solutions. This can be seen in

[1]Computer Science Division, Stellenbosch University, South Africa [2]National Institute for Theoretical and Computational Sciences, South Africa. Correspondence to: Lucas Roos <roosluan@gmail.com>, Steve Kroon <kroon@sun.ac.za>.

Proceedings of the Geometry-grounded Representation Learning and Generative Modeling at the 41st International Conference on Machine Learning, Vienna, Austria. PMLR Vol Number, 2024.

the observation that the filters in the first layer of CNNs tend to learn multiple copies of one another in different poses (Knigge et al., 2022), mimicking equivariance in order to minimize the training error. Equivariant networks, on the other hand, have a priori been constrained to be equivariant, and so their trainable parameters are unaffected by the soft-constraints towards equivariance imposed on them by the data. Thus, in the context of tasks and training procedures that benefit from encoding symmetries, less-equivariant networks have fewer *effective* parameters than more-equivariant networks, giving the latter an advantage in terms of model flexibility, and making comparisons by equating the number of trainable parameters somewhat unfair.

Recent works (Balestriero & Baraniuk, 2018; 2021) have investigated the connection between deep networks with piecewise linear activations and affine spline functions, and several works (Takai et al., 2021; Hanin & Rolnick, 2019; Montufar et al., 2014; Novak et al., 2018) have quantified the expressivity of different networks in terms of the number of affine regions they can partition their input space into. Inspired by these works, we measure the number of affine regions in networks equivariant to different groups, along with several other spline-inspired complexity metrics, and use these as proxies for model flexibility. Based on these proposed complexity metrics, we empirically verify that equating the number of trainable parameters gives more-equivariant networks an advantage in terms of model flexibility. Instead, we propose comparing networks using a fixed computational budget—a constraint more relevant to practitioners—and illustrate that this more fairly equates model flexibility.

In summary, this work:

- Proposes complexity metrics for piecewise affine neural networks based on viewing them as affine splines. Using these, it is shown that equating the number of trainable parameters affords more-equivariant networks substantially more flexibility.
- Demonstrates that equating *compute* more effectively controls for flexibility, allowing one to better disentangle the impact of enforcing equivariance from the increased expressivity they are typically granted in the literature.
- Further investigates the impact of enforcing equivariance on the proposed spline complexity metrics, after more fairly controlling for model flexibility by equating compute. This uncovers some interesting differences between networks equivariant to different groups.

2. Background

2.1. Equivariance

A function $\Phi : \mathcal{X} \to \mathcal{Y}$ is equivariant to a group G with respect to group actions $\cdot$ on $\mathcal{X}$, and $\circ$ on $\mathcal{Y}$, if $\Phi(g \cdot x) = g \circ \Phi(x)$ for all $g \in G$[1]. Invariance is a special case of equivariance, where $g \circ y = y$ for all $g \in G$, $y \in \mathcal{Y}$, and hence $\Phi(g \cdot x) = \Phi(x)$.

2.2. Equivariant Distributions

In a classification problem, two distributions are of primary interest: the input distribution $p(x)$, and the conditional label distribution $p(y|x)$. We say a distribution is invariant to G if $p(g \cdot x) = p(x)$, for all $g \in G$. Similarly, a conditional distribution is equivariant to G if $p(y|x) = p(g \circ y|g \cdot x)$, and invariant if $g \circ y = y$, for all $g \in G$.

2.3. Group Convolution

Cohen & Welling (2016) introduced group convolutional layers, which are linear neural network layers that generalize regular convolutions to be equivariant to a specified group G. By composing group convolution layers, interspersed with pointwise nonlinearities, and appropriately designed batch-norm, dropout, and pooling layers, one obtains a network that is end-to-end equivariant to the action of G.

When dealing with convolution it is convenient to view network features as functions rather than vectors. Every layer in a neural network can be viewed as a *feature map*, $\mathcal{F} = \{f_c : c \in \mathbb{N}; c \leq C\}$, consisting of a set of *signals*, $f_c : G \to \mathbb{R}$, stacked over C channels. To propagate a feature map forward through the network, it is convolved with a *filter*, $\Psi^k = \{\psi_c^k : c \in \mathbb{N}; c \leq C\}$, consisting of a set of *kernels*, $\psi_c^k : G \to \mathbb{R}$, stacked over C channels. A *filter bank* consists of K stacked filters $\Psi = \{\Psi^k : k \in \mathbb{N}; k \leq K\}$, each with C channels.

Let a group H act transitively on a space G, denoted by juxtaposing symbols. Then the group convolution[2] over H of a feature map, $\mathcal{F}$, with a filter, Ψ^k, is defined as

$$\left[\mathcal{F} * \Psi^k\right](h) = \sum_{c=1}^{C} \sum_{g \in G} f_c(g)\, \psi_c^k\left(h^{-1} g\right). \qquad (1)$$

2.4. Parameters and Compute of Group Convolutions

Inspecting the group convolution in Equation (1) reveals that the filter bank of a group convolution can be stored in a tensor of shape $(K, C, |G|)$, where C and K are the numbers of input and output channels respectively, and $|G|$ is the order of the incoming group, G[3]. The $KC|G|$ values

[1] We use G to refer to both the group as a whole, or its underlying set, depending on the context.

[2] Although technically correlation, it is commonly referred to as convolution in the deep learning community.

[3] The number of elements in the translation group is technically infinite, but convolutional layers typically assume the signals and kernels drop off to zero outside a (pre-specified) region.

stored inside this tensor are referred to as the *trainable parameters* of the layer. In order to implement the group convolution, each filter in the filter bank is transformed $|H|$ times by the action of the outgoing group H, expanding the filter bank to dimensions $(K, |H|, C, |G|)$. We refer to this tensor as the *group-expanded filter bank*, and to its elements as the *group-expanded parameters*. Similarly, we will refer to K and C as numbers of channels, with $K|H|$ and $C|G|$ as the corresponding numbers of *group-expanded channels*—for G-CNN layers, equating group-expanded channels is equivalent to equating group-expanded parameters. For simplicity, we assume going forward that the incoming and outgoing groups and number of channels are identical. This is true of all the architectures we use in our experiments, except for their initial ("lifting") and output layers.

Suppose one is comparing two architectures that are equivariant to different groups, G_1 and G_2, but are otherwise identical. If one desires to equate the number of trainable parameters (i.e. the filter bank sizes) between instances of these architectures, then for every layer, the number of channels, C_1 and C_2, must satisfy $C_1\sqrt{|G_1|} = C_2\sqrt{|G_2|}$. We refer to this approach of comparing architectures as the "trainable parameters" approach (TP). However, ignoring the relatively small cost of expanding the filter bank, the number of operations and memory requirements for forward and backward passes through (typical implementations of) these layers depend on the size of the group-expanded filter banks. In order to equate computational cost (i.e. equate group-expanded filter bank sizes) the number of channels must satisfy $C_1|G_1| = C_2|G_2|$. We refer to this approach of comparing models as the "group-expanded parameters" approach (GEP).

2.5. Spline Perspective on Model Complexity

In Section 2.5.1, we describe some properties of neural networks with continuous piecewise affine (CPWA) activation functions, and in Sections 2.5.2 and 2.5.3, we propose complexity metrics based on smoothing splines.

2.5.1. CPWA Networks' Polytopal Complexes

Because composing CPWA functions results in a CPWA function, networks consisting of affine layers interspersed with CPWA nonlinearities are themselves also CPWA. These networks can be modeled by high-dimensional affine splines, an idea explored by Balestriero & Baraniuk (2018; 2021). Every affine region in a CPWA network's input space has an associated weight matrix and bias vector, and is supported on a polytope. Collectively, these polytopes partition the input space into a *polytopal complex*. Figure 2 provides an example of such an extracted polytopal complex in two dimensions, for a ReLU network invariant to horizontal and or vertical flips.

Although exact methods for extracting the full polytopal complex from a given neural network exist, their computational cost suffers from the curse of dimensionality: the number of polytopes scales exponentially with the input dimension (Montufar et al., 2014). A more tractable subproblem is to instead extract the complex on a lower-dimensional subspace. We extract the polytopal complex of networks over multiple stitched-together line segments, called a polyline. Parameterized over this curve, the network is simplified to a univariate CPWA function with multivariate output.

2.5.2. Spline Complexity Metrics

When fitting an affine spline with multivariate output on a given interval of interest, $[0, L]$, there are two main ways of controlling the flexibility of the model: first, by selecting the number of knots, K, and their locations (in ascending order), $\{t_1, \ldots, t_K\}$, which partition $[0, L]$ into $K + 1$ intervals; second, by choosing how the $K + 1$ weight vectors, $W = \{\mathbf{w}_0, \ldots, \mathbf{w}_K\}$, and bias vector, $\mathbf{b}$, of the affine regions are fitted. Setting $\mathbf{f}_{W,\mathbf{b}}(0) = \mathbf{b}$, $t_0 = 0$ and $t_{K+1} = L$, the spline can be written as

$$\mathbf{f}_{W,\mathbf{b}}(t) = \mathbf{f}_{W,\mathbf{b}}(t_k) + \mathbf{w}_k(t - t_k), \text{ for } t_k \leq t \leq t_{k+1}.$$

The model parameters W and $\mathbf{b}$ are fit by minimizing a regularized objective function containing a complexity penalty term that controls the flexibility of the model. The knot locations and complexity penalty are hyperparameters of the model, typically selected using cross-validation. Smoothing splines use a smoothness penalty as a complexity criterion, given by $S(f) = \int_0^L |f''(t)|^2 dt$, where f is the spline function. In this form, the penalty isn't directly applicable to CPWA functions, as their second derivatives are undefined at their knots, and zero elsewhere. Instead, we use a discrete analogue, that is also applicable to univariate CPWA functions with multivariate output, given by

$$S(W, \mathbf{b}) = \sum_{k=0}^{K} \|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2.$$

Novak et al. (2018) measured the expected Frobenius norm of the Jacobian of trained networks. Similarly, we also measure the expected *gradient* norm over the curve,

$$\mathrm{Sn}(W, \mathbf{b}) = \frac{1}{L} \sum_{k=0}^{K} \|\mathbf{w}_k\| (t_{k+1} - t_k),$$

which we refer to as *sensitivity*, because it describes how quickly the function changes when the input is perturbed.

2.5.3. KNOT COMPLEXITY

In addition to the *number* of knots, one can also assess the complexity of the knot *distribution*. We quantify the complexity of the knot distribution, F, as the degree to which it deviates from a uniform distribution, $U(t) = \frac{t}{L}$, using the Cramér–von Mises criterion (Cramér, 1928),

$$\omega^2(F) = \frac{1}{L}\int_0^L \left(F(t) - \frac{t}{L}\right)^2 dt. \tag{2}$$

Typically, an empirical cumulative distribution function (ECDF) is used to estimate F from observed samples, but for our purposes, we can improve on this by fitting a piecewise linear distribution function instead (see Appendix A for a full discussion). We refer to Equation (2) as a *uniformity* criterion and illustrate it in Figure 1.

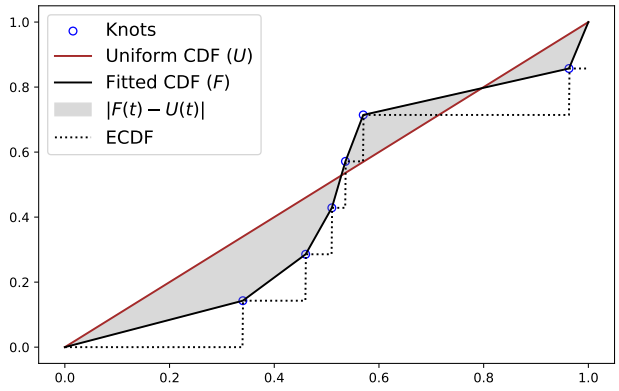

Figure 1. The fitted knot distribution and resulting uniformity criterion visualized on the interval $[0, 1]$ for six observed knots. An ECDF using the number of observations plus one as a normalizing constant is provided for comparison. Shaded in gray is the difference between the fitted distribution and the uniform CDF, which is squared and then integrated to compute the uniformity criterion.

3. Methods

3.1. Locating the Polytopal Complex Boundaries in One Dimension

When performing a forward pass of a test point $\mathbf{x}$ through a neural network, $\mathbf{\Phi}$, one can create a binary string by concatenating the sign of every neuron's pre-activation, denoted by $\Phi^{(i)}(\mathbf{x})$ for the ith neuron, coding positive values with 1. This is called the *activation pattern* of $\mathbf{\Phi}$ at $\mathbf{x}$, which we denote by $\mathbf{A}_{\mathbf{\Phi}}(\mathbf{x})$. If $\mathbf{\Phi}$ is a ReLU network, then connected regions with identical activation patterns form convex polytopes.

Suppose one would like to find the smallest timestep $t > 0$, such that $\mathbf{A}_{\mathbf{\Phi}}(\mathbf{x}) \neq \mathbf{A}_{\mathbf{\Phi}}(\mathbf{x} + \mathbf{v}t)$, for a given location, $\mathbf{x}$, and direction, $\mathbf{v}$. This coincides with (at least) one neuron's pre-activation crossing zero, and in keeping with spline terminology, we refer to these timestep locations as *knots*. Let $s^{(i)}(\mathbf{x}, \mathbf{v})$ denote the rate of change, called the *velocity*, of the ith neuron's pre-activation in the (positive) direction of $\mathbf{v}$, given by

$$s^{(i)}(\mathbf{x}, \mathbf{v}) = \left.\frac{\partial \Phi^{(i)}(\mathbf{x} + \mathbf{v}t)}{\partial_+ t}\right|_{t=0},$$

where ∂_+ denotes the derivative from the right. This can efficiently be calculated for all i using forward-mode auto-differentiation. The next knot then lies at[4]

$$t_{\text{next}} = \min_i \left\{ \frac{\Phi^{(i)}(\mathbf{x})}{s^{(i)}(\mathbf{x}, \mathbf{v})} : \left(\Phi^{(i)}(\mathbf{x})\right)\left(s^{(i)}(\mathbf{x}, \mathbf{v})\right) < 0 \right\},$$

with no further knots in the direction of $\mathbf{v}$ if the set is empty. Using this formula repeatedly, one can, up to numerical precision, find every knot of the network on a line segment, $\mathbf{x}(t) = \mathbf{x}_0 + \mathbf{v}t$, $t \in [0, 1]$.

3.2. Selection of Control Points

We focus on describing neural networks' behavior on the data manifold, as this is the primary region in which they are trained and used. For simple, low-dimensional data, this can be reasonably accomplished by hand, as illustrated in Figure 2, but for datasets with large input dimension, a more methodical approach is necessary. Novak et al. (2018) analyzed the behavior of trained networks over ellipses passing through sampled triples of datapoints; however, they note that these trajectories deviate considerably from the data manifold when the datapoints belong to different classes.

To focus our attention on the data manifold, we instead use a variational auto-encoder (VAE), independently trained on the same dataset as the neural networks to be analyzed. First, we select two points in latent space, such as two latent class means of the dataset. Next, we uniformly sample many points on the line between these two points, and decode them to input space. The resulting sequence of *control points* defines a polyline in input space, for which we extract the univariate CPWA functions of trained networks using the approach described in Section 3.1.

3.3. Isolating the Impact of Enforcing Equivariance

When comparing flexibility between networks equivariant to different groups, it is important to consider the impact orientation has on their predictions. For example, distinguishing

[4] Max-pooling layers can be dealt with similarly, with a transition occurring when the maximum neuron preactivation changes.

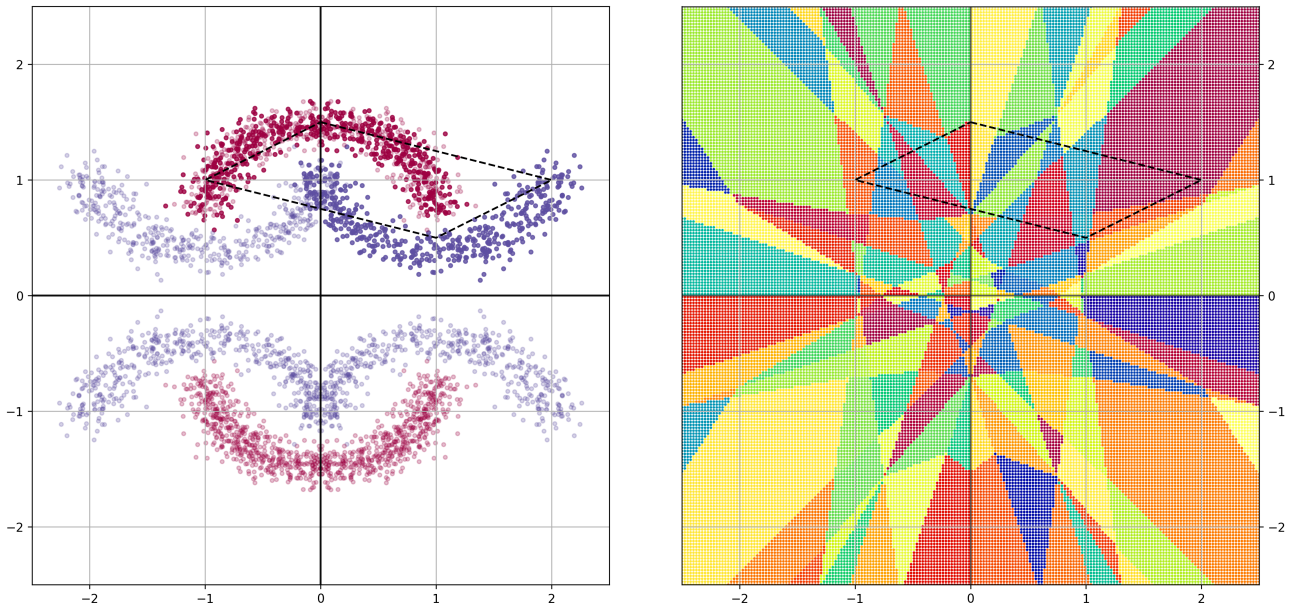

Figure 2. Left: A moons dataset, augmented with vertical and or horizontal flips. Canonical samples are displayed with a higher opacity than augmented samples. Right: The (approximate) polytopal complex extracted from a small ReLU network invariant to horizontal and or vertical flips trained on the (left) moons dataset. Black dashed lines illustrate the polyline over which the polytopal complex is extracted for our experiments.

between a 6 and a 9 is straightforward if the orientation of the digit in question is known; if not, one must rely on identifying the more subtle differences between the way 6's and 9's are typically drawn, which is an inherently more difficult task. Invariant networks are thus at a disadvantage when inputs are presented in a canonical orientation, thereby conveying class information, as they are essentially blind to orientation. Additionally, invariant networks' input samples are effectively augmented with all transformations of the group, giving invariant networks larger *effective* training sets.

By augmenting the training data uniformly at random with transformations from the group, regular and invariant networks obtain identical effective training sets, and any class information conveyed via orientation is destroyed. This levels the playing field between regular and invariant networks, enabling one to isolate the impact of enforcing equivariance on the complexity of their fitted functions.

4. Investigation

This section investigates the impact of constraining networks to be equivariant on the complexity of their fitted functions, for subgroups of the square and discrete translations. Specifically, we consider the impact on the univariate CPWA functions extracted from the networks. Empirical evidence is obtained by studying these extracted functions. The CPWA functions are extracted over polylines close to the data manifold, defined by selected sequences of control points in their input space. For the toy dataset control points are selected by hand, illustrated in Figure 2, while for image datasets we use the approach discussed in Section 3.2, interpolating between class means in latent space. We analyze the extracted functions using the complexity metrics described in Section 2.5.2, and compare them across networks equivariant to different groups.

All architectures utilize ReLU nonlinearities and appropriate equivariant batch normalization at every layer, and max pooling at the end of the network to reduce equivariance to invariance for the final classification layer. As motivated in Section 3.3, datasets are made invariant to the desired group of transformations through online augmentation during both training and testing. We use the Adam optimizer with default hyperparameter settings in PyTorch, training until the training loss does not improve for three epochs. Results are aggregated over 50 different random seeds for each configuration, with each seed producing a different weight initialization and data training/test split.

Moons K_4 We first investigate the impact of enforcing group equivariance in multilayer perceptrons (MLPs) on an illustrative toy dataset, shown in Figure 2. This dataset is the moons dataset augmented with transformations from the

group of horizontal and or vertical flips, isomorphic to the Klein four-group, K_4. We train MLPs with three different groups of invariance: the trivial group (order 1), the group of vertical flips (FlipW—order 2), and the group generated by both vertical and horizontal flips (K_4—order 4). Each MLP consists of two hidden layers with a constant number of hidden channels. For every group setting, we sweep the number of hidden group-expanded channels over the values $\{12, 16, 24, 32, 44\}$. We extract and analyze the CPWA functions of the trained networks over a polyline lying suitably close to the data manifold, illustrated in Figure 2.

MNIST $O(2)$ As in Weiler & Cesa (2019), we augment MNIST to be invariant to the orthogonal group ($O(2)$) by randomly rotating and flipping images during training and testing. We reproduce the P4CNN architecture[5] of Cohen & Welling (2016), which consists purely of group convolutional layers and pooling operators. P4CNN is equivariant to discrete translations and a selected subgroup of the symmetries of the square, which is the group generated by flips and $90°$ degree rotations. We sweep over different numbers of group-expanded channels, $\{16, 24, 32, 48, 64\}$, and subgroups of the symmetries of the square, {Trivial, FlipH, Rot90, FlipRot90}[6].

First, we first select a sequence of all 10 digits such that it minimizes the shortest-traveling salesman tour of their latent class means, for a VAE trained on the original MNIST dataset (without augmentation). Next, from this sequence of latent class means, we generate a polyline in input space using the approach described in Section 3.2, sampling a thousand samples between every class mean. This ensures that every digit is visited exactly once, while keeping the distance traveled in latent space as short as possible. Finally, we extract the CPWA functions of the trained networks over this polyline.

Downsampled MNIST $\mathbb{Z}_7^2$ Comparing CNNs to conventional multilayer perceptrons (MLPs) poses additional challenges. Firstly, CNNs typically use kernel sizes smaller than the image itself, effectively setting certain weights to zero. Secondly, creating a CNN with the same number of trainable parameters as an MLP is expensive, as the equivariance group of a CNN is much larger than, for example, the symmetries of the square, which has order 8—the (cyclic) translation group on an MNIST image has order $28 \times 28 = 784$. To address these challenges, we opt to downsample each MNIST image to 7×7 pixels, by averaging 4×4 blocks of pixels, and then use cyclic convolutions with full 7×7 kernels for our CNNs. This approach reduces the cyclic translation group order to a more manageable $7 \times 7 = 49$, and removes the effective "zero-weights" induced by small kernel sizes, allowing for direct comparison between CNNs and MLPs. Further, during both training and testing, samples are augmented with transformations from the group of discrete cyclic shifts, isomorphic to $\mathbb{Z}_7^2$.

We sweep over different numbers of group-expanded channels, $\{49, 147, 343, 931\}$, and groups: the trivial group (order 1), the group of cyclic horizontal translations (Translate H—order 7), and the group of both horizontal and or vertical translations ($\mathbb{Z}_7^2$—order 49). We extract and analyze the CPWA functions of the fitted networks over the same polyline as for our regular MNIST experiments, downsampled to 7×7 pixels.

4.1. Results and Discussion

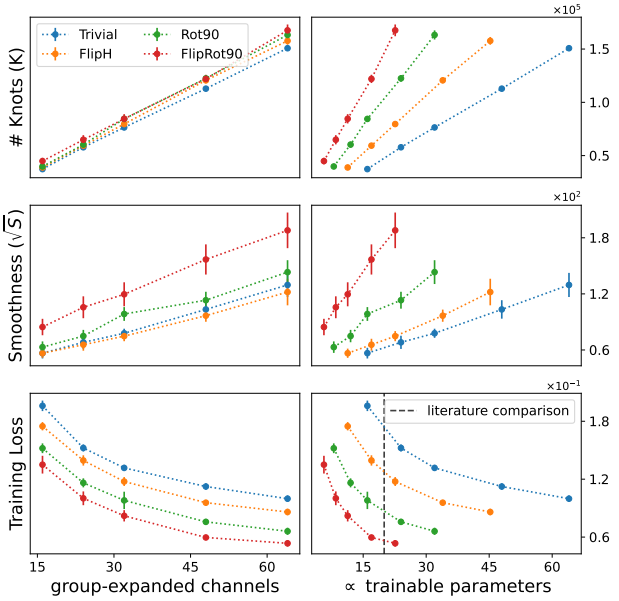

Figure 3. The number of knots, smoothness criterion, training loss, and test loss are recorded over varying numbers of group-expanded channels for MNIST $O(2)$. The left column shows model metrics as functions of their number of group-expanded parameters; the right column shows model metrics as a function of their number of trainable parameters. The black vertical dashed line indicates the basis for comparison in Cohen & Welling (2016). Error bars indicate 95% confidence intervals for the mean; some are too tight to be visible.

Figure 3 shows the key metrics recorded for MNIST $O(2)$, with the dashed vertical line in the bottom-right plot showing the basis of comparison used in Cohen & Welling (2016). We observe that when networks are compared across varying

[5]We exchanged the strided spatial max-pooling layer at the second layer of the network with a strided spatial mean-pooling layer for ease of implementation of knot extraction.

[6]Note, because the size of the feature maps along the translation dimensions are constant between different group settings, we omit it as part of the group-expanded channels calculation. For example, we report the size of the group FlipRot90 as 8.

numbers of group-expanded parameters (GEP approach—left column), the trends in their complexity metrics follow similar patterns. In contrast, when networks are compared across varying numbers of trainable parameters (TP approach—right column), the trends for more-equivariant networks are accelerated, with more-equivariant networks using considerably more knots, being much less smooth, and minimizing the training error to a substantially greater degree. This suggests that Cohen & Welling's (2016) results reflect benefits not solely attributable to the enforced equivariance constraints, but also to the additional expressivity granted to their more-equivariant models.

Based on our insights from Figure 3, going forward, we will group networks using only the GEP approach. Figure 4 plots the proposed metrics for various models trained on the described tasks. Salient plots are highlighted in different colors for reference in the following discussion.

Number of Knots We observe that equivariance to different groups results in networks with the same number of group-expanded channels broadly having similar numbers of knots: more-equivariant networks usually have slightly more, despite their fewer trainable parameters. This difference is more pronounced for Downsampled MNIST, where the relative group orders vary greatly.

Uniformity Criterion Networks exhibit similar trends in the uniformity of their knot distributions, all of them becoming more uniform as the layers widen. While MNIST does show some differences, there do not appear to be substantial across-the-board differences in uniformity induced by equivariance to different groups.

Sensitivity We observe that more-equivariant networks exhibit larger expected gradient norms (with regard to their logits), indicating that these functions are more sensitive to local perturbations. This phenomenon may be related to the findings of (Gruver et al., 2023), who observed that CNNs—which are equivariant to integer translations but not continuous ones—were less equivariant to "small", *local*, translations than networks not explicitly constrained to be equivariant, such as vision transformers. (Gruver et al., 2023)'s findings could potentially be explained by the observation that G-CNNs have greater gradient norms in general, not just in the directions of infinitesimal group transformations. Novak et al. (2018) reported that smaller expected Jacobian norms were predictive of greater generalization. Our results serve as an exception to this: in our case, greater equivariance led to greater generalization (see Appendix B), despite greater expected gradient norms.

Smoothness Criterion More-equivariant networks had significantly larger smoothness criteria, indicating less smooth fitted functions, which is again most pronounced for Downsampled MNIST. Noticing that the trends for the expected gradient norm and smoothness criterion are similar, we plot the ratio of the square root of the smoothness criterion to the expected gradient norm in Figure 5. This plot no longer shows statistically significant differences between networks equivariant to different groups, suggesting that the increased smoothness criterion values in more-equivariant networks can largely be explained by their larger gradients.

Training Loss We have seen that more-equivariant networks generally exhibited greater complexity. Here we see that these changes typically also manifest in these models' improved capability to minimize the training error. A notable exception for Downsampled MNIST was that encoding height-wise translational equivariance was detrimental to performance at the smallest layer width setting. While this gap in performance vanished at larger layer widths, we additionally trained models equivariant to width-wise translational equivariance to further investigate the phenomenon. Figure 5 includes width-wise translation equivariant models in the training loss, and shows that, unlike height-wise translation, encoding width-wise translation equivariance improves performance even at the smallest layer width. This suggests that when expressivity is a bottleneck, imposing equivariance may hinder performance, but can be alleviated by selecting which group to enforce symmetries to.

4.2. Further Discussion

When group-expanded parameters were equated, more-equivariant networks exhibited characteristics typical of the same, or even slightly *more* flexibility—they usually had more knots, were less smooth, and were able to fit the training set to a greater degree, despite having considerably fewer trainable parameters. However, this increased flexibility cannot be attributed to an increase in the expressivity of equivariant models: a less-equivariant model with an equal number of group-expanded channels at every layer can perfectly emulate more-equivariant network by using appropriately selected weights. Rather, this improvement should be attributed to the reduced search space over which functions are fitted, and the potential impact this has on the optimization process during training.

We also evaluated other metrics for a more comprehensive comparison of the qualitative differences of networks equivariant to different groups. These results run in line with our main findings, and can be found in Appendix B.

5. Related Work

Equivariant Networks Equivariant neural networks have been developed for various discrete groups extending beyond translational symmetry on images, such as the sym-

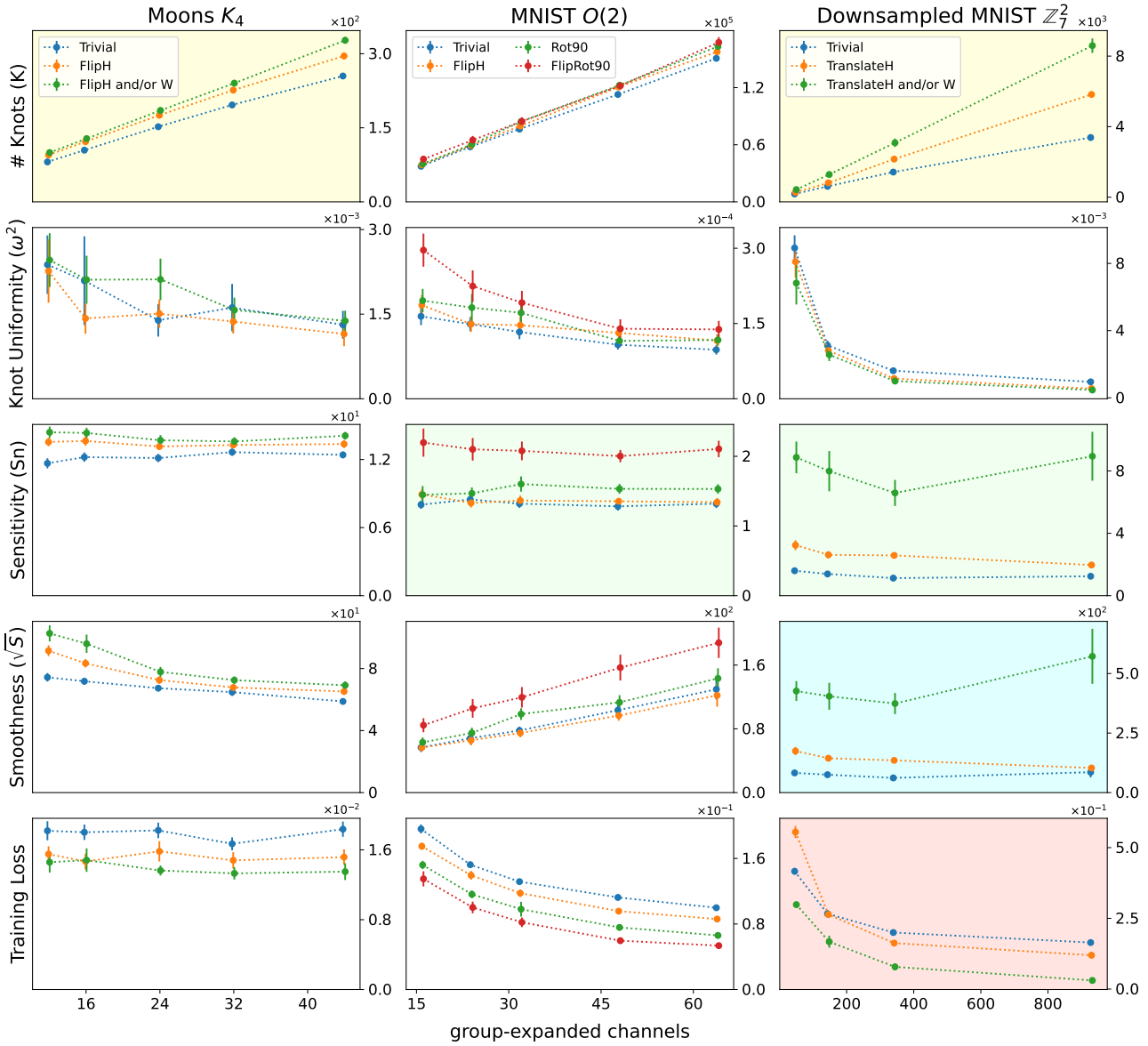

Figure 4. From top to bottom: the number of knots, uniformity criterion, sensitivity, square root of the smoothness criterion, and training loss of networks equivariant to various groups, over varying numbers of group-expanded channels for different datasets. Error bars indicate 95% confidence intervals for the mean; some are too tight to be clearly visible. Notice how similar trends are broadly displayed, regardless of the group to which the network is equivariant. Salient plots are highlighted in different colors and are discussed in the main body. Code for our experiments is available at https://github.com/ljroos/On-Fairly-Comparing-Group-Equivariant-Networks.

metries of the square (Cohen & Welling, 2016), hexagonal symmetries (Hoogeboom et al., 2018), and subgroups of $E(2)$ (Weiler & Cesa, 2019), all showing improved performance when number of trainable parameters fixed are kept fixed. Klee et al. (2023) also found equivariance to be beneficial for large-scale pre-training on ImageNet, but at a substantial computational cost when trainable parameters were matched. Similar to our work, both Weiler & Cesa (2019) and Klee et al. (2023) report on experiments where compute was matched in terms of group-expanded parameters. Weiler & Cesa (2019) found that more-equivariant networks could improve classification accuracy, even when compute was matched. However, this finding did not persist for the larger-scale tasks performed in Klee et al. (2023),

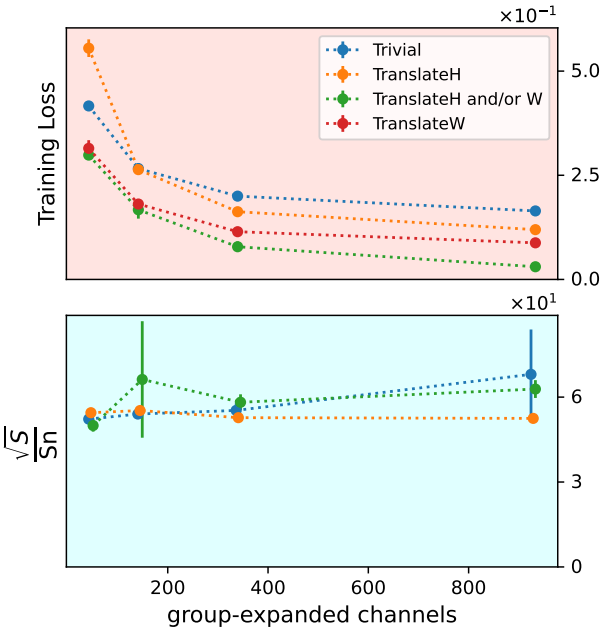

Figure 5. Additionally generated statistics for Downsampled MNIST $\mathbb{Z}_7^2$. Top: Training losses, with the group of width-wise cyclic translations additionally included. Bottom: The ratio of the square root of the smoothness criterion and the sensitivity. Background colors match corresponding plots in Figure 4.

where it was speculated that equivariant networks may still perform better on tasks with *equivariant* conditional label distributions (as opposed to invariant ones), as found by Gerken et al. (2022). Like Cohen & Welling (2016), we focus on extending CNNs to be equivariant to the symmetries of the square, as these are easily implementable and particularly relevant to images, without the complexities introduced by other, larger, groups.

Polytopal Complexity Montufar et al. (2014) derived theoretical bounds on the maximum number of affine regions in a ReLU feedforward network, but Hanin & Rolnick (2019) found that in practice, the number tends to be much lower. While recent work by Berzins (2023) proposed an improved algorithm for extracting the polytopal complex, they note that the problem remains computationally intractable for high-dimensional inputs. To mitigate the curse of dimensionality, Novak et al. (2018) and Hanin & Rolnick (2019) restrict the extraction to one- or two-dimensional subspaces. Our work follows their example, extracting the complex over one-dimensional curves, but instead uses a VAE to sample polylines near the data manifold.

G-CNN Expressivity Xiong et al. (2020) derived bounds on the number of affine regions of ReLU CNNs, and analytically showed that, when controlling for the number of trainable parameters, input dimension and number of layers, ReLU CNNS are asymptotically much more expressive per parameter than fully-connected ReLU networks, which our plots corroborate. Takai et al. (2021) showed that, despite clearly being less expressive, permutation-invariant networks have identical bounds on the number of affine regions as corresponding networks without weight tying, and thus proposed further grouping the affine regions into equivalence classes which can be counted. Lengyel & Gemert (2021) observed that trained G-CNN weights become correlated across the group dimension, and along with Knigge et al. (2022) made proposals to exploit this redundancy in order to ease the computational burden of equivariance to larger groups. In contrast, our work challenges the idea that equivariant networks inherently require a scaling of resources compared to their less-equivariant counterparts. Rather, more-equivariant networks may reap more benefits from additional compute, by better exploiting the extra expressivity granted by wider layers.

6. Conclusion

In this work, we have explored the flexibility of G-CNNs under different computational budgets, in the context of tasks with invariant input and conditional label distributions. We proposed multiple metrics for measuring the complexity of these networks, which allowed us to empirically analyze the impact of enforcing equivariance on network flexibility when keeping computational requirements fixed. Our results demonstrate that in this setting, G-CNNs broadly exhibit similar, or even slightly more, flexibility, regardless of the order of the group to which they are equivariant. These findings call into question the common practice of comparing G-CNNs by fixing the number of trainable parameters, which we argue unfairly grants more-equivariant additional expressivity. We emphasize that this does not invalidate any performance gains observed from enforcing equivariance; rather, it reframes the source of improvement attributable to utilizing equivariant models: encoding symmetries may improve performance, but, somewhat independently, more-equivariant models may also benefit from wider layers. Overall, this paper offers a new perspective on the impact of equivariance on model flexibility, suggesting how future work may more fairly compare architectures equivariant to different groups in a computationally relevant manner.

Acknowledgements

We thank DeepMind for financially supporting L. Roos through a scholarship. We also thank the anonymous reviewers for their helpful feedback, and Shane Josias for thoughtful discussions.

References

Balestriero, R. and Baraniuk, R. A spline theory of deep learning. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 374–383. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/balestriero18b.html.

Balestriero, R. and Baraniuk, R. G. Mad max: Affine spline insights into deep learning. *Proceedings of the IEEE*, 109 (5):704–727, 2021. doi: 10.1109/JPROC.2020.3042100.

Bartlett, P. L., Long, P. M., Lugosi, G., and Tsigler, A. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020. doi: 10.1073/pnas.1907378117. URL https://www.pnas.org/doi/abs/10.1073/pnas.1907378117.

Bekkers, E., Lafarge, M., Veta, M., Eppenhof, K., Pluim, J., and Duits, R. *Roto-Translation Covariant Convolutional Networks for Medical Image Analysis*, pp. 440–448. 09 2018. ISBN 978-3-030-00927-4. doi: 10.1007/978-3-030-00928-1_50.

Berzins, A. Polyhedral complex extraction from ReLU networks using edge subdivision. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2234–2244. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/berzins23a.html.

Cohen, T. and Welling, M. Group equivariant convolutional networks. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2990–2999, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/cohenc16.html.

Coles, S. *An introduction to statistical modeling of extreme values*. Springer Series in Statistics. Springer-Verlag, London, 2001. ISBN 1-85233-459-2.

Cramér, H. On the composition of elementary errors. *Scandinavian Actuarial Journal*, 1928(1):13–74, 1928. doi: 10.1080/03461238.1928.10416862. URL https://doi.org/10.1080/03461238.1928.10416862.

Dey, N., Chen, A., and Ghafurian, S. Group equivariant generative adversarial networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=rgFNuJHHXv.

Elesedy, B. and Zaidi, S. Provably strict generalisation benefit for equivariant models. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2959–2969. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/elesedy21a.html.

Gerken, J., Carlsson, O., Linander, H., Ohlsson, F., Petersson, C., and Persson, D. Equivariance versus augmentation for spherical images. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 7404–7421. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/gerken22a.html.

Gruver, N., Finzi, M. A., Goldblum, M., and Wilson, A. G. The Lie derivative for measuring learned equivariance. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=JL7Va5Vy15J.

Hanin, B. and Rolnick, D. Complexity of linear regions in deep networks. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2596–2604. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/hanin19a.html.

Hoogeboom, E., Peters, J. W., Cohen, T. S., and Welling, M. HexaConv. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1vuQG-CW.

Klee, D., Park, J. Y., Platt, R., and Walters, R. A comparison of equivariant vision models with ImageNet pretraining. In *NeurIPS 2023 Workshop on Symmetry and Geometry in Neural Representations*, 2023. URL https://openreview.net/forum?id=3ItzNHPov9.

Knigge, D. M., Romero, D. W., and Bekkers, E. J. Exploiting redundancy: Separable group convolutional networks on Lie groups. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 11359–11386. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/knigge22a.html.

Kondor, R. and Trivedi, S. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine*

Learning, volume 80 of *Proceedings of Machine Learning Research*, pp. 2747–2755. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/kondor18a.html.

Lafarge, M. W., Bekkers, E. J., Pluim, J. P., Duits, R., and Veta, M. Roto-translation equivariant convolutional networks: Application to histopathology image analysis. *Medical Image Analysis*, 68:101849, 2021. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2020.101849. URL https://www.sciencedirect.com/science/article/pii/S1361841520302139.

Lengyel, A. and Gemert, J. v. Exploiting learned symmetries in group equivariant convolutions. In *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 759–763, 2021. doi: 10.1109/ICIP42928.2021.9506362.

Montufar, G. F., Pascanu, R., Cho, K., and Bengio, Y. On the number of linear regions of deep neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/109d2dd3608f669ca17920c511c2a41e-Paper.pdf.

Novak, R., Bahri, Y., Abolafia, D. A., Pennington, J., and Sohl-Dickstein, J. Sensitivity and generalization in neural networks: an empirical study. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HJC2SzZCW.

Takai, Y., Sannai, A., and Cordonnier, M. On the number of linear functions composing deep neural network: Towards a refined definition of neural networks complexity. In Banerjee, A. and Fukumizu, K. (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 3799–3807. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/takai21a.html.

Walters, R., Li, J., and Yu, R. Trajectory prediction using equivariant continuous convolution. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=J8_GttYLFgr.

Weiler, M. and Cesa, G. General E(2)-equivariant steerable CNNs. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/45d6637b718d0f24a237069fe41b0db4-Paper.pdf.

Xiong, H., Huang, L., Yu, M., Liu, L., Zhu, F., and Shao, L. On the number of linear regions of convolutional neural networks. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10514–10523. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/xiong20a.html.

Zhu, X., Wang, D., Biza, O., Su, G., Walters, R., and Platt, R. Sample efficient grasp learning using equivariant models. In *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, June 2022. doi: 10.15607/RSS.2022.XVIII.071.

A. Uniformity Criterion Discussion

A.1. Cramér–von Mises Test

The Cramér–von Mises test is a non-parametric statistical test that allows one to test the hypothesis that a sample of K observations, $t_1, \ldots, t_K$, listed in ascending order, comes from a given reference distribution. To do so, it uses the Cramér–von Mises criterion, which we now state in full:

$$\omega^2 = \int \left(F(t) - U(t)\right)^2 dU(t). \tag{3}$$

Here, U is the CDF of the reference distribution to be tested against, which we selected to be the uniform distribution for our uniformity criterion in Equation (2). F is set as the ECDF of the sampled observations, given by

$$\text{ECDF}(t) = \frac{1}{K} \sum_{k=1}^{K} I(t_k \leq t). \tag{4}$$

For some applications using the ECDF, the number of observations plus one is used in the denominator (Coles, 2001). We refer to this as the adjusted ECDF, given by

$$\text{ECDF}_{\text{adj}}(t) = \frac{1}{K+1} \sum_{k=1}^{K} I(t_k \leq t). \tag{5}$$

A.2. Piecewise Linear Distribution Estimator

Because we do not intend to use the criterion in Equation (3) as a test statistic for a hypothesis test, but rather as a tool to measure the complexity of the distribution of knots, we are not restricted to fitting F using the ECDF in Equation (4). Rather, with a few simple modeling assumptions, we improve on the ECDF to take into account the boundaries of the interval $[0, L]$ on which the true CDF is supported.

Although the ECDF in Equation (4) is an unbiased estimate of the true CDF, its harsh inductive bias may not be ideal for our task. To make the harshness of its inductive biases concrete, $\text{ECDF}(t) = 0$ if $t < t_1$, and $\text{ECDF}(t) = 1$ if $t > t_K$. Intuitively stated, according to a fitted ECDF, it is impossible for a test point to lie outside the range given by the minimum and maximum training observations.

Defining $t_0 := 0$ and $t_{K+1} := L$, the interval $(0, L]$ can be partitioned into $K+1$ intervals, $\{(t_k, t_{k+1}] : 0 \leq k \leq K\}$. We now derive a piecewise linear distribution estimator, based on two inductive biases: firstly, that a test point, T, is equally likely to lie within any one of the $K+1$ intervals, i.e. $P(t_k < T \leq t_{k+1}) = \frac{1}{K+1}$, for all $0 \leq k \leq K$; and secondly, that the distribution within an interval is uniform i.e. the rate of change of the CDF is constant between any two consecutive training observations.

Using the first inductive bias we deduce

$$\begin{aligned} F(t_k) &= P(T \leq t_k) \\ &= \sum_{i=1}^{k} P(t_{i-1} < T \leq t_i) \\ &= \frac{k}{K+1}. \end{aligned}$$

Considering Equation (5), we see that $F(t_k) = \text{ECDF}_{\text{adj}}(t_k)$, except when $t > t_k$. Our estimator can thus be seen as a kind of piecewise linear modification of the adjusted ECDF, which linearly interpolates the adjusted ECDF between training observations. For conciseness, we define the following.

$$
\begin{aligned}
F_k &:= F(t_k), \\
d_k &:= t_{k+1} - t_k, \\
M &:= \frac{1}{K+1}, \\
m_k &:= \frac{M}{d_k}.
\end{aligned}
$$

From the second inductive bias, it follows that $\frac{dF(t)}{dt} = m_k$ for $t_k < t \leq t_{k+1}$. Thus, we arrive at our piecewise linear distribution estimator,

$$
F(t) = F_k + m_k \left(t - t_k\right), \text{ for } t_k \leq t \leq t_{k+1}. \tag{6}
$$

A.3. Uniformity Criterion Calculation

Suppose then we wish to compute the uniformity criterion. Substituting our piecewise linear estimator from Equation (6) into the uniformity criterion given in Equation (2), we get

$$
\begin{aligned}
\omega^2\left(F\right) =& \frac{1}{L} \int_0^L \left(F_k + m_k \left(t - t_k\right) - \frac{t}{L} \right)^2 dt \\
=& \frac{1}{L} \sum_{k=0}^{K} \int_{t_k}^{t_{k+1}} \left(F_k + m_k \left(t - t_k\right) - \frac{t}{L} \right)^2 dt,
\end{aligned}
$$

splitting the criterion into $K+1$ parts which can be computed individually and summed together at the end. We expand each individual component further, for brevity defining terms along the way using the walrus operator, ":=", similar to its use in the Python programming language,

$$
\begin{aligned}
I_k :&= \int_{t_k}^{t_{k+1}} \left(F_k + m_k \left(t - t_k\right) - \frac{t}{L} \right)^2 dt \\
&= \int_{t_k}^{t_{k+1}} \left(\left(A_k := F_k - m_k t_k\right) + \left(B_k := m_k - \frac{1}{L}\right) t \right)^2 dt \\
&= \int_{t_k}^{t_{k+1}} \left(A_k + B_k t\right)^2 dt \\
&= \int_{t_k}^{t_{k+1}} \left(A_k^2 + 2A_k B_k t + B_k^2 t^2\right) dt \\
&= \left[A_k^2 t + A_k B_k t^2 + \frac{1}{3} B_k^2 t^3 \right]_{t_k}^{t_{k+1}} \\
&= A_k^2 \left(d_k^{(1)} := t_{k+1} - t_k\right) + A_k B_k \left(d_k^{(2)} := t_{k+1}^2 - t_k^2\right) + \frac{1}{3} B_k^2 \left(d_k^{(3)} := t_{k+1}^3 - t_k^3\right) \\
&= A_k^2 d_k^{(1)} + A_k B_k d_k^{(2)} + \frac{1}{3} B_k^2 d_k^{(3)}.
\end{aligned}
$$

While this expansion provides a neat algebraic formulation in terms of A_k, B_k and $d_k^{(\cdot)}$, naively computing I_k using this expansion may suffer numerical instability when $d_k^{(1)}$ becomes very small. Alternatively, one can use a midpoint approximation to the integral, $\int_a^b g(t)dt \approx g\left(\frac{a+b}{2}\right)(b-a)$:

$$\int_{t_k}^{t_{k+1}} \left(F_k + m_k\left(t - t_k\right) - \frac{t}{L}\right)^2 dt \approx \left(F_k + m_k\left(\frac{t_k + t_{k+1}}{2} - t_k\right) - \frac{t_k + t_{k+1}}{2L}\right)^2 \left(t_{k+1} - t_k\right)$$
$$= \left(F_k + \frac{M}{2} - \frac{t_k + t_{k+1}}{2L}\right)^2 d_k^{(1)}.$$

A.4. Entropy Criterion

We additionally quantify the complexity of the fit piecewise linear distribution F using its entropy. The entropy of F is similar to the uniformity criterion, in that both are optimized when F is uniform—the entropy is maximized and the uniformity criterion is minimized. Noting that the density function of F is piecewise constant, given by $f(t) = m_k = \frac{M}{d_k}$ for $t_k < t \leq t_{k+1}$, the entropy of F is given by

$$\begin{aligned}
\mathbb{H}(F) &= -\mathbb{E}_{T\sim f}\left[\log f\left(T\right)\right] \\
&= -\int f(t)\log f(t)dt \\
&= -\sum_{k=0}^{K}\int_{t_k}^{t_{k+1}} \left(\frac{M}{d_k}\right)\log\left(\frac{M}{d_k}\right)dt \\
&= -\sum_{k=0}^{K}\left(\frac{M}{d_k}\right)\log\left(\frac{M}{d_k}\right)\int_{t_k}^{t_{k+1}} dt \\
&= -\sum_{k=0}^{K} M\log\left(\frac{M}{d_k}\right) \\
&= M\sum_{k=0}^{K}\left(\log d_k - \log M\right) \\
&= \frac{1}{K+1}\sum_{k=0}^{K}\left(\log d_k + \log\left(K+1\right)\right) \\
&= \log\left(K+1\right) + \frac{1}{K+1}\sum_{k=0}^{K}\log\left(t_{k+1} - t_k\right).
\end{aligned}$$

B. Further Results

We report additional statistics to supplement Figure 4 in Figure 6. Included, is the additional entropy criterion for the knot distribution from Appendix A.4, and its behavior follows a similar trend to the uniformity criterion's. We also add the square root smoothness criterion normalized by sensitivity for all datasets, as we did for Downsampled MNIST in Figure 5. Finally, we include statistics for the test loss of the network: the test loss itself is included, and correlates well with the training loss, though is slightly higher than the training loss, as one would expect. The amount by which the test loss is higher is quantified by the generalization gap = test loss − training loss.

Entropy Criterion The entropy criterion behaves similarly to the uniformity criterion—both roughly indicating that the knot distribution becomes more uniform as layers widen.

Sensitivity Normalized Smoothness $\frac{\sqrt{S}}{\mathrm{Sn}}$ As initially observed for Downsampled MNIST in Figure 5, normalizing the square root smoothness by the sensitivity removes the statistically significant differences in smoothness between networks equivariant to different groups.

Test Loss and Generalization Gap As one would expect, our more-equivariant networks do generalize better than their less-equivariant counterparts, as evidenced by their lower test errors. Interestingly, while more-equivariant networks typically attain lower test errors, their generalization gaps are usually *larger*. This occurrence is known as *benign* overfitting (Bartlett

et al., 2020)—the phenomenon where a model overfits to the training data, but still generalizes well to unseen test data. The presence of benign overfitting corroborates the validity of our complexity metrics as proxies for model flexibility, as it associates the increased fitted function complexity of more-equivariant networks with a kind of overfitting.

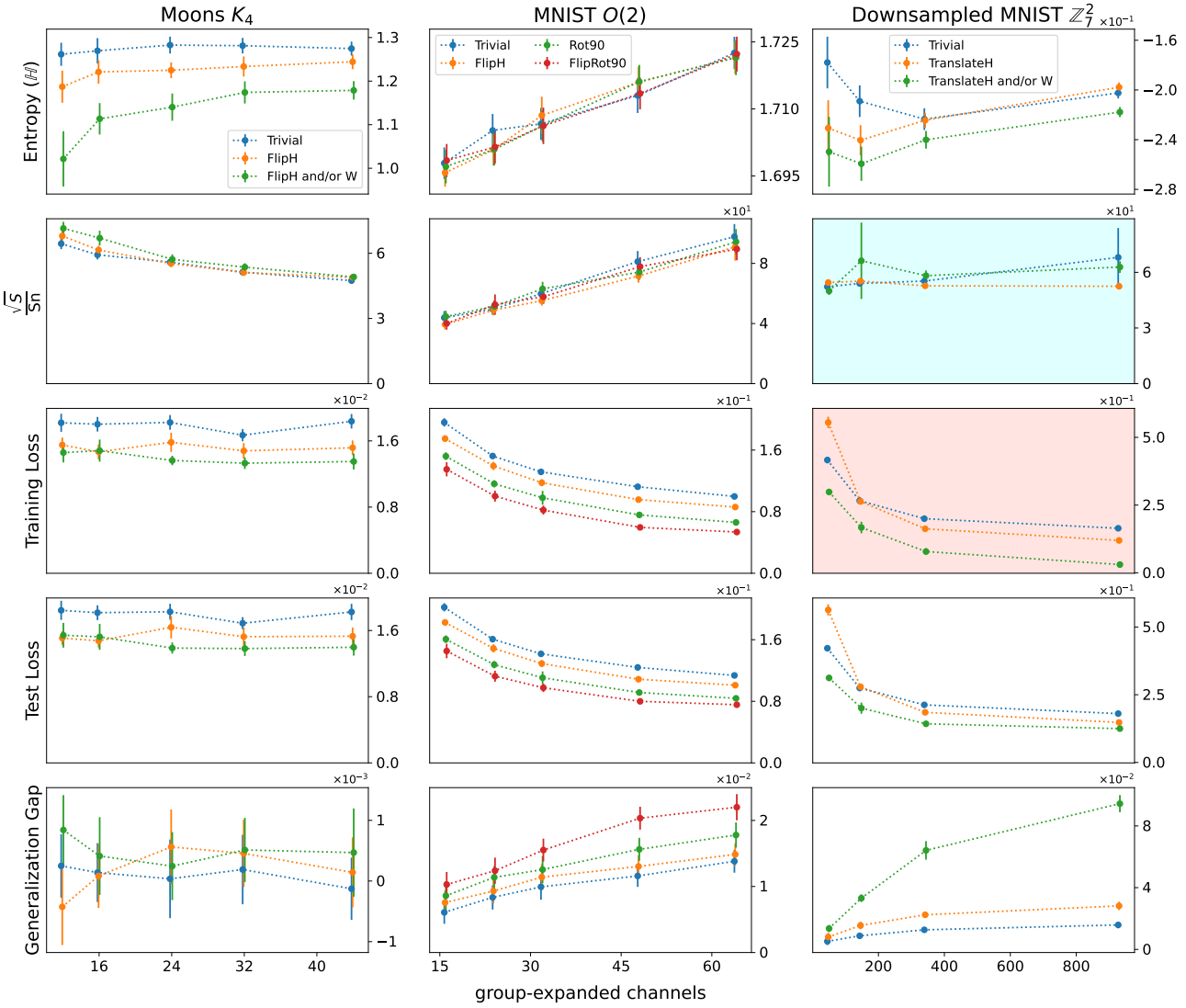

Figure 6. Additional statistics to supplement Figure 4, with the training loss intentionally repeated to allow for comparison with test loss, and matching colors for repeated plots. From top to bottom: Knot entropy, square root smoothness normalized by sensitivity, training loss, test loss, and generalization gap (test loss − training loss) of networks equivariant to various groups, over varying numbers of group-expanded channels for different datasets. Error bars indicate 95% confidence intervals for the mean; some are too tight to be clearly visible. Notice how similar trends are broadly displayed, regardless of the group to which the network is equivariant.