Continuous Rotation Group Equivariant Network Inspired by Neural Population Coding

Zhiqiang Chen[1,2], **Yang Chen**[2], **Xiaolong Zou**[3], **Shan Yu**[2,4*]
[1]Beijing Academy of Artificial Intelligence, Beijing, China
[2]Laboratory of Brain Atlas and Brain-inspired Intelligence, Institute of Automation(CASIA), Beijing, China
[3]Qiyuan Lab, Beijing, China
[4]School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), Beijing, China
chenzhiqiang@mails.ucas.ac.cn,yang.chen@ia.ac.cn,xiaolz@mail.pku.edu.cn,shan.yu@nlpr.ia.ac.cn

Abstract

Neural population coding can represent continuous information by neurons with a series of discrete preferred stimuli, and we find that the bell-shaped tuning curve plays an important role in this mechanism. Inspired by this, we incorporate a bell-shaped tuning curve into the discrete group convolution to achieve continuous group equivariance. Simply, we modulate group convolution kernels by Gauss functions to obtain bell-shaped tuning curves. Benefiting from the modulation, kernels also gain smooth gradients on geometric dimensions (e.g., location dimension and orientation dimension). It allows us to generate group convolution kernels from sparse weights with learnable geometric parameters, which can achieve both competitive performances and parameter efficiencies. Furthermore, we quantitatively prove that discrete group convolutions with proper tuning curves (bigger than 1x sampling step) can achieve continuous equivariance. Experimental results show that 1) our approach achieves very competitive performances on MNIST-rot with at least 75% fewer parameters compared with previous SOTA methods, which is efficient in parameter; 2) Especially with small sample sizes, our approach exhibits more pronounced performance improvements (up to 24%); 3) It also has excellent rotation generalization ability on various datasets such as MNIST, CIFAR, and ImageNet with both plain and ResNet architectures.

Introduction

Deep neural networks (DNNs) have exhibited great potential in the past decade, achieving excellent performance on a variety of visual tasks, including classification(Lin, Chen, and Yan 2013), detection(Girshick 2015; Ren et al. 2015; He et al. 2017), segmentation(Kamnitsas et al. 2016; Chen et al. 2014; Ronneberger, Fischer, and Brox 2015), etc. Compared with fully connected networks(Hsu, Li, and Psaltis 1990), convolutional neural networks(CNN)(Lecun et al. 1998; Krizhevsky, Sutskever, and Hinton 2012; He et al. 2016) have intrinsic decoupling ability for translation, which allows the network to learn essential features regardless of the position. Besides the parameter efficiency by sharing weights in different locations, CNN also reduces the latent learning space due to the built-in translation symmetry. In order to take advantage of more symmetries and further improve the decoupling ability of the network, the group equivariant theory and the group equivariant CNN (G-CNN) were proposed (Cohen and Welling 2016). G-CNN constructed rotation group convolutions by rotating the convolution kernels to multiple orientations, thereby establishing the intrinsic rotation symmetry.

The manner of group convolution is similar to neural population coding(Averbeck, Latham, and Pouget 2006) that represents information by a group of neurons with a series of different preferred stimuli. For example, the 'place cells', neurons of orientation, color and direction in visual cortical areas, and many other neurons all encode the information in a population coding manner(Borst and Theunissen 1999; Usrey and Reid 1999; Zemel, Dayan, and Pouget 1998).

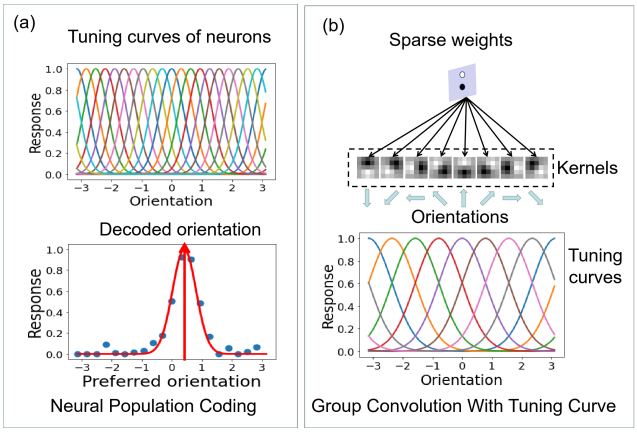


Figure 1: Population coding. (a) A standard model of neural population coding(Pouget, Dayan, and Zemel 2000). Upper figure shows the tuning curves of 20 neurons with different preferred orientations. Blue points in the bottom figure are responses of the 20 neurons to an orientation of 0.15π. The red solid line is the posterior distribution of the orientation (Gauss fitting) estimated from the responses of neurons. The red arrow is the decoded orientation, which is in continuous space. (b) Group convolution with bell-shaped tuning curve. Solid lines are the orientation tuning curves of the kernels in group convolution. With proper tuning curves, the group convolution can achieve continuous equivariance.

*Corresponding author: Shan Yu

Due to the practical limitation, the preferred stimuli of neurons and group convolution are both in discrete space, while the information to encode is usually in continuous space. In group equivariant networks, some works adopted interpolation approximation to achieve continuous equivariance, such as Taylor expansion(Shen et al. 2020) and B-spline(Bekkers 2019). However, these methods usually constructed elaborate basic kernels(Weiler, Hamprecht, and Storath 2018; Weiler and Cesa 2019), and some(Shen et al. 2020) were under the assumption that the input is smooth. In biological nervous system, there is a simple and robust mechanism called population coding to solve this problem. Figure 1(a) demonstrates a standard model of neural population coding for orientation(Pouget, Dayan, and Zemel 2000). The left figure shows the bell-shaped tuning curves of 20 neurons with different preferred orientations. In the right figure, the blue points are the responses of the 20 neurons. The red solid line shows the Gauss fitting of the responses, and the peak (red arrow) is the decoded orientation. Though the preferred orientations are discrete, the population pattern with proper tuning curves can robustly encode and decode information in continuous space.

Inspired by the population coding, we propose a novel group convolution with bell-shaped tuning curves. As Figure 1(b) shows, we use a set of sparse weights with learnable geometric parameters to generate group convolution kernels, which has great parameter efficiency. At the end of the generation, a Gaussian modulation is adopted to generate kernels with bell-shaped tuning curves. We prove that discrete group convolutions can achieve continuous equivariance with proper tuning curves (bigger than $1\times$ sampling step). And we quantitatively analyze the equivariant errors with different widths of tuning curves. Practically, we construct Continuous Rotation Group Equivariant Networks (CRGEN), and experimental results show that the proposed CRGEN can achieve very competitive performance with less than 25% parameters compared with previous SOTA rotation group equivariant networks on MNIST-rot. Furthermore, CRGEN exhibited excellent rotation generalization and small sample learning abilities on multiply datasets(e.g. MNIST, CIFAR, ImageNet) with various network architectures(i.e. plain and ResNet architectures).

Related Works

There are plenty of explorations of invariance in the field of computer vision such as ScatNet(Bruna and Mallat 2013), PCANet(Chan et al. 2015), and TIRBM(Sohn and Lee 2012). In some works, networks achieved rotation-invariant by data augmentation during the training phase(Xi et al. 2018; Luo et al. 2020). However, these methods mainly paid attention to invariant features. By decoupling the transformation from the object, equivariant networks(Cohen and Welling 2016) can preserve both invariant features and the information of the transformation itself. In addition to the translation equivariance of CNN, equivariant networks on more groups such as rotating groups(Cohen and Welling 2016; Dieleman, De Fauw, and Kavukcuoglu 2016; Cheng et al. 2018), scaling groups(Worrall and Welling 2019; Sifre and Mallat 2013), mirror groups(Cohen and Welling 2016), and general Lie group(Finzi et al. 2020) have been proposed. G-CNN(Cohen and Welling 2016) proposed a group convolution framework and designed the rotation group equivariant network. Due to the limitation of sampling lattices, only some specific symmetries of convolutional kernel such as 4 or 6 orientation symmetries(Cohen and Welling 2016; He et al. 2021; Hoogeboom et al. 2018) can be achieved directly. To exploit more symmetries, SFCNN(Weiler, Hamprecht, and Storath 2018), E2CNN(Weiler and Cesa 2019) and PDO-eConvs(Shen et al. 2020) designed some basic kernels to generate the group convolution kernels. PDO-eConvs and B-Spline CNN (Bekkers 2019) achieved continuous equivariances by interpolation methods such as Taylor expansion and B-spline approximation. SWSL(Chen et al. 2022) replaced shallow layers of CNN by the group convolution, and achieved better performances with much fewer parameters and no extra computations in shallow layers. Furthermore, E^4-Net(He et al. 2021) can achieve both parameter and computational efficiencies. Besides equivariant convolutions, some works also studied the equivariance of subsampling(Xu et al. 2021) and nonlinearity(Franzen and Wand 2021) layer. Recently, equivariant networks were applied to much more domains such as graph CNN(Puny et al. 2021), reinforcement learning(Wang, Walters, and Platt 2022), protein docking(Ganea et al. 2021), RNA structure prediction(Townshend et al. 2021), and so on, which showed a great potential of the equivariant networks.

Prior Knowledge of Group Equivariant Networks

Group equivariance is defined as (Cohen and Welling 2016; Shen et al. 2020):

$$\Phi[T_g[f]] = T'_g[\Phi[f]], g \in G, \tag{1}$$

where T_g is a transformation operation with parameter $g \in G$, f is inputs and Φ is a general feature mapping function. Equivariance means that transforming the inputs before feature extracting is equal to transforming the outputs after feature extracting. Invariance is a special case of equivariance, where T'_g is identity. Compared with invariance, equivariance disentangles the invariant features while remaining the transforming information. G-CNN proposed a general method to construct group equivariant convolutions(Cohen and Welling 2016). Figure2 (a) and (b) show the rotation group convolutions. For the input image, group convolution is obtained by several rotations of the same kernel as shown in Figure 2 (a). For the convolution of the hidden layer, group convolution is obtained by several rotations and cyclic shifts as shown in Figure 2 (b). It can be summarized in a general mathematical form (Finzi et al. 2020). Conveniently, we use a function style representation to denote images, feature maps and kernels rather than a matrix style. For example, the input image is denoted as $y = f(x)$, where x is the location of a pixel and y is the value of this pixel. Formally, let $f : \mathbb{Z}^2 \to \mathbb{R}$ and $h_l : \mathbb{Z}^2 \times G \to \mathbb{R}^{C_l}$ be the input images and the lth layer feature maps with C_l channels. Let $\psi_0 : \mathbb{Z}^2 \to \mathbb{R}^{C_1}$ and $\psi_l : \mathbb{Z}^2 \times G \to \mathbb{R}^{C_{l+1} \times C_l}$ be the convolution kernels in the lth layer. To simplify it, we

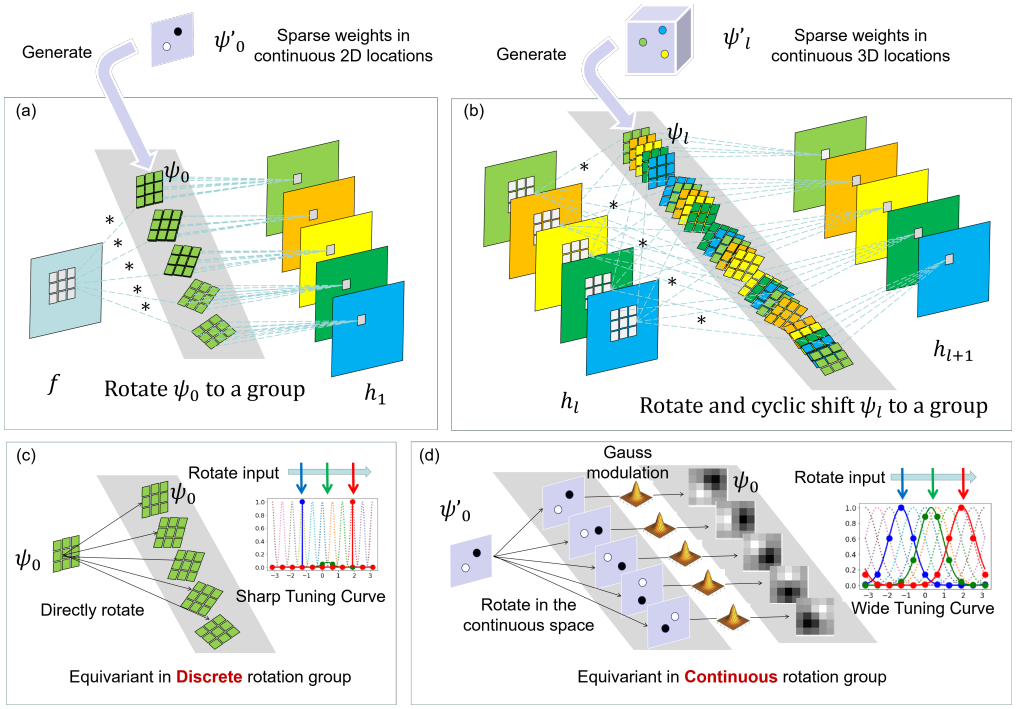


Figure 2: Construction of group equivariant convolution. (a) The group convolution for the input image. The group convolution kernels are obtained by several rotations for the same kernel. (b) Group convolutions for hidden layers. The group convolution kernels are obtained by several rotations and cyclic shifts for the same kernel. In our approach, the kernels are generated from a set of sparse weights in continuous location space. (c) shows that the group convolution kernels are obtained by directly rotating the kernel ψ_0. Each kernel has a sharp tuning curve, so it's not equivariant when the rotation degree isn't exactly on the discrete rotation group. (d) shows the rotation group convolution kernels are generated by ψ'_0, which is defined on the continuous space, and are modulated by Gauss kernel. So the kernel can be rotated without the limitation of discrete mesh grid, and has a tuning curve over the rotation. With wide tuning curve, the kernel group can obtain continuous rotation group equivariance by population encoding.

set $C_l = 1$, which does not influence the equivariance. Then for any $g \in G = \{i\frac{2\pi}{n} | i = 0, 1, 2, ..., n-1\}$, the group convolution on G is given by:

$$\begin{aligned} h_1(x, g) &= [f \circledast \psi_0](x, g) = f(x) * \psi_0(R_g^{-1}x) \\ &= \sum_{x' \in \mathbb{Z}^2} f(x')\psi_0(R_g^{-1}(x - x')), \\ h_{l+1}(x, g) &= [h_l \circledast \psi_l](x, g) \\ &= \sum_{g' \in G} h_l(x, g') * \psi_l(R_g^{-1}x, g - g') \\ &= \sum_{g' \in G} \sum_{x' \in \mathbb{Z}^2} h_l(x', g')\psi_l(R_g^{-1}(x - x'), g - g'), \\ R_g &= \begin{pmatrix} \cos(g) & \sin(g) \\ -\sin(g) & \cos(g) \end{pmatrix}, \end{aligned} \tag{2}$$

where $\circledast$ is the group equivariant convolution, $*$ is conventional convolution, and R_g is a rotation matrix with degree $g \in G$. Here, x denotes the location, and g denotes the degree, which can be regarded as the index of each element in a matrix. In equation 2, rotation and the cyclic shift are formulated as $R_g^{-1}x$ and $g - g'$, respectively. In addition to rotation and mirror group equivariances in G-CNN, Eq. 2 can also be applied to many other groups such as scaling, shearing, the mix of scaling and shearing and 3D rotation(under homogeneous coordinates) transformation groups, provided that the group satisfies $R_{g_1}R_{g_2}x = R_{g_1+g_2}x$(Cohen and Welling 2016). See detailed proofs of the equivariance in Appendix[1] A.

Continuous Group Equivariant Convolution

Conventional group equivariant convolution usually adopted interpolation approximation to achieve continuous equivariance. According to neural population coding, neurons with discrete preferred stimuli with proper tuning curves can achieve continuous equivariance as shown in Figure 1 (a). Inspired by neural population coding, we propose a method to incorporate bell-shaped tuning curves into the group convolution. In this section, we will show how to construct the group convolution with tuning curves and how to achieve continuous equivariance.

[1]https://gitee.com/chenzq/aaai_2024_appendix.git

Group Convolutions with Tuning Curves

In Figure 2 (a) and (b), we show the general framework to construct the group equivariant convolutions. Based on this framework, we use a set of sparse parameters to generate the group convolution kernels. Figure 2 (d) shows the details to generate the group convolution kernels with tuning curves. Specifically, denote the sparse parameters on continuous space as ψ' and denote the generated kernels on discrete mesh grid as ψ. Similar to Eq. 2, we represent them as functional styles:

$$\begin{aligned} &\psi'_0(x_i) = w_i, \qquad \psi_0(x) = \sum_i w_i \zeta(x - x_i) \\ &\psi'_l(x_j, g_j) = w_j, \quad \psi_l(x, g) = \sum_j w_j \zeta'(x - x_j, g - g_j) \end{aligned} \tag{3}$$

where x and g are location and orientation mesh grids on the discrete space, respectively, and x_i, x_j and g_j are location and orientation parameters defined in continuous space. $i \in \{1, 2, ..., m_0\}$, and $j \in \{1, 2, ..., m_l\}$, where m_0 and m_l are numbers of the sparse weights for ψ_0 and ψ'_l, respectively. Here, ζ and ζ' are the Gaussian modulation function as:

$$\begin{aligned} &\zeta(x - x') = \frac{1}{N} e^{-\frac{||x-x'||^2}{2\sigma_x^2}}, N = \sum_x e^{-\frac{||x-x'||^2}{2\sigma_x^2}}, \\ &\zeta'(x - x', g - g') = \frac{1}{N'} e^{-\frac{||x-x'||^2}{2\sigma_x^2}} e^{-\frac{||g-g'||^2}{2\sigma_g^2}}, \\ &N' = \sum_x \sum_g e^{-\frac{||x-x'||^2}{2\sigma_x^2}} e^{-\frac{||g-g'||^2}{2\sigma_g^2}}, \end{aligned} \tag{4}$$

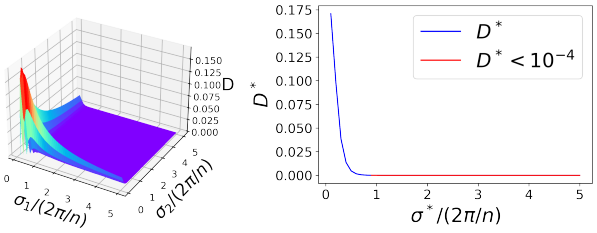
where $x \in \{(i, j)|i, j \in [-s, s] \in \mathbb{Z}\}, x' \in \mathbb{R}^2, g \in \{i\frac{2\pi}{n}|i = 0, 1, 2, ..., n-1\}, g' \in [0, 2\pi)$. x' and g' are the continuous locations and orientations to encode, σ_x and σ_g are the standard deviations, and N and N' are normalization factor. By the Gaussian modulation, each kernel has a tuning curve and each weight w_j has optimizable location x_j and orientation g_j. In addition, the transformation on x and g can be converted to x_j and g_j, which are in the continuous space (see details in appendix B).

Continuous Group Equivariance

In population coding, population neurons can't decode continuous stimulus well if the tuning curve is too sharp. Here, we quantitatively measure the relationship between the standard deviation of the tuning curve and the equivariance of group convolution. Group convolution can be regarded as the composition of several equivariant convolutions of ζ' (or ζ). So we define the error D of the equivariance as the following:

$$D(\sigma_1, \sigma_2, r) = \frac{2||[[L'_r\zeta'_1] \circledast \zeta'_2](0, g) - [L'_r[\zeta'_1 \circledast \zeta'_2]](0, g)||}{||[[L'_r\zeta'_1] \circledast \zeta'_2](0, g)|| + ||[L'_r[\zeta'_1 \circledast \zeta'_2]](0, g)||}, \tag{5}$$

where $r \in [0, 2\pi), g \in G$. σ_1 and σ_2 are the standard deviations of ζ'_1 and ζ'_2 on orientation dimension, respectively. L'_r is the rotation transformation, and $[L'_r\zeta'](x, g) = \zeta'(R_r^{-1}x, g - r)$. $||\cdot||$ is the length of vector. Figure 3 (a) shows the maximum error as σ_1 and σ_2 vary. It shows that the maximum error is small when both σ_1 and σ_2 are large. In figure 3 (b), σ^* is the minimum of σ_1 and σ_2 and D^* is the maximum of $D(\sigma_1, \sigma_2, r)$. When $\sigma^* > 2\pi/n$ ($1\times$ of the sampling step), $D^* < 10^{-4}$. We can neglect the error when both σ_1 and σ_2 are bigger than $1\times$ of the sampling step. In a conclusion, when the standard deviation of the tuning curve is proper, group convolution can achieve continuous equivariance, because group convolution is the composition of the convolutions of ζ' (or ζ) (detailed proof shown in Appendix C). It holds true not only in the rotation group but also in the translation group and other groups.



(a) (b)

Figure 3: (a) shows the maximum equivariant error vs. σ_1 and σ_2, where σ_1 and σ_2 are the standard deviations of the tuning curves. (b) shows the maximum equivariant error vs. σ^*, where σ^* is the minimum of σ_1 and σ_2.

Experiments

We build Continuous Rotation Group Equivariant Networks (CRGEN) by the group convolutions with tuning curves. Firstly, we test the rotation equivariance in common used benchmark MNIST-rot (Larochelle et al. 2007). Secondly, we test the small sample learning ability on MNIST-rot. Finally, we test the rotation generalization ability on MNIST, CIFAR, and ImageNet. See more details about datasets on appendix D.

Model

CRGEN-T:CRGEN-T is a tiny network containing 6 layers with 10 kernels each layer, which is similar to G-CNN (Cohen and Welling 2016). Each kernel has 16 orientation with 3 sparse points (i.e., $m_l = 3$ in Eq. 4), which has trainable locations and orientations. The generated kernels have a spatial size of 5×5.

CRGEN-S & CRGEN-L: CRGEN-S use lager networks with 8,8,16,16,16,32,32 kernels for each layer, and CRGEN-L has double kernels compared to. Each kernel has a spatial size of 3×3 with fixed location and trainable orientation parameters. More details about models and training setting are on appendix D.

CRGEN-LN: For deeper networks, the stacks of the proposed group convolution will cause blurring problem. To address this problem, we use pair-wise learning manner and add an equivariant loss to drive the whole network to learn the continuous equivariance (CRGEN-LN). Figure 5 shows the architecture of CRGEN-LN. We use the discrete rota-

Models	Err. (%)	Para.
ORN-8 (Zhou et al. 2017)	2.25	0.53M
TI-Pooling (Laptev et al. 2016)	2.2	13.3M
CNN (Cohen and Welling 2016)	5.03	22k
G-CNN (Cohen and Welling 2016)	2.28	25k
PDO-eConv (Shen et al. 2020)	1.87	26k
E^4-Net (He et al. 2021)	1.17	41.4K
SFCNN (Weiler, Hamprecht, and Storath 2018)	0.88	6.5M
CRGEN-T	1.16	6.2K
CRGEN-S	**0.85**	40K

Table 1: Errors on MNIST-rot without rotation augmentation.

Models	Err. (%)	Para.
H-Net (Worrall et al. 2017)	1.69	-
TIPooling (Zhou et al. 2017)	1.54	-
RotEqNet (Marcos et al. 2017)	1.09	-
PTN-CNN (Esteves et al. 2018)	0.89	-
E2CNN (Weiler and Cesa 2019)	0.716	6.5M
SFCNN (Weiler, Hamprecht, and Storath 2018)	0.714	6.5M
PDO-eConv (Shen et al. 2020)	0.709	650K
CRGEN-S	0.714	40K
CRGEN-L	**0.706**	150K

Table 2: Errors on MNIST-rot with rotation augmentation.

tion group equivariant convolution G-CNN as the backbone, and use the proposed continuous rotation group equivariant convolution on the final layer. Because the backbone is G-CNN, the whole network does not have the continuous rotation group equivariance. So we also add an equivariant loss to drive CRGEN-LN to learn the continuous rotation group equivariance.

Performance on MNIST-rot

Without Rotation Augmentation: In Table 1, we compare the performance of the proposed CRGEN with previous methods. We do not apply rotation augmentation on the training set of MNIST-rot. CRGEN-S has a similar architecture to G-CNN and PDO-eConv. Benefiting from the sparse parameters with flexible geometric parameters, the total parameters of CRGEN-T are much fewer than G-CNN and PDO-eConv (6.2K vs. 25K and 26K), while CRGEN-T achieves a lower error (1.16% vs. 2.28% and 1.87%). Compared with an efficient network E^4-Net, CRGEN-S can still achieve better performance (error of 0.85% vs. 1.17%) with similar parameters (40K vs. 41.4K). CRGEN-T can achieve similar performance (error of 1.16% vs. 1.17%) with much fewer parameters (6.2K vs. 41.4K). Compared with SFCNN, CRGEN-S can achieve a slightly lower error (0.85% vs. 0.88%) using much fewer parameters (40K vs. 6.5M). Appendix E.1 provides a plot of Table 1 to better display Para. vs. Err..

With Rotation Augmentation: When adopting rotation augmentation during training, the errors will be even lower. As Table 2 shows, CRGEN-S can achieve a 0.714% error(lower is better) which is the same as the network SFCNN and slightly higher than PDO-eConv (0.709% error). CRGEN-S needs only 40K parameters, while SFCNN and PDO-eCouv need 6.5M and 650K parameters, respectively. This means that our approach only uses 1/160 parameters of SFCNN and 1/16 parameters of PDO-eConv to achieve the same or competitive performance. When we double the filters of CRGEN-S, the performance can be further improved. CRGEN-L surpasses the performance of PDO-eConv with only a quarter of the parameters (150K vs. 650K).

Small Sample Learning: We also evaluate the performance in the case of small samples. We use the architecture of CRGEN-S without rotation augmentation. For comparison, we construct the conventional CNN and G-CNN with the same architecture as CRGEN-S and scale the channel numbers to control the total parameters. The amount of training samples varies from 100 to 12000. In the case of 100 samples, it means that each class only has 10 samples on average. In Figure 4, (a) shows accuracies of CNN, G-CNN, and CRGEN-S. (The detailed accuracies are in Appendix E.1). CNN and G-CNN have 92K parameters, which are more than double parameters of the proposed CRGEN (40K). Benefit from the rotation equivariance, G-CNN can achieve better performance than conventional CNN. Compared with G-CNN, CRGEN-S can achieve significantly better performance. The performance improvements vary from 1.61% to 24.12%. With only 200 samples(20 samples for each class on average), CRGEN can achieve an accuracy of 80.17%, while G-CNN can achieve only less than 60% and CNN even less than 50%. It shows that CRGEN has a great small sample learning ability.

Rotation Generalization Ability

We evaluated the rotation generalization ability of CRGEN on both MNIST and natural image datasets. We test the accuracy when the training set is non-rotated and the test set is randomly rotated. For MNIST, we test it with CRGEN-S. For the natural image, we construct a new network CRGEN-LN. As shown in Figure 5, CRGEN-LN adds a population coding based equivariant convolution layer at the end of G-CNN, and it adopts a pair-wise-learning manner and adds an equivariant loss to obtain equivariance of more orientations. The equivariant loss is that $Loss = CLS + \lambda MSE([H[L_r f]](x), [L'_r[Hf]](x))$, where f and $L_r f$ are the pair-wise input images, H is the whole network, CLS is the loss of the classification loss, and MSE is the equivariance loss with weight λ. Benefited the population encoding of the final layer, r of L'_r can be any angle in continuous rotation group. Practically, we utilize 8 orientation convolution in the final layer for CRGEN-LN. The input training data are rotated in k separate orientations. Each iteration we choose two orientations of them and use the proposed loss to drive the network to learn the equivariance. The k orientation is set as $(90^\circ + 90^\circ/k) * i, i = 0, 1, ..., k-1$.

Rotation Generalization Ability on MNIST: For the training set, we apply rotation augmentation with a probability of p. $p = 0$ means no rotation augmentation, and $p = 1$ means all training data is randomly rotated. For the test set, all test data is randomly rotated. Figure 4 (b) show

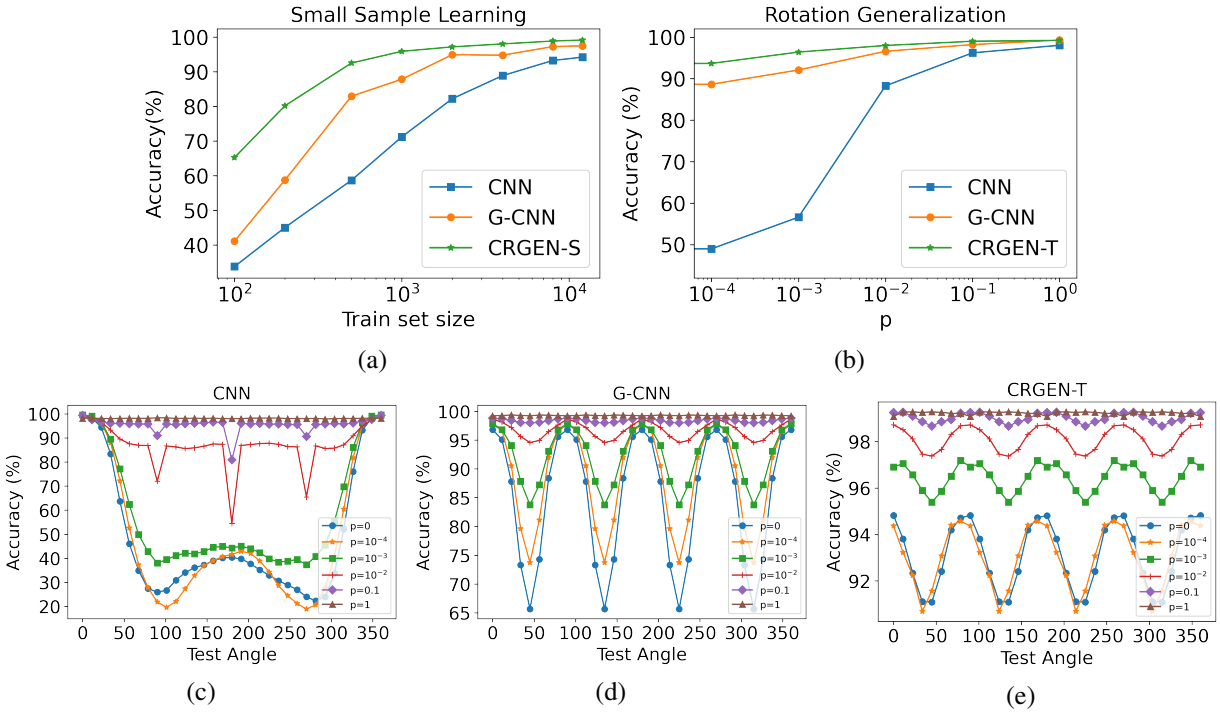


Figure 4: (a) Accuracies in small sampling learning on MNIST-rot. The number of training samples ranges from 100 to 12000. The plot shows accuracies of CNN, G-CNN, and the proposed CRGEN-S. (b) Accuracies with different proportions rotation training samples on MNIST. Training data are under rotation augmentated with probability p and test data are randomly rotated. (c), (d), (e) are the accuracies on each angle with different p of CNN, G-CNN, and the proposed CRGEN-T, respectively.

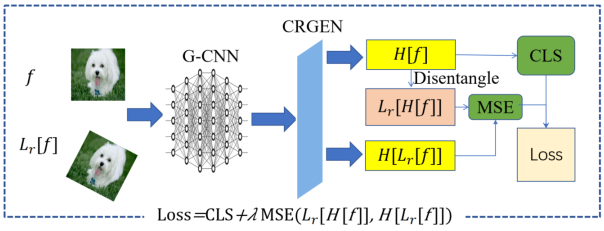


Figure 5: Architecture of CRGEN-LN. The backbone is G-CNN, and the final layer is the proposed continuous rotation group equivariant convolution. We use a loss function to drive it to learn equivariance on continuous rotation group.

the performance of different networks with p ranging from 0 to 1. The proposed CRGEN significantly performs better than CNN and G-CNN. When $p = 0$, networks are trained all by upright digits, so networks need to generalize what it has learned in arbitrary orientations. The conventional CNN lack this ability and it only achieves an accuracy of 44.70%. G-CNN can perform better than CNN (84.04% vs. 44.70%) benefiting from the equivariance of rotation, while the proposed CRGEN can achieve an even higher accuracy of 93.38% using only 6.2K parameters (21.7K for G-CNN). As p increases, CRGEN always performs better than CNN and G-CNN. When the training set is fully augmentated by rotation ($p = 1$), CRGEN and G-CNN achieve similar accuracy (99.29% vs. 99.32%) and still perform better than CNN (98.05%). (See detailed accuracies of Figure 4 (b) in Appendix E.2)

We also test the accuracy in each orientation. Figures 4(c), 4(d) and 4(e) shows the detailed accuracies at each angle for CNN, G-CNN, and CRGEN, respectively. Each line is a model with a specific p and each point in the line represents the accuracy when the test samples are rotated by a specific degree (from 0° to 360°). In Figure 4(c), CNN works well near the rotation angle of 0 degrees, but its performance drops rapidly (even less than 20%) away from 0 degrees. It shows that conventional CNN has a poor rotation generalization ability. In Figure 4(d) and 4(e), G-CNN and CRGEN show much better rotation generalization abilities. When $p = 0$, G-CNN can still achieve above 65% accuracy at any angle, while CRGEN-T can even achieve above 90% accuracy. We also test the rotation generalization ability with different rotation ranges of angle in training phrase, and the results also demonstrate the excellent rotation generalization of the proposed method. The detailed experimental results and analysis are in Appendix E.2.

Rotation Generalization Ability on Multi-datasets: Secondly, we also test on multiple datasets and network architectures. In Table 3, we set the pair-wise inputs as the same images of 0° and 135° ($k = 2$). For each dataset and network architecture, the models of CNN, G-CNN and CRGEN-LN all utilize similar architectures. For CRGEN-LN, we set the weight of the equivariant loss $\lambda = 0.1$. Com-

MNIST				
Arch.	Model	Depth	Para.	Acc.(%)
Plain	CNN	7	20.0K	60.27
	G-CNN	7	21.7K	96.23
	CRGEN-LN	7	**18.6K**	**97.82**
CIFAR10				
Arch.	Model	Depth	Para.	Acc.(%)
Plain	CNN	6	1.14M	50.76
	G-CNN	6	**0.29M**	77.41
	CRGEN-LN	6	**0.29M**	**79.66**
ResNet	CNN	110	162K	50.27
	G-CNN	110	46K	61.98
	CRGEN-LN	110	**44K**	**66.34**
ImageNet-100				
Arch.	Model	Depth	Para.	Acc.(%)
Plain	CNN	18	2.78M	54.56
	G-CNN	18	0.75M	72.42
	CRGEN-LN	18	**0.72M**	**72.75**
ResNet	CNN	18	3.16M	57.80
	G-CNN	18	0.84M	50.83
	CRGEN-LN	18	**0.81M**	**72.34**

Table 3: Accuracies on multiple datasets (i.e., MNIST, CIFAR10, and ImageNet-100) and multiple architectures (i.e., plain and ResNet architectures). Training data are rotated with only two specific angles 0° and 135°. The test data are randomly rotated.

pared with CNN, CRGEN-LN achieves accuracy increments ranging from 14.54% to 37.55% with much fewer parameters on MNIST, CIFAR10, and ImageNet-100. For plain architectures, CRGEN-LN works slightly better than G-CNN, and the accuracy increments ranges from 0.33% and 2.25%. For ResNet architectures, CRGEN-LN works significantly better than G-CNN, and it achieves accuracy increments of 4.36% and 21.51% on CIFAR10 and ImageNet-100, respectively. Especially on ImageNet-100, G-CNN with ResNet architecture does not work well, which is even worse than CNN (50.83% vs. 57.80%). By adding a population coding based equivariant convolution layer and an equivariant loss, CRGEN-LN with ResNet architecture works significantly better with an accuracy of 72.34%. Furthermore, we also carry out experiments on ResNet architectures with different depths and widths, and the results all shows that the proposed CRGEN-LN has better rotation generalization than both G-CNN and vanilla ResNet. (See detailed results and analysis in Appendix E.3)

Parameter Analysis

In Table 4, we analyze parameters of r, m_l, σ_x, and σ_g. r is the number of orientations and m_l is the number of sparse points in each kernel. σ_x and σ_g are the standard deviations on spatial and orientation dimensions, respectively. We use the CREGN-T on MNIST-rot as the baseline network. And we change the 4 parameters to analyze the influence of each parameter. For different r, the error is lower when r is larger. For different m_l, the error is lower when m_l increases from 2 to 7. And the parameters increase approximately linearly as m_l increases. For σ_x and σ_g, we set them as multiple times of their sampling steps (sampling steps: 1 on spatial dimension and $2\pi/16$ on orientation dimension). When we set them as about $1\times$ of their sampling steps, the error is the lowest. If we set it too large or too small, the error will increase, which is consistent with the analysis in theory. We also carry out the experiments of CRGEN-LN with different k and λ on MNIST. When $\lambda = 0$(without equivariant loss), CRGRN-LN still works better than CNN and GCNN. See detailed results and analysis in Appendix E.3.

r	4	8	12	16	20
Error(%)	2.72	1.64	1.25	1.16	**1.102**
m_l	2	3	5	7	9
Para.	3.8K	6.2K	11.1K	16.0K	20.8K
Error(%)	1.26	1.16	1.05	**0.998**	1.10
σ_x (and $\frac{\sigma_g}{2\pi/16}$)	0.5	0.75	1	1.25	1.5
Error(%)	1.29	1.17	**1.16**	1.28	1.48

Table 4: Parameter analysis of CRGEN-T on MNIST-rot.

Limitations and Future Works

The proposed group convolution is generated from sparse weights, which is parameter efficient. If we share the Gauss modulation by applying the Gauss convolution for the input features first and then applying a sparse convolution, it's also computional efficient (see details in Appendix F). However, sparse convolutions need additional works for engineering, we leave it for future works. Tuning curves help discrete group convolutions to achieve continuous equivariance. However, it also reduces the resolution especially for deep stacks of the proposed group convolutions. We solve it by adopting fewer proposed layers and imposing an equivariant loss to keep continuous equivariance. In the future works, learning-based equivariant networks still leave a huge room to study, and we believe that the idea of the tuning curve is still helpful.

Conclusion

Inspired by neural population coding, we propose a novel group equivariant convolution with tuning curves, which can achieve continuous equivariant by discrete group convolution. We use sparse weights with learnable geometric parameters to generate the group kernels, and it can achieve both competitive performance and parameter efficiency. Quantitative analysis shows that the equivariant error can be neglected when the standard deviation of the tuning curve is large enough(i.e., bigger than $1\times$ of the sampling step). We construct Continuous Rotation Group Equivariant Networks(CRGEN) by group convolutions with tuning curves. The experimental results demonstrated that CRGEN can obtain much competitive performance on the MNIST-rot with much fewer parameters (less than 25%) compared with previous SOTA methods. More experiments also show that the proposed CRGEN also has better small sample learning and rotation generalization abilities than the conventional CNN and discrete rotation group equivariant networks.

Acknowledgments

This work was supported by National Key R&D Program of China(2022ZD0116313), International Partnership Program of Chinese Academy of Sciences(173211KYSB2020002), CAS Project for Young Scientists in Basic Research(YSBR-041).

References

Averbeck, B. B.; Latham, P. E.; and Pouget, A. 2006. Neural correlations, population coding and computation. *Nature reviews neuroscience*, 7(5): 358–366.

Bekkers, E. J. 2019. B-Spline CNNs on Lie groups. In *International Conference on Learning Representations*.

Borst, A.; and Theunissen, F. E. 1999. Information theory and neural coding. *Nature neuroscience*, 2(11): 947–957.

Bruna, J.; and Mallat, S. 2013. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1872–1886.

Chan, T.-H.; Jia, K.; Gao, S.; Lu, J.; Zeng, Z.; and Ma, Y. 2015. PCANet: A simple deep learning baseline for image classification? *IEEE transactions on image processing*, 24(12): 5017–5032.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2014. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*.

Chen, Z.; Xu, T.-B.; Li, J.; and He, H. 2022. Sharing Weights in Shallow Layers via Rotation Group Equivariant Convolutions. *Machine Intelligence Research*, 19(2): 115–126.

Cheng, X.; Qiu, Q.; Calderbank, R.; and Sapiro, G. 2018. RotDCF: Decomposition of Convolutional Filters for Rotation-Equivariant Deep Networks. In *International Conference on Learning Representations*.

Cohen, T.; and Welling, M. 2016. Group equivariant convolutional networks. In *International conference on machine learning*, 2990–2999.

Dieleman, S.; De Fauw, J.; and Kavukcuoglu, K. 2016. Exploiting Cyclic Symmetry in Convolutional Neural Networks. In *International Conference on Machine Learning*, 1889–1898.

Esteves, C.; Allen-Blanchette, C.; Zhou, X.; and Daniilidis, K. 2018. Polar Transformer Networks. In *International Conference on Learning Representations*.

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

Franzen, D.; and Wand, M. 2021. General Nonlinearities in SO (2)-Equivariant CNNs. *Advances in Neural Information Processing Systems*, 34.

Ganea, O.-E.; Huang, X.; Bunne, C.; Bian, Y.; Barzilay, R.; Jaakkola, T.; and Krause, A. 2021. Independent SE (3)-Equivariant Models for End-to-End Rigid Protein Docking. *arXiv preprint arXiv:2111.07786*.

Girshick, R. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

He, K.; Gkioxari, G.; Dollar, P.; and Girshick, R. 2017. Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99): 1–1.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

He, L.; Chen, Y.; Dong, Y.; Wang, Y.; Lin, Z.; et al. 2021. Efficient Equivariant Network. *Advances in Neural Information Processing Systems*, 34.

Hoogeboom, E.; Peters, J. W.; Cohen, T. S.; and Welling, M. 2018. Hexaconv. *arXiv preprint arXiv:1803.02108*.

Hsu, K.-Y.; Li, H.-Y.; and Psaltis, D. 1990. Holographic implementation of a fully connected neural network. *Proceedings of the IEEE*, 78(10): 1637–1645.

Kamnitsas, K.; Ledig, C.; Newcombe, V. F. J.; Simpson, J. P.; Kane, A. D.; Menon, D. K.; Rueckert, D.; and Glocker, B. 2016. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Medical Image Analysis*, 36: 61.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet classification with deep convolutional neural networks. In *International Conference on Neural Information Processing Systems*, 1097–1105.

Laptev, D.; Savinov, N.; Buhmann, J. M.; and Pollefeys, M. 2016. Ti-pooling: transformation-invariant pooling for feature learning in convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 289–297.

Larochelle, H.; Erhan, D.; Courville, A.; Bergstra, J.; and Bengio, Y. 2007. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, 473–480.

Lecun, Y.; Bottou, L.; Bengio; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*.

Lin, M.; Chen, Q.; and Yan, S. 2013. Network in network. *arXiv preprint arXiv:1312.4400*.

Luo, C.; Zhu, Y.; Jin, L.; and Wang, Y. 2020. Learn to Augment: Joint Data Augmentation and Network Optimization for Text Recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Marcos, D.; Volpi, M.; Komodakis, N.; and Tuia, D. 2017. Rotation equivariant vector field networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 5048–5057.

Pouget, A.; Dayan, P.; and Zemel, R. 2000. Information processing with population codes. *Nature Reviews Neuroscience*, 1(2): 125–132.

Puny, O.; Atzmon, M.; Ben-Hamu, H.; Smith, E. J.; Misra, I.; Grover, A.; and Lipman, Y. 2021. Frame Averaging for Invariant and Equivariant Network Design. *arXiv preprint arXiv:2110.03336*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: towards real-time object detection with region proposal networks. In *International Conference on Neural Information Processing Systems*.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*.

Shen, Z.; He, L.; Lin, Z.; and Ma, J. 2020. PDO-eConvs: Partial Differential Operator Based Equivariant Convolutions. *arXiv preprint arXiv:2007.10408*.

Sifre, L.; and Mallat, S. 2013. Rotation, Scaling and Deformation Invariant Scattering for Texture Discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sohn, K.; and Lee, H. 2012. Learning invariant representations with local transformations. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 1339–1346.

Townshend, R. J.; Eismann, S.; Watkins, A. M.; Rangan, R.; Karelina, M.; Das, R.; and Dror, R. O. 2021. Geometric deep learning of RNA structure. *Science*, 373(6558): 1047–1051.

Usrey, W. M.; and Reid, R. C. 1999. Synchronous activity in the visual system. *Annual review of physiology*, 61(1): 435–456.

Wang, D.; Walters, R.; and Platt, R. 2022. SO(2)-Equivariant Reinforcement Learning. *arXiv preprint arXiv:2203.04439*.

Weiler, M.; and Cesa, G. 2019. General E(2) - Equivariant Steerable CNNs. *arXiv preprint arXiv:1911.08251*.

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

Worrall, D. E.; Garbin, S. J.; Turmukhambetov, D.; and Brostow, G. J. 2017. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5028–5037.

Worrall, D. E.; and Welling, M. 2019. Deep scale-spaces: Equivariance over scale. *arXiv preprint arXiv:1905.11697*.

Xi, Y.; Zheng, J.; Li, X.; Xu, X.; Ren, J.; and Xie, G. 2018. SR-POD: Sample rotation based on principal-axis orientation distribution for data augmentation in deep object detection. *Cognitive Systems Research*, 52(DEC.): 144–154.

Xu, J.; Kim, H.; Rainforth, T.; and Teh, Y. 2021. Group Equivariant Subsampling. *Advances in Neural Information Processing Systems*, 34.

Zemel, R. S.; Dayan, P.; and Pouget, A. 1998. Probabilistic interpretation of population codes. *Neural computation*, 10(2): 403–430.

Zhou, Y.; Ye, Q.; Qiu, Q.; and Jiao, J. 2017. Oriented response networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 519–528.