Variational Partial Group Convolutions for Input-Aware Partial Equivariance of Rotations and Color-Shifts

Hyunsu Kim [1] **Yegon Kim** [1] **Hongseok Yang** [2] **Juho Lee** [1 3]

Abstract

Group Equivariant CNNs (G-CNNs) have shown promising efficacy in various tasks, owing to their ability to capture hierarchical features in an equivariant manner. However, their equivariance is fixed to the symmetry of the whole group, limiting adaptability to diverse partial symmetries in real-world datasets, such as limited rotation symmetry of handwritten digit images and limited color-shift symmetry of flower images. Recent efforts address this limitation, one example being Partial G-CNN which restricts the output group space of convolution layers to break full equivariance. However, such an approach still fails to adjust equivariance levels across data. In this paper, we propose a novel approach, Variational Partial G-CNN (VP G-CNN), to capture varying levels of partial equivariance specific to each data instance. VP G-CNN redesigns the distribution of the output group elements to be conditioned on input data, leveraging variational inference to avoid overfitting. This enables the model to adjust its equivariance levels according to the needs of individual data points. Additionally, we address training instability inherent in discrete group equivariance models by redesigning the reparametrizable distribution. We demonstrate the effectiveness of VP G-CNN on both toy and real-world datasets, including MNIST67-180, CIFAR10, ColorMNIST, and Flowers102. Our results show robust performance, even in uncertainty metrics.

(a) MNIST **(b)** Flowers102

Figure 1. Illustrative example of partial equivariance. (a) $180°$-rotation of 6 is regarded as 9 but 7 is not. (b) The color-shifted image of Barberton Daisy looks similar to Osteospermum.

1. Introduction

Convolutional Neural Networks (CNNs) have demonstrated remarkable success in numerous computer vision tasks, owing to their ability to capture hierarchical features in an equivariant manner. Other approaches, such as Group Equivariant CNNs (G-CNNs) (Cohen & Welling, 2016; 2017; Weiler & Cesa, 2019; Romero et al., 2022), extend equivariance to various symmetry groups, enhancing model robustness across different transformations. However, a limitation arises from the rigidity of these models, as the choice of the equivariance group is fixed a priori.

In real-world scenarios, datasets often exhibit equivariance to diverse types of transformations, and the nature of equivariance might not be the same across all data instances. For example, in the classification of handwritten images like MNIST, images of 6 or 9 may be described more naturally by invariance to partial rotations between $-90°$ and $90°$, while a $180°$ rotation might distort the classification between 6 and 9, as shown in Fig. 1a. In contrast, the other digits, $0, 1, 2, 3, 4, 5, 7, 8$, may possess full equivariance to rotation. The challenge then lies in developing a neural network architecture that adapts the level of equivariance to the specific needs of the data.

Existing efforts have addressed this issue, such as Partial G-CNN (Romero & Lohit, 2022), which learns varying levels of equivariance at different layers, or Relaxed G-CNN

[1]Kim Jaechul Graduate School of AI, KAIST, Daejeon, South Korea [2]School of Computing, KAIST, Daejeon, South Korea [3]AITRICS, Seoul, South Korea. Correspondence to: Hyunsu Kim <kim.hyunsu@kaist.ac.kr>, Juho Lee <juholee@kaist.ac.kr>, Hongseok Yang <hongseok.yang@kaist.ac.kr>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

(Wang et al., 2022; van der Ouderaa et al., 2022), which incorporates relaxed kernel design. In particular, Partial G-CNN restricts the distribution of the output group space to break full equivariance. They introduce a convolution layer with a distribution whose support domain does not cover all group elements, effectively breaking equivariance. While this method has shown promising results, it imposes the same level of equivariance for all data points.

In this paper, we introduce a new group equivariant convolution that captures different levels of partial equivariance in a data-specific manner. We redesign the distribution of output group elements in Partial G-CNN to be conditioned on the input. For efficient computation, the data-dependent conditional distribution refers to features extracted from the previous layer, as these contain information about the input data. To train the conditional distribution without overfitting, we adopt Variational inference, treating the group elements in each layer as random variables. Thus, the problem becomes maximizing the evidence lower bound (ELBO), consisting of the log-likelihood for classification and the Kullback-Leibler (KL) divergence between the conditional distribution and a certain prior for regularization. Therefore, while the conditional distribution is regularized towards full equivariance, if full equivariance is harmful for the given data, it modifies the distribution to provide partial equivariance. Additionally, we address the unstable training issue in discrete group equivariance, which Partial G-CNN suffers from, by redesigning the reparametrizable distribution of the group elements. Our method, called Variational Partial G-CNN (VP G-CNN), shows promising results in terms of test accuracy and uncertainty metrics. It also demonstrates the ability to detect different levels of equivariance for each data point in one toy dataset, MNIST67-180, and three real-world datasets: CIFAR10, ColorMNIST, and Flowers102.

To sum up, our contributions can be summarized as follows:

1. We propose input-aware partially equivariant group convolutions, which capture different levels of equivariance across data based on variational inference.
2. We resolve the unstable training issue of discrete group equivariance involved in Partial G-CNN by redesigning the reparametrizable distribution for the discrete groups.
3. We demonstrate promising results on real-world datasets: CIFAR10, ColoredMNIST, and Flowers102, alongside demonstrating strong calibration performance.

2. Preliminaries

2.1. Group Equivariance and Partial Equivariance

A representation of a group G on a Euclidean space $\mathbb{R}^n$ can be defined as a function ρ mapping G to the general linear group on $\mathbb{R}^n$ (i.e., the group of invertible $n \times n$ matrices with matrix multiplication as group composition and identity matrix as identity element), ensuring that ρ preserves the composition operator and the identity element of the group. When we possess representations of a group G in Euclidean spaces $\mathcal{X}$ and $\mathcal{Y}$, denoted as $\rho_{\mathcal{X}}$ and $\rho_{\mathcal{Y}}$ respectively, a function $\Phi : \mathcal{X} \to \mathcal{Y}$ is termed *equivariant to* G if, for all $g \in G$ and $\boldsymbol{x} \in \mathcal{X}$, the following condition holds:

$$\Phi\big(\rho_{\mathcal{X}}(g)(\boldsymbol{x})\big) = \rho_{\mathcal{Y}}(g)\big(\Phi(\boldsymbol{x})\big). \tag{1}$$

In simpler terms, this condition implies that Φ does not actively utilize information that can be altered by group elements g.

As a more general concept, partial group equivariance, or *partial equivariance*, can be defined as follows:

Definition 2.1 ((S, ε, G)-Partial Equivariance)**.** Let $\Psi : \mathcal{X} \to \mathcal{Y}$ be a function and G be a group acting on $\mathcal{X}$. The function Ψ is partially G-equivariant with respect to a subset $S \subseteq \mathcal{X}$ and an error threshold $\varepsilon > 0$ if the following holds,

$$\sup_{g\in G} \big\|\Psi(\rho_{\mathcal{X}}(g)(\boldsymbol{x})) - \rho_{\mathcal{Y}}(g)\big(\Psi(\boldsymbol{x})\big)\big\| = 0, \quad \boldsymbol{x} \in S, \tag{2}$$
$$\sup_{g\in G} \big\|\Psi(\rho_{\mathcal{X}}(g)(\boldsymbol{x}')) - \rho_{\mathcal{Y}}(g)\big(\Psi(\boldsymbol{x}')\big)\big\| \leq \varepsilon, \quad \boldsymbol{x}' \in \mathcal{X} \setminus S,$$

that is, it is equivariant on a given subset S and approximately equivariant outside S.

The set S is determined with respect to the given dataset and group, typically defined as a subset of $\mathcal{X}$ that excludes certain inputs known to possess specific symmetries. For example, in the MNIST dataset with respect to the $SO(2)$ group, subset S includes digit images other than 6 and 9. Notice that for $\boldsymbol{x} \in S$, the function Ψ must exhibit full equivariance, while for $\boldsymbol{x} \notin S$, it must exhibit ε-approximate equivariance. This definition ensures that equivariance is enforced on a specific subset S of the domain, while allowing for ε-approximate equivariance with respect to inputs outside S.

Definition 2.2 ((C, ε, G)-Partial Equivariance on Feature Map)**.** Let G be a group acting on $\mathcal{F}$ and $\Phi : \mathcal{F} \to \mathcal{F}$ be a map between functions $f : G \to \mathbb{R}^d$ representing input feature maps on group G. The function Φ is partially G-equivariant with respect to a subset $C \subseteq \mathcal{F}$ and an error threshold $\varepsilon > 0$ if for all $u \in G$, it satisfies that:

$$\sup_{g\in G} \|\Phi(\mathcal{L}_g f)(u) - (\mathcal{L}_g \Phi(f))(u)\| = 0, \quad f \in C, \tag{3}$$
$$\sup_{g\in G} \|\Phi(\mathcal{L}_g f')(u) - (\mathcal{L}_g \Phi(f'))(u)\| \leq \varepsilon, \quad f' \in \mathcal{F} \setminus C$$

where $\mathcal{L}_g$ is the group representations of the group element g.

2.2. G-CNN and Partial G-CNN

The convolutional layers of CNNs for an image can be described in terms of a function $f : \mathbb{R}^2 \to \mathbb{R}^3$ that maps the position of a pixel to its RGB vector and represents the input image, and a kernel $k : \mathbb{R}^2 \to \mathbb{R}^{3\times d}$, where d is the output feature dimension. They output $(k * f) : \mathbb{R}^2 \to \mathbb{R}^d$ defined by $(k * f)(\boldsymbol{y}) = \int_{\mathbb{R}^2} k(\boldsymbol{x} - \boldsymbol{y}) f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}$. The convolutional neural network exhibits translation equivariance due to the property $\mathcal{L}_g(k * f) = k * \mathcal{L}_g f$, where $\mathcal{L}_g$ denotes a translation (shift) operation of image pixels: $\mathcal{L}_g f(\boldsymbol{x}) = f(\boldsymbol{x} - \boldsymbol{t})$. Likewise, the convolutional layers of G-CNN utilize the equivariance property of the convolution operation on an extended space defined on a certain group G, which may include a translation group.

Lifting convolution. We want to do the group convolution on a group G, but the input like an image is typically a map defined on a space $E \subseteq \mathbb{R}^m$ and so it needs to be lifted to a map from the group G. The lifting convolution performs this lifting. If there is an embedding of E to G so that E can be regarded as a subgroup of G, we have, for an input feature map $f : E \to \mathbb{R}^3$ and a kernel map $k : G \to \mathbb{R}^{3\times d}$, the following lifting convolution $k *_{\text{lift}} f : G \to \mathbb{R}^d$: for all $u \in G$,

$$(k *_{\text{lift}} f)(u) = \int_{v\in E} k(v^{-1}u) f(v) \mathrm{d}\mu_E(v) \tag{4}$$

where elements in E are viewed as group elements in G, and μ_E is the restriction of the left Haar measure of G to the subgroup E. Under an appropriate condition, the lifting convolution defined on group G is equivariant to the group G, i.e. for all $g \in G$,

$$(k *_{\text{lift}} \mathcal{L}_g f)(u) = \mathcal{L}_g(k *_{\text{lift}} f)(u), \tag{5}$$

where $\mathcal{L}_g f(u) = f(g^{-1}u)$.

Group convolution. The group convolution generalizes the regular convolution for equivariances with respect to general groups. Once the inputs are feature maps from G, the group equivariant convolution for an input feature map $f : G \to \mathbb{R}^d$ and a kernel $k : G \to \mathbb{R}^{d\times n}$, where n is the output feature dimension, is defined as follows:

$$(k * f)(u) = \int_{v\in G} k(v^{-1}u) f(v) \mathrm{d}\mu_G(v), \tag{6}$$

where μ_G is the left Haar measure of the group G. Similarly to the regular convolution, the group convolution is G-equivariant, that is, $k * \mathcal{L}_g f = \mathcal{L}_g(k * f)$.

Partial group convolution. Inspired by Augerino (Benton et al., 2020), Partial G-CNN (Romero & Lohit, 2022) introduced a partially equivariant group convolution whose output feature space is determined by a distribution $q(u)$, where $u \in G$. It modified the group convolution as follows:

$$(k * f)(u) = \int_{v\in G} q(u) k(v^{-1}u) f(v) \mathrm{d}\mu_G(v). \tag{7}$$

For instance, when G is the 2-dimensional rotation group $SO(2)$ with radian values in $[-\pi, \pi]$, the distribution $q(u)$ can be defined as the push forward of the exponential map $\exp : \mathbf{g} \to G$ of the distribution $\mathrm{Unif}[R(-\theta), R(\theta)]$ on the Lie algebra $\mathbf{g}$, where θ is a learnable parameter on radian space and $R : \mathbb{R} \to so(2)$, and represents the maximum possible rotations in $\mathbb{R}^2$. That is,

$$u = \exp(t), \quad t \sim \mathrm{Unif}[R(-\theta), R(\theta)]. \tag{8}$$

If the full equivariance (i.e. $\theta = \pi$) is harmful for training, the model modifies the θ to be less than π. However, Partial G-CNN fails to guarantee the partial equivariance for a non-empty S in Definition 2.1, when $\theta < \pi$. This is because, when θ becomes less than π, Partial G-CNN loses equivariance to G for *all* $\boldsymbol{x} \in \mathcal{X}$. This departure from equivariance violates the condition specified for a subset S if $S \neq \emptyset$. The model either exhibits full equivariance when $\theta = \pi$ or broken equivariance when $\theta < \pi$. For convenience, we omit the exponential map and mapping R when we describe the distribution of group elements, and write $q(u; \theta) = \mathrm{Unif}[-\theta, \theta]$ or $q(u) = \mathrm{Unif}[-\theta, \theta]$.

Color equivariance H_m. We aim to achieve equivariance not only with respect to the standard group SE(2), but also concerning color shifts. In (Lengyel et al., 2023), color equivariance is defined as being equivariant to changes in hue. It is explained that the Hue-Saturation-Value (HSV) color space represents hue using an angular scalar value, and shifting hue involves a straightforward additive adjustment followed by a modulo operation. When translating the HSV representation into the three-dimensional RGB space, a hue shift corresponds to a rotation along the $(1, 1, 1)$ diagonal vector. Color equivariance is established in terms of a group by defining H_m, which consists of multiples of $360/m^\circ$ rotations around the $(1, 1, 1)$ vector in $\mathbb{R}^3$. H_m is a subgroup of $SO(3)$, the group of all rotations about the origin in $\mathbb{R}^3$. The group operation is matrix multiplication, acting on the continuous space of RGB pixel values in $\mathbb{R}^3$. Consequently, color-equivariant convolutions can be constructed using discrete $SO(3)$ convolutions when the RGB pixels of an image are treated as $\mathbb{R}^3$ vectors forming three-dimensional point clouds.

3. Variational Partial G-CNN

3.1. Input-Aware Partial Convolution

In order to achieve partial equivariance defined in Definition 2.1, we need to make the distribution $q(u)$ input-aware, and design $q(u|\boldsymbol{x})$ for each input $\boldsymbol{x}$. One approach is to put $q(u|\boldsymbol{x})$ for every layer, but doing so would be memory-inefficient, especially for the continuous group convolutions. This approach requires retaining the group elements sampled from $q(u|\boldsymbol{x})$ for all convolution layers during feed-forwarding.

Therefore, for partial equivariance, our new convolution at layer $l+1$ uses $q(u|f^{(l)})$ where $f^{(l)}$ is the output of the previous layer l. Since as a feature, $f^{(l)}$ contains information about the input data, this scheme has a potential to identify data-specific equivariance, while being memory-efficient. Concretely, we modify the convolutions in Eqs. 4 and 6 as follows:

$$\begin{aligned}(k *_{\text{lift}} f)(u) &= \int_{v\in E} q(u|f)k(v^{-1}u)f(v)\mathrm{d}\mu_E(v),\\ (k * f)(u) &= \int_{v\in G} q(u|f)k(v^{-1}u)f(v)\mathrm{d}\mu_G(v).\end{aligned} \tag{9}$$

The distribution $q(u|f)$ here must be partially equivariant in order to achieve partial equivariance in these convolutions. For example, if the input f is the image of digit 7 or 8, which require full equivariance to $SO(2)$, $q(u|f)$ can be just the uniform distribution for all rotations in $\mathbb{R}^2$: $q(u|f) = \mathrm{Unif}[-\pi,\pi]$. Note that in this case, $q(u|f)$ is equivariant to $SO(2)$ in the following sense: $q(u|f) = q(gu|\mathcal{L}_g f)$ for all $g \in SO(2)$. On the other hand, for the images of digit 6 or 9, which require only partial equivariance to $SO(2)$, $q(u|f)$ can be a uniform distribution with a narrower range, such as $\mathrm{Unif}[-\pi/2,\pi/2]$, or just a dirac-delta distribution $\delta(u)$. Note that in this case, $q(u|f)$ may fail to satisfy the equivariance condition, i.e., $q(u|f) \neq q(gu|\mathcal{L}_g f)$ for some $g \in G$. The next proposition gives one sufficient condition for ensuring partial equivariance of our convolutions:

Proposition 3.1. *Assume that the conditional distribution $q(u|f)$ is partially equivariant with respect to a group G and an equivariant subset $C \subseteq \mathcal{F}$ in the following sense:*

$$\begin{aligned}&\sup_{g\in G} \|q(u|f) - q(gu|\mathcal{L}_g f)\| = 0, \quad f \in C,\\ &\sup_{g\in G} \|q(u|f') - q(gu|\mathcal{L}_g f')\| \leq \varepsilon, \quad f' \in \mathcal{F}\setminus C,\end{aligned} \tag{10}$$

where $\mathcal{L}_g f(u) = f(g^{-1}u)$, and kernel k and input f of the group convolutions defined in Eq. 9 are bounded. Then, the group convolutions are also partially equivariant to G and C.

The proof is presented in Appendix A.1. For continuous groups, the integrals in the convolutions are intractable, so we typically employ Monte Carlo approximation to estimate the convolution operation by uniformly sampling from the Haar measure $\mathrm{d}\mu_G$. Thus, the approximate partially equivariant group convolution is determined as follows:

$$(k * f)(u_j) = \sum_{v_i} q(u_j|f)k(v_i^{-1}u_j)f(v_i). \tag{11}$$

Now, we describe how the distribution $q(u|f)$ can be trained and implemented using variational inference with the reparametrization trick.

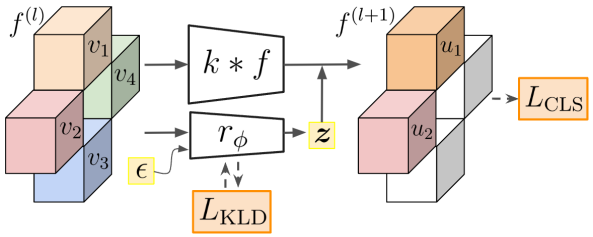


Figure 2. Architecture of Variational Partial Group Convolutions. The colored boxes are the features at each layer and the white boxes are zero features removed out by the distribution $q(u|f)$, where $u = r_\phi(f,\epsilon)$.

3.2. Variational Inference of $q(u|f)$

If we train $q(u|f)$ with only the classification loss, since it encompasses all features f, it may overfit by tending to become another classifier itself, leading to a trivial distribution. To prevent this situation, we adopt variational framework to train the distribution $q(u|f)$. Our goal is to maximize the log-likelihood $\log p(y|\boldsymbol{x})$ for $\boldsymbol{x}, y$ from a dataset $\mathcal{D}$ and it can be described as follows:

$$\begin{aligned}&\log p(y|\boldsymbol{x}) \\ &= \int_G \log p(y|f^{(0)},u^{(1)},\ldots,u^{(L)})\Pi_{l=1}^L p(u^{(l)})\mathrm{d}\mu_G(u^{(l)}),\end{aligned} \tag{12}$$

where $\boldsymbol{x} = f^{(0)}$, L is the number of layers of the model, and $u^{(l)}$ is the output group elements at layer l.

To estimate the approximate posterior $q(u^{(l)}|f^{(l)})$ at layer l, we maximize the evidence lower bound (ELBO) of the log-likelihood in Eq. 12:

$$\begin{aligned}&L_{\text{VP}} = \\ &\mathbb{E}_{\{u^{(l)}\}_{l=1}^L}\left[\log \frac{p(y|f^{(0)},\{u^{(l)}\}_{l=1}^L)\Pi_{l=1}^L p(u^{(l)})}{\Pi_{l=1}^L q(u^{(l)}|f^{(l)})}\right],\end{aligned} \tag{13}$$

where the expectation is over $\{u^{(l)}\}_{l=1}^L \sim \Pi_{l=1}^L q(u^{(l)}|f^{(l)})$. Then, $\mathbb{E}_{\mathcal{D}}[\log p(y|\boldsymbol{x})] \geq L_{\text{VP}}$ and by maximizing L_{VP}, we can maximize the log-likelihood indirectly. The approximate posterior $q(u^{(l)}|f^{(l)})$ is the partially equivariant distribution shown in Eq. 9.

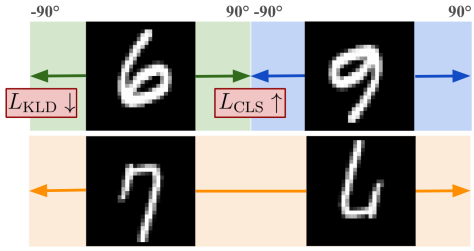


Figure 3. As the L_{KLD} increases, the distribution $p(u|f)$ expands, but upon reaching a certain point where L_{CLS} is affected, the distribution becomes constrained.

In fact, ELBO can be viewed as two components consisting of maximizing likelihood for classification and minimizing Kullback-Leibler (KL) divergence between the approximate posterior $q(u^{(l)}|f^{(l)})$ and prior $p(u^{(l)})$ for regularization.

$$
\begin{aligned}
L_{\text{VP}} &= L_{\text{CLS}} - \sum_{l=1}^{L} L_{\text{KLD}}^{(l)}, \\
L_{\text{CLS}} &= \mathbb{E}_{u^{(1)},\dots,u^{(L)}}\left[\log p(y|f^{(0)}, u^{(1)}, \dots, u^{(L)})\right] \\
L_{\text{KLD}}^{(l)} &= \text{D}_{\text{KL}}\left(q(u^{(l)}|f^{(l)}) \| p(u^{(l)})\right).
\end{aligned} \tag{14}
$$

The prior distribution is set to be a uniform distribution in which the probabilities of every group elements are the same, which corresponds to the full equivariance. Therefore, the model regularize $q(u|f)$ to preserve the full equivariance but if the full equivariance is harmful for training, it adjust the distribution $q(u|f)$ far from the uniform distribution. This principle is illustrated in Fig. 3. In practice, a hard regularization of the KL divergence is possible to disturb training of the target model. Therefore, we control strength of L_{KLD} by adopting a coefficient $\lambda \in [0, 1]$,

$$
L_{\text{VP}} = L_{\text{CLS}} - \lambda \sum_{l=1}^{L} L_{\text{KLD}}^{(l)}. \tag{15}
$$

λ is a hyperparameter that user can assign.

To efficiently train the distribution $q(u|f)$, we need to estimate the gradient of the loss with low variance. Thanks to reparametrization trick (Kingma & Welling, 2014), if we design the distribution possible to allow the backpropagation, we get estimates of the gradient with low variance. The gradient of ELBO can be estimated as follows:

$$
\begin{aligned}
\nabla_\theta L_{\text{CLS}} &= \mathbb{E}_{\epsilon^{(1)},\dots,\epsilon^{(L)}}\left[\nabla_\theta \log p_\theta(y|\boldsymbol{x}, \boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(L)})\right], \\
\nabla_\phi L_{\text{KLD}}^{(l)} &= \nabla_\phi \, \text{D}_{\text{KL}}\left(q_\phi(\boldsymbol{z}^{(l)}|f^{(l)}) \| p(\boldsymbol{z}^{(l)})\right) \\
&= \mathbb{E}_{\epsilon^{(l)}}\left[\nabla_\phi \log \frac{p(\boldsymbol{z}^{(l)})}{q_\phi(\boldsymbol{z}^{(l)}|f^{(l)})}\right],
\end{aligned} \tag{16}
$$

where $\boldsymbol{z}^{(l)} = r_\phi^{(l)}(f^{(l)}, \epsilon^{(l)})$ and θ, ϕ are the parameters of the classifier p_θ and the group element encoder r_ϕ, respectively, and θ includes ϕ because the classifier shares parameter with the encoder. The architecture of the input-aware partial group convolution is summarized in Fig. 2.

The partially equivariant distribution $q(u^{(l)}|f^{(l)})$ is sampled by uniformly drawing noise ϵ and feed-forward through the group element encoder r_ϕ. The reparametrizable encoder is designed differently across the continuous group and the discrete group. Although our method is able to apply multi-dimensional continuous and dicrete groups when appropriate distribution is defined, we narrow down the scope to the continuous two-dimensional rotation group $SO(2)$ and the discrete color-shift group H_m, which are widely tackled in the examples of the partial equivariance.

Rotation $SO(2)$ (continuous). Similar to Partial G-CNN (Romero & Lohit, 2022), we can define $q(u|f)$ a uniform distribution $\text{Unif}[-\theta, \theta]$ but θ is calculated from encoding of the input feature, $\theta = e_\phi(f)$, $\theta \in [0, 1]$, then $r_\phi(f, \epsilon)$ is described as

$$
r_\phi(f, \epsilon) = \epsilon\pi \cdot e_\phi(f), \quad \epsilon \sim \text{Unif}[-1, 1]. \tag{17}
$$

If $\theta = 1$, the probabilities of all group elements are the same, while if $\theta = 0$, the distribution becomes a dirac-delta distribution whose value is non-zero only at zero-rotation. This distribution is reparametrizable so we can estimate the gradient as in Eq. 16 with low variance.

Color-shift H_m (discrete). The color-shift group H_m has m number of group elements and each represents $360/m°$ rotations around the $(1, 1, 1)$ vector in the three-dimensionl RGB vector space. To sample group elements in such a discrete group, Partial G-CNN utilizes Gumbel-Softmax trick (Maddison et al., 2017) with Straight-Through estimation but it suffers from unstable training (Romero & Lohit, 2022). We observe that the distribution $p(u)$ with learnable parameters irregularly change their distribution during training and this may be due to the multi-modality of Gumbel-Softmax. Therefore, we propose another probability distribution that samples the discrete group without Gumbel-Softmax and mimick the distribution described in the continuous group.

For sampling, we first encode the input feature to $\theta = e_\phi(f)$, $\theta \in [0, \infty)$ and sample $\{\epsilon_i\}_{i=1}^m$ from a discrete uniform distribution $\text{Unif}\{1, 2, \dots, m\}$, corresponding to the uniform distribution in the continuous group. Then, we compute importance weights for each ϵ_i as

$$
w_i = \frac{\exp(\epsilon_i/\theta)}{\sum_{i=1}^m \exp(\epsilon_i/\theta)}. \tag{18}
$$

Here, θ determines smoothness of the softmax function across each ith component; if θ is large enough, w_i converges to almost uniform. Now using Straight-Through estimator, we select which group element in $\{u_i\}_{i=1}^m$ should

be non-zero.

$$q(u_i|f) = \begin{cases} 1, & \text{if } w_i > \frac{1}{m} - \eta, \\ 0, & \text{otherwise,} \end{cases} \tag{19}$$

where $\eta \in [0, 1/m]$ is a hyperparameter that determines how easy to be selected as non-zero. As θ increases, the difference in magnitude between w_i decreases, and more elements surpass the threshold. Conversely, as θ decreases, the difference in magnitude between w_i increases, and fewer elements surpass the threshold. This principle is analogous to the distribution of continuous groups. For example, if η is zero, w_i should be greater than $1/m$ to be non-zero so it always select only one group elements, whereas if η is $1/m$, it always select every elements in the group. The model trains value of θ so that it decides how many group elements are appropriate for given input. For instance, $m = 3$, $\eta = 7/12$, $\{\epsilon_i\}_{i=1}^m = \{3, 2, 1\}$, and then the threshold $1/m - \eta = 0.25$. For $\theta = 1$, $\{w_i\} = \{0.67, 0.24, 0.09\}$ and 0.67 is the only value larger than 0.25, thereby only u_1 is selected. For $\theta = 3$, $\{w_i\} = \{0.45, 0.32, 0.23\}$ and $0.45, 0.32$ are above the threshold, thus u_1 and u_2 are selected. Since at least one of the softmax result in Eq. 18 for m candidates should be greater than $1/m$, Eq. 19 always selects at least one group elements.

3.3. Implementation

Utilizing the input-aware partial group convolution for every layers would be the best strategy to gain performance. However, there are limitations to performance improvement compared to the increase in parameters. Hence, throughout the experiments we set a portion of layers to be the input-aware partial convolution in a network. In fact, once at least one of the convolutional layers exhibits input-aware partial equivariance, the entire network becomes partially equivariant.

Proposition 3.2. *If at least one of the convolutional layers in a G-CNN is partially equivariant to a group G and an equivariant subset $C \subseteq \mathcal{F}$, and its activation functions are equivariant with respect to G and L-Lipschitz continuous, and its kernel functions are bounded, then the entire G-CNN is also partially equivariant to G and C.*

Its proof is described in Appendix A.2. For example, in the CIFAR10 dataset, we apply the input-aware partial group convolution in the lifting convolution and the last group convolution only. In the Flower102 dataset, we apply it in the last two group convolution only. In addition, we use light-weighted encoder e_ϕ, which calculate θ as in Eqs. 17 and 18, consisting of two global average pooling layers, two one-dimensional convolution, and one linear layer. The detailed architecture is described in Appendix C.

4. Related Work

Group equivariant networks. G-CNN (Cohen & Welling, 2016) proposed a convolutional neural network architecture ensuring equivariance to a group of input transformations, including translation, rotation, and reflection, thereby enhancing the model's ability to learn and generalize from data with inherent symmetries in a given dataset. Steerable CNN (Cohen & Welling, 2017) introduced a framework for constructing rotation-equivariant convolutional neural networks, enabling efficient and flexible modeling of rotational symmetries in image data by leveraging the theory of group representations. $E(2)$-CNN (Weiler & Cesa, 2019) demonstrated constraints based on group representations, simplifying them to irreducible representations and providing a general solution for $E(2)$, thereby covering continuous group equivariance for images. CEConv (Lengyel et al., 2023) extended equivariance from geometric to photometric transformations by incorporating parameter sharing over hue shifts, interpreted as a rotation of RGB vectors, offering enhanced robustness to color changes in images.

Approximate equivariance. RPP (Finzi et al., 2021) involved placing one equivariant neural network (NN) and one non-equivariant NN in parallel, with a prior imposed on the parameters of each NN. In contrast, PER (Kim et al., 2023) replaced the two components with a single non-equivariant NN and introduced a regularizer to drive the non-equivariant NN towards equivariance. Relaxed G-CNN (Wang et al., 2022) introduced a small linear kernel to G-CNN, which slightly breaks the group equivariance of the model. In Partial G-CNN (Romero & Lohit, 2022), a distribution of group elements in the output was adopted, allowing group convolutions to consider only a subset of group elements in the hidden space.

Input-aware automatic data augmentation. MetaAugment (Zhou et al., 2021) presents an efficient approach to learning a sample-aware data augmentation policy for image recognition by formulating it as a sample reweighting problem, where an augmentation policy network adjusts the loss of augmented images based on individual sample variations. AdaAug (Cheung & Yeung, 2022) learns adaptive data augmentation policies in a class-dependent and potentially instance-dependent manner, addressing the limitations of methods like AutoAugment (Cubuk et al., 2019) and Population-based Augmentation (Ho et al., 2019) by efficiently adapting augmentation policies to specific datasets. InstaAug (Miao et al., 2023) learns input-specific augmentations automatically by introducing a learnable invariance module that maps inputs to tailored transformation parameters, facilitating the capture of local invariances. Singhal et al. (2023) designed a method to capture multi-modal partial invariance by parameterizing the distribution of instance-specific augmentation using normalizing flows.

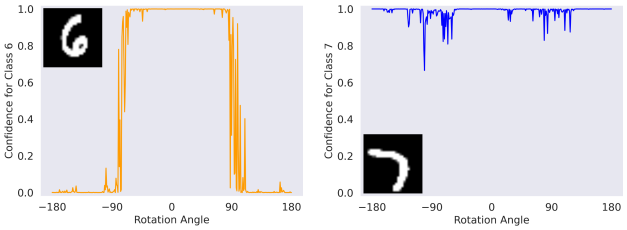


Figure 4. Partial equivariance trained on MNIST67-180. The x-axis represents the rotation angle of the input and the y-axis represents the model's confidence for the corresponding class. The model exhibits equivariance to rotations on semi-circle for image 6, whereas it shows full equivariance for image 7.

5. Experiments

In commonly addressed tasks, approximate equivariance often manifests in forms such as rotation and color shifts. To evaluate VP G-CNN's partial equivariance in rotations, we conduct experiments on two datasets: MNIST67-180 and CIFAR10 (Krizhevsky & Hinton, 2009). For color shifts, we assess performance on long-tailed colorMNIST, which exhibits full equivariance for color shifts but has imbalanced classes, and on Oxford Flower102 (Nilsback & Zisserman, 2008), where partial equivariance is data-specific, as depicted in Fig. 1b. We compare our model with four baseline methods: ResNet (T(2)-CNN), G-CNN, Partial G-CNN, and InstaAug. InstaAug (Miao et al., 2023) is an AutoAugment technique that learns the appropriate distribution of augmentations for each data instance. Detailed hyperparameters used to train VP G-CNN and the baselines are listed in Appendix B. The source code demonstrating the experiements in colorMNIST and Flowers102 is available at `https://github.com/yegonkim/partial_equiv`.

Model architecture for $SE(2)$. The group $SE(2)$ consists of translations $T(2)$ and rotations $SO(2)$. Similar to Partial G-CNN for $SE(2)$, we employ the extended version of G-CNN proposed by Finzi et al. (2020), Continuous Kernel Convolution (CKConv) (Romero et al., 2022). However, we use the input-aware partial group convolution as defined in Eq. 9, and we parametrized the convolutional kernels k as SIRENs (Sitzmann et al., 2020). The overall structure is based on ResNet (He et al., 2016) and it consists of one lifting convolution, two residual blocks, and one last linear layer. According to Proposition 3.2, we apply the input-aware convolution on the lifting convolution and the last group convolution and the other convolutions are all partial group convolution of Partial G-CNN. We define $q(u|f)$ the straight-through distribution as proposed in Eq. 17.

MNIST67-180 (toy dataset). Inspired by MNIST6-180, as introduced in (Romero & Lohit, 2022), we created a new classification dataset named MNIST67-180. This dataset is derived from the MNIST handwritten dataset (LeCun et al., 2010) and consists of images labeled as either 6 or 7, along with their corresponding $180°$-rotated versions labeled as 9 and 7, respectively. Consequently, images of 6 should be classified as 6 within a rotation range of $[-90°, 90°]$, and as 9 within other angles of rotation. Meanwhile, images of 7 should always be classified as 7, regardless of the angle of rotation. We demonstrate the learned partial equivariance for some of the data.

We plot the probabilities of assigning the label 6 for image 6 and the label 7 for image 7 with respect to the test samples of MNIST67-180 rotated at whole angles in $[0°, 360°]$. As shown in Fig. 4, the model learns to predict image 6 as 6 within the rotation range of $[-90°, 90°]$, while it learns to predict image 7 as 7 within the rotation range of $[-180°, 180°]$. This proves that our VP G-CNN learns an appropriate level of equivariance that varies for each type of data.

Table 1. Test accuracy on CIFAR10 with $SE(2)$-CNNs. *P* and *VP* denote that their architecture includes Partial and VP convolutional layers, respectively. ✓in the InstaAug column means the training is conducted with the augmentation of InstaAug.

Group	#Elems.	Partial	InstaAug	CIFAR10
$T(2)$	1	-	-	$82.0_{\pm 0.2}$
			✓	$81.9_{\pm 0.4}$
$SE(2)$	4	-	-	$83.9_{\pm 0.3}$
			✓	$81.2_{\pm 1.8}$
		P	-	$\mathbf{85.1}_{\pm \mathbf{0.6}}$
		VP	-	$\mathbf{85.1}_{\pm \mathbf{0.4}}$
	8	-	-	$86.8_{\pm 0.6}$
			✓	$82.4_{\pm 0.5}$
		P	-	$\mathbf{87.3}_{\pm \mathbf{0.4}}$
		VP	-	$\mathbf{87.6}_{\pm \mathbf{0.2}}$

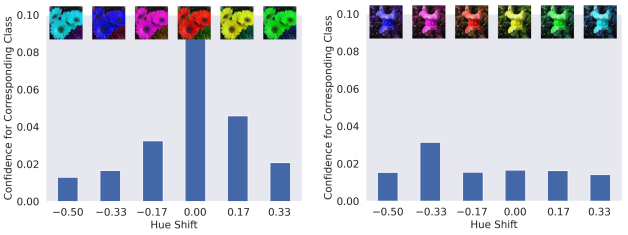
CIFAR10. We verify that VP G-CNN for rotation also works well in the widely-used image classification benchmark, CIFAR10. CIFAR10 is a collection of natural object images, such as airplanes, dogs, and so on, and it does not exhibit partial equivariance because the class should not change even if we rotate the image. However, the training and test datasets do not contain rotated images; they only pose upright. This leads partial group convolutions to be partially equivariant. As shown in Table 1, Partial G-CNN and VP G-CNN show competitive performance compared to fully equivariant G-CNN (3rd and 7th rows). This explains that partial equivariance is helpful in CIFAR10. Since the equivariance levels across the data do not differ enough, Partial G-CNN (5th and 9th rows) and VP G-CNN (6th and 10th rows) show comparable performance. On the other hand, InstaAug (4th and 8th rows) presents poor performance even when applied in the regular CNN. This is caused by the unstable training of InstaAug.

Table 2. Test accuracy on long-tailed ColorMNIST and Flowers102 with $T(2) \times H_m$ equivariant CNNs. *P* and *VP* denote that their architecture includes Partial and VP convolutional layers, respectively. A ✓in the InstaAug column means the training is conducted with the augmentation of InstaAug.

Group	#Elems.	Partial	InstaAug	ColorMNIST	Flowers102
$T(2)$	1	-	-	71.0±0.2	64.6±0.3
			✓	70.5±0.6	66.1±1.5
$T(2)$ $\times H_m$	3	-	-	**87.1**±0.1	68.0±0.5
			✓	87.3±0.9	64.3±1.1
		P	-	61.4±0.8	67.2±1.5
		VP	-	**85.4**±1.0	**69.4**±0.6
	6	-	-	88.7±0.3	65.0±0.7
			✓	87.9±0.3	62.2±0.8
		P	-	63.5±0.6	66.8±0.7
		VP	-	**88.4**±1.1	**69.3**±0.4

Model architecture for $T(2) \times H_m$. The product group $T(2) \times H_m$ includes translations and color-shifts. Color Equivariant Convolutions (CEConv) (Lengyel et al., 2023) extend the regular CNN, which has $T(2)$ equivariance, to have H_m equivariance. Similar to CEConv, we build ResNet18 consisting of input-aware partial group convolutions of CEConv. We apply the input-aware partial convolution on the last two blocks out of 7 blocks, and the other blocks are all CEConvs with full equivariance. Since CEConv utilizes discrete group elements (3 in Lengyel et al. (2023)) in the Hue spaces, we set $q(u|f)$ as the straight-through distribution as proposed in Equation Eq. 19. For Flowers102, we also use CEConv to build the fully equivariant G-CNN, but we construct it as the hybrid network consisting of both CEConv and the regular convolutions as in Lengyel et al. (2023).

Long-tailed colorMNIST. Long-tailed colorMNIST is a classification task comprising 30 classes of colored digit images. In this task, digits are presented in three different colors against a gray background, requiring classification based on both digits {0,1,2,3,4,5,6,7,8,9} and RGB colors {red, green, blue}. The distribution of samples per class follows a power law, leading to a significant class imbalance with some classes having substantially more samples than others. Since colorMNIST implies full equivariance for color-shift, as shown in the second last column of Table 2, CEConv (3rd and 7th rows) shows powerful performance compared to the other baselines, including Partial G-CNN (5th and 9th rows) and InstaAug (4th and 8th rows). Partial G-CNN suffers from unstable training in the discrete groups, thereby it shows poor performance. VP G-CNN (6th and 10th rows), however, provides competitive results, especially when the group elements are 6, which proves the robustness of our model even in the fully equivariant task.



(a) Barberton Daisy **(b)** Snapdragon

Figure 5. The x-axis represents the magnitude of the shift in the Hue space of the input, while the y-axis represents the model's confidence for the corresponding class. The image at zero hue-shift represents the original image. (a) Barberton Daisy exhibits partial equivariance because a shift of -0.17 or 0.17 overlaps with other flowers, while (b) Snapdragon demonstrates full equivariance owing to its distinctive appearance.

Flowers102. This dataset consists of 102 different categories of flowers, with each category containing between 40 and 258 images. Each image is labeled with the corresponding category of flower it depicts. As explained in Fig. 1b, color-shifts of some flowers cause confusion to classifiers, and the range of color-shifts that avoids confusion is non-trivial for each dataset. As seen in the last column of Table 2, since G-CNN (3rd and 7th rows) is constructed as a hybrid network, it still works well in such partially equivariant data. On the other hand, Partial G-CNN (5th and 9th rows) still performs poorly due to their instability. Finally, VP G-CNN (6th and 10th rows) significantly outperforms the baselines because of the input-aware partial equivariance and the improved $q(u|f)$ distribution design in discrete groups.

Learned invariance. To verify that our model learns the partial equivariance correctly, we analyze the confidence distribution across the magnitude of the color-shift. Fig. 5 exhibits the model's confidence for each color-shifted image, and the range of the color-shift is described in $[-0.5, 0.5]$. A -0.5 or 0.5 shift produces a complementary color of an image. For Barberton Daisy, the model shows equivariance almost only at 0 shift, while it shows full equivariance for Snapdragon. As explained in Fig. 1b, if we shift that flower with a -0.17 magnitude, which converts it to purple, it looks like Osteospermum. Conversely, if we shift it with a +0.17 magnitude, which converts it to yellow, it looks like sunflowers, also included in Flowers102. Hence, Barberton Daisy requires no equivariance with respect to the color-shift. On the other hand, due to its unique appearance, Snapdragon is identifiable even if we change its color. Therefore, Snapdragon requires full equivariance, and VP G-CNN captures it properly. The plots for some other flowers can be checked in Fig. 7 of Appendix D.

Unfortunately, assessing learned equivariance over the entire dataset through a few metrics can be challenging. Hence,

Table 3. Test uncertainty metrics on Flowers102.

Metrics	G-CNN	Partial G-CNN	VP G-CNN (Ours)
NLL (↓)	0.0171	0.0346	**0.0121**
BS (↓)	0.0042	0.0086	**0.0035**

we plotted the minimum and maximum invariance error of the trained model for each class in Flowers102 in Fig. 8 of Appendix D. In these plots, the height of bars represents the minimum (or maximum) invariance error across data in each class. Tall bars indicate flowers that require non-equivariance, while short bars represent flowers that require strong equivariance. In the minimum plot, although Tiger Lily (6th from the left) shows non-equivariance, Moon Orchid (7th) and Snapdragon (11th) exhibit relatively strong equivariance. Note that the maximum plot shows that every class includes at least one instance of non-equivariance, indicating that not every flower in a class requires strong equivariance despite their appearance being relatively unique.

Calibration performance. We further compare the calibration performance to evaluate the effectiveness of Variational inference in Flower102. We compute two uncertainty metrics: negative log-likelihood (NLL) and Brier score (BS). NLL represents the discrepancy between the predicted distribution and the actual distribution of the data, quantifying how well the model's predictions match the observed data. On the other hand, BS measures the average squared difference between predicted probabilities and the actual outcomes; lower scores indicate that the predicted probabilities are closer to the actual outcomes. As shown in Table 3, utilizing Variational inference, VP G-CNN achieves lower NLL and BS scores compared to other methods. This indicates that variational inference is effective not only with respect to accuracy but also in uncertainty quantification.

Stability of proposed discrete distribution. We compared two discrete distributions for $p(u|f)$ over training time: the Gumbel-Softmax of Partial G-CNN and the Novel Distribution of VP G-CNN. For the same architecture based on VP CEResNet in the Flowers102 task, we only altered the distribution and compared them. That is, the Gumbel-Softmax distribution is also designed to be input-aware by predicting the parameters from the encoder r_ϕ as depicted in Fig. 9 of Appendix F.3. In each plot, every point represents the probability of each group element u_1, u_2, u_3 sampled from $p(u|f)$, and the x-axis denotes the training epochs. For Gumbel-Softmax, the probabilities of each group element frequently vary even at the end of training, while the novel distribution exhibits converged probability distributions (1/3,1/3,1/3) after 300 epochs with minor variations at 575 epochs.

Computational cost. Since our method requires an extra encoder r_ϕ in a few layers to compute the group distribution, additional computational cost is inevitable. Table 5 of Appendix E is a table comparing the computational cost across different methods, in terms of the number of parameters (#Params) and FLOPs, with CEResNet set as a reference value of 1. CEResNet consists of 1 linear layer, 4 CE residual blocks, and 1 initial CEConv. In our method (VP CEResNet on Flowers102), we replaced one head-side CE residual block (consisting of 3 CEConvs) and one tail-side CEConv with a VP CE residual block and single VP CEConv, respectively. As observed in Table 5, while the number of parameters slightly increases due to the encoder r_ϕ utilizing only 1D convolutions, the additional FLOPs are negligible compared to those of CEResNet and Partial CEResNet.

Additional comparisons. Furthermore, we conduct one additional experiment on CIFAR100, as depicted in Table 6 of Appendix F.1, and independently compare our method with another automatic augmentation baseline, AdaAug (Cheung & Yeung, 2022), on Flowers102 as shown in Table 7 of Appendix F.2. We demonstrate that our method competes effectively with other baselines on CIFAR100 and outperforms AdaAug on Flowers102.

6. Conclusion

We have introduced a new partially equivariant convolution designed to handle partial equivariance encountered in real-world datasets, particularly for groups of rotations and color-shifts. As observed in datasets like MNIST or Flowers102, partial equivariance must be determined based on the input. Unlike the previous Partial G-CNN methods, our approach, named VP G-CNN, learns the appropriate equivariance level for the given input by designing the output group element's distribution in the convolution to be input-aware. We interpret this distribution from a Variational inference perspective as an approximate posterior, enabling us to train the input-dependent distribution less prone to overfitting. Additionally, we have improved the training process, which was unstable in Partial G-CNN, by redesigning the distribution of group elements in the discrete group. We validated our approach on one toy dataset and three real-world datasets, including a fully Equivariant dataset, and confirmed that it captures appropriate partial equivariance for the input while outperforming baseline methods in terms of color equivariance. As an extension of this work, we anticipate that our approach could serve as a guide for constructing a group equivariant network architecture capable of automatically determining the necessary equivariance to a subgroup from a given group, such as the general linear group.

Acknowledgements

This research was partly supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT)(No.RS-2019-II190075, Artificial Intelligence Graduate School Program(KAIST); No.2022-0-00184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics; No.2022-0-00713, Meta-learning Applicable to Real-world Problems), and the National Research Foundation of Korea(NRF) grants funded by the Korea government(MSIT)(No. 2022R1A5A7083908; No. RS-2023-00279680).

Impact Statement

This paper does not include any ethical issues and bad societal consequences. This paper presents a new partial group convolution for mainly image classifications regarding mathematical group symmetry present in data, which does not cause ethical or social issues.

References

Benton, G. W., Finzi, M., Izmailov, P., and Wilson, A. G. Learning invariances in neural networks from training data. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020. 3

Cheung, T. and Yeung, D. Adaaug: Learning class- and instance-adaptive data augmentation policies. In *International Conference on Learning Representations (ICLR)*, 2022. 6, 9, 15

Cohen, T. and Welling, M. Group equivariant convolutional networks. In *Proceedings of The 33rd International Conference on Machine Learning (ICML 2016)*, 2016. 1, 6

Cohen, T. S. and Welling, M. Steerable cnns. In *International Conference on Learning Representations (ICLR)*, 2017. 1, 6

Cubuk, E. D., Zoph, B., Mané, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation strategies from data. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 113–123. Computer Vision Foundation / IEEE, 2019. 6

Finzi, M., Stanton, S., Izmailov, P., and Wilson, A. G. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *Proceedings of The 37th International Conference on Machine Learning (ICML 2020)*, 2020. 7

Finzi, M., Benton, G., and Wilson, A. G. Residual pathway priors for soft equivariance constraints. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, 2021. 6

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '16, pp. 770–778. IEEE, June 2016. doi: 10.1109/CVPR.2016.90. URL http://ieeexplore.ieee.org/document/7780459. 7

Ho, D., Liang, E., Chen, X., Stoica, I., and Abbeel, P. Population based augmentation: Efficient learning of augmentation policy schedules. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, 2019. 6

Kim, H., Lee, H., Yang, H., and Lee, J. Regularizing towards soft equivariance under mixed symmetries. In *Proceedings of The 40th International Conference on Machine Learning (ICML 2023)*, 2023. 6

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y. (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114. 5

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 7

LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010. 7

Lengyel, A., Strafforello, O., Bruintjes, R.-J., Gielisse, A., and van Gemert, J. Color equivariant convolutional networks, 2023. 3, 6, 8

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=S1jE5L5gl. 5

Miao, N., Rainforth, T., Mathieu, E., Dubois, Y., Teh, Y. W., Foster, A., and Kim, H. Learning instance-specific augmentations by capturing local invariances. In *Proceedings of The 40th International Conference on Machine Learning (ICML 2023)*, 2023. 6, 7

Nilsback, M. and Zisserman, A. Automated flower classification over a large number of classes. In *Sixth Indian Conference on Computer Vision, Graphics & Image Processing, ICVGIP 2008, Bhubaneswar, India, 16-19 December 2008*, pp. 722–729. IEEE Computer Society, 2008. doi: 10.1109/ICVGIP.2008.47. URL https://doi.org/10.1109/ICVGIP.2008.47. 7

Romero, D. W. and Lohit, S. Learning partial equivariances from data. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022. 1, 3, 5, 6, 7

Romero, D. W., Kuzina, A., Bekkers, E. J., Tomczak, J. M., and Hoogendoorn, M. Ckconv: Continuous kernel convolution for sequential data. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=8FhxBtXSl0. 1, 7

Singhal, U., Esteves, C., Makadia, A., and Yu, S. X. Learning to transform for generalizable instance-wise invariance. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 6188–6198. IEEE, 2023. 6

Sitzmann, V., Martel, J., Bergman, A., Lindell, D., and Wetzstein, G. Implicit neural representations with periodic activation functions. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 7462–7473. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/53c04118df112c13a8c34b38343b9c10-Paper.pdf. 7

van der Ouderaa, T., Romero, D. W., and van der Wilk, M. Relaxing equivariance constraints with non-stationary continuous filters. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022. 2

Wang, R., Walters, R., and Yu, R. Approximately equivariant networks for imperfectly symmetric dynamics. In *Proceedings of The 39th International Conference on Machine Learning (ICML 2022)*, 2022. 2, 6

Weiler, M. and Cesa, G. General e(2)-equivariant steerable cnns. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019. 1, 6

Zhou, F., Li, J., Xie, C., Chen, F., Hong, L., Sun, R., and Li, Z. Metaaugment: Sample-aware data augmentation policy learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, 2021. 6

A. Proofs

A.1. Proof of Proposition 3.1

Using change of variables, we expand the convolution integral when the group action $\mathcal{L}_g$ acts on feature f.

$$(k * \mathcal{L}_g f)(u) = \int_{v\in G} q(u|\mathcal{L}_g f)k(v^{-1}u)f(g^{-1}v)\mathrm{d}\mu_G(v) \tag{20}$$

$$= \int_{v'\in G} q(u|\mathcal{L}_g f)k(v'^{-1}g^{-1}u)f(v')\mathrm{d}\mu_G(v'). \tag{21}$$

On the other hand, the convolution when $\mathcal{L}_g$ acts on the output of the convolution is

$$\mathcal{L}_g(k * f)(u) = \int_{v\in G} q(g^{-1}u|f)k(v^{-1}g^{-1}u)f(v)\mathrm{d}\mu_G(v). \tag{22}$$

The equivariance error is represented by the difference between the group action on the input and on the output. Then, we bound the l_2-norm of the equivariance error using the Cauchy-Schwarz inequality:

$$\|(k * \mathcal{L}_g f)(u) - \mathcal{L}_g(k * f)(u)\|_2^2 = \left\| \int_{v\in G} \Big[q(u|\mathcal{L}_g f) - q(g^{-1}u|f)\Big]k(v^{-1}g^{-1}u)f(v)\mathrm{d}\mu_G(v) \right\|_2^2 \tag{23}$$

$$\leq \int_{v\in G} \left\|q(u|\mathcal{L}_g f) - q(g^{-1}u|f)\right\|_2^2 \mathrm{d}\mu_G(v) \int_{v\in G} \left\|k(v^{-1}g^{-1}u)f(v)\right\|_2^2 \mathrm{d}\mu_G(v) \tag{24}$$

$$\leq \begin{cases} 0, & f \in C, \\ \epsilon^2 \cdot \int_G \left\|k(v^{-1}g^{-1}u)f(v)\right\|_2^2 \mathrm{d}\mu_G(v), & f \in \mathcal{F}\setminus C \end{cases} \tag{25}$$

According to the partial equivariance of $q(u|f)$ as in Proposition 3.1, the error becomes zero when $f \in C$. Conversely, when $f \in \mathcal{F}\setminus C$, since the kernel k and input f are bounded, the equivariance error is bounded by a certain value ϵ'. The proof for the lifting convolutions is the same because the integral in Eq. 25 is still bounded even when integrated over E instead of G

A.2. Proof of Proposition 3.2

By mathematical induction, it is enough to show that a G-CNN with two convolution layers, one fully equivariant and another partially equivariant, is itself partially equivariant. For *fully* G-equivariant $(k_2 * f)(t)$, *partially* G-equivariant $(k_1 * f)(u)$, G-equivariant & L-Lipschitz continuous activation functions σ, and the bounded kernel k_2, the following equivariance error is bounded as:

$$\|(k_2 * \sigma(k_1 * \mathcal{L}_g f))(t) - \mathcal{L}_g(k_2 * \sigma(k_1 * f))(t)\|_2^2 \tag{26}$$

$$= \left\| \big(k_2 * \sigma\big[\mathcal{L}_g(k_1 * f) - (k_1 * \mathcal{L}_g f)\big]\big)(t) \right\|_2^2 \tag{27}$$

$$= \left\| \int_{v\in G} k_2(v^{-1}t)\sigma\big[\mathcal{L}_g(k_1 * f) - (k_1 * \mathcal{L}_g f)\big](v)\mathrm{d}\mu_G(v) \right\|_2^2 \tag{28}$$

$$\leq \int_{v\in G} \left\|k_2(v^{-1}t)\right\|_2^2 \mathrm{d}\mu_G(v) \int_{v\in G} \left\|\sigma\big(\mathcal{L}_g(k_1 * f) - (k_1 * \mathcal{L}_g f)\big)(v)\right\|_2^2 \mathrm{d}\mu_G(v) \tag{29}$$

$$\leq L^2 \int_{v\in G} \left\|k_2(v^{-1}t)\right\|_2^2 \mathrm{d}\mu_G(v) \int_{v\in G} \left\|\big(\mathcal{L}_g(k_1 * f) - (k_1 * \mathcal{L}_g f)\big)(v)\right\|_2^2 \mathrm{d}\mu_G(v). \tag{30}$$

Thus, the equivariance error is determined by the equivariance error of the partially equivariant convolution $k_1 * f$:

$$\left\|\big(\mathcal{L}_g(k_1 * f) - (k_1 * \mathcal{L}_g f)\big)(v)\right\|_2^2, \tag{31}$$

which means if the equivariance error of the partially equivariant network is zero, the equivariance error of the whole network is also zero.

On one hand, for *partially* G-equivariant $(k_2 * f)(t)$ and *fully* G-equivariant $(k_1 * f)(u)$ with the associability of the convolution,

$$\|(k_2 * \sigma(k_1 * \mathcal{L}_g f))(t) - \mathcal{L}_g(k_2 * \sigma(k_1 * f))(t)\| = \left\| \Big(k_2 * \mathcal{L}_g\big(\sigma(k_1 * f)\big)\Big)(t) - \mathcal{L}_g(k_2 * \sigma(k_1 * f))(t) \right\| \tag{32}$$

$$= \|(k_2 * \mathcal{L}_g f')(t) - \mathcal{L}_g(k_2 * f')(t)\|, \tag{33}$$

(a) The hyperparameter settings used to learn VP G-CNN.

Dataset	MNIST67-180	CIFAR10	ColoredMNIST	Flowers102
Batch Size	64	64	64	64
Epochs	300	300	1500	400
Optimizer	AdamW	AdamW	Adam	Adam
Learning Rate	0.001	0.001	0.001	0.0002
Optimizer (r_ϕ)	SGD	SGD	Adam	AdamW
Learning Rate (r_ϕ)	0.001	0.001	0.0001	0.0001
Weight Decay	0.001	0.001	0.00001	0.00001
Architecture Base	ResNet18	ResNet18	7-layer CNN	7-layer CNN
# of Conv. Layers	3	5	7	7
Normalization	BatchNorm	BatchNorm	BatchNorm	BatchNorm
Kernel Network	SIREN	SIREN	-	-

(b) Hyperparameters for baseline models

Dataset	ColoredMNIST	Flowers102	CIFAR10
Batch Size	256	64	64
Epochs	1500	300	300
Optimizer	Adam	Adam	Adam
Learning Rate	0.001	0.001	0.001
Weight Decay	0.00001	0	0.0001
Architecture Base	7-layer CNN	ResNet18	ResNet18
Kernel Network	-	-	SIREN
Optimizer*	Adam	Adam	AdamW
Learning Rate*	0.0001	0.0001	0.0001
Optimizer**	Adam	Adam	Adam
Learning Rate**	0.0001	0.0001	0.001
Entropy regularization **	0.0001	0.0001	0.0001

* PG-CNN specific parameters

** InstaAug specific parameters

Table 4. Hyperparameter settings.

where $f' = \sigma(k_1 * f)$, which corresponds to the equivariance error of $k_2 * f'$. The proof is still valid when k_1 or k_2 are the lifting convolution because it is also G-equivariant and it has bounded kernels.

B. Hyperparameters

We list hyperparameters used for training VP G-CNN and the baselines in every dataset in Table 4.

C. Architecture of Encoder e_ϕ

We use light-weighted encoder e_ϕ, consisting of two global average pooling layers, two one-dimensional convolution, and one linear layer. The encoder is illustated in Fig. 6. The parentheses above the network indicate dimensions of input tensors in each layer. C, C', C'', C'' denotes the number of channels, G is the number of group elements, H is height of the features, and W is width of the features.

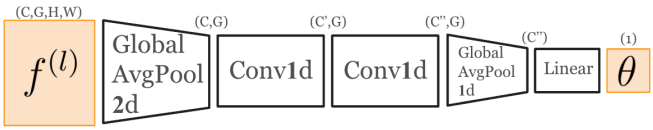
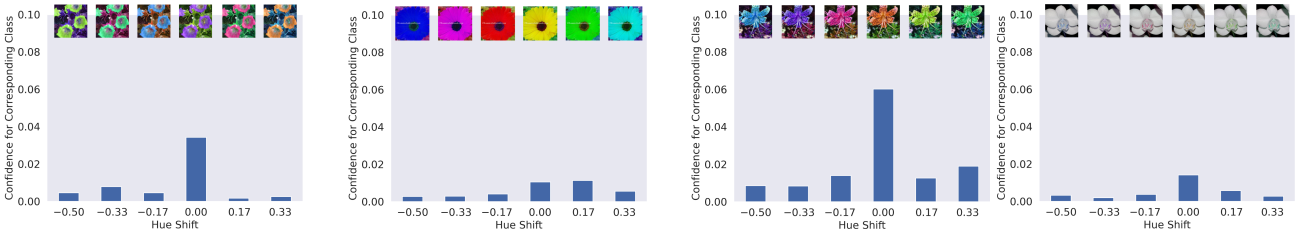


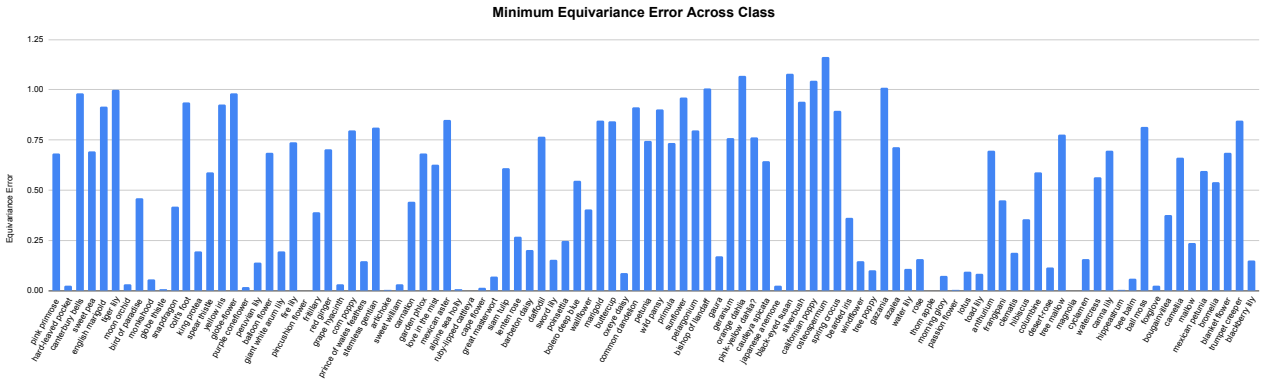
Figure 6. Encoder network $e_\phi(f)$ architecture.

D. Additional Plots in Flowers102

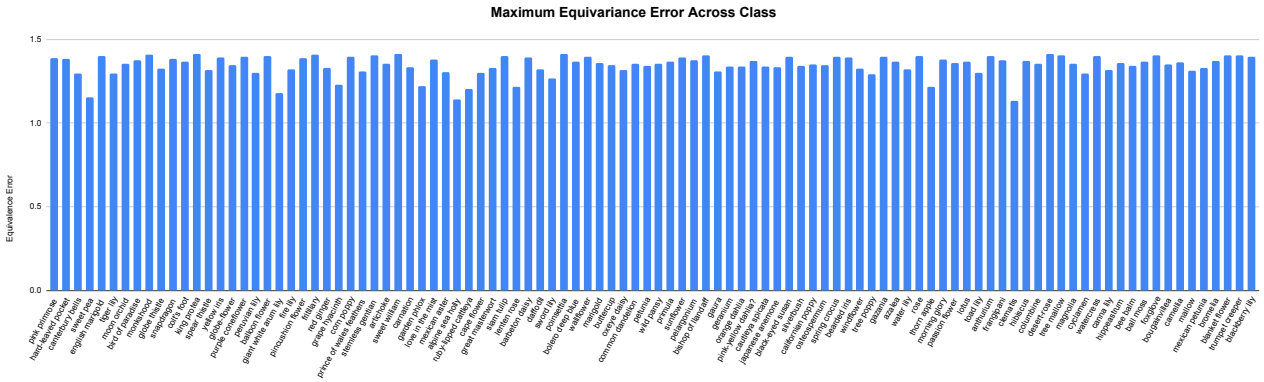


(a) Canterbury Bells **(b)** English Marigold **(c)** Tiger Lily **(d)** Moon Orchid

Figure 7. Confidence across color-shifts of input image in Flowers102.



(a) Minimum



(b) Maximum

Figure 8. Mimimum and Maximum equivariance errors across classes in Flowers102. Tall bars indicate flowers that require non-equivariance, while short bars represent flowers that require strong equivariance.

E. Computational Cost

Since our method requires an extra encoder r_ϕ in a few layers to compute the group distribution, additional computational cost is inevitable. Below is a table comparing the computational cost across different methods, in terms of the number of parameters (#Params) and FLOPs, with CEResNet set as a reference value of 1. CEResNet consists of 1 linear layer, 4 CE residual blocks, and 1 initial CEConv. In our method (VP CEResNet on Flowers102), we replaced one head-side CE residual block (consisting of 3 CEConvs) and one tail-side CEConv with a VP CE residual block and single VP CEConv, respectively.

Table 5. Computational cost comparison of models used in Flowers102. * denotes the model reported in Table 2. † indicates a model whose layers are all VP.

Metrics	CEResNet	ResNet w/ InstaAug	Partial CEResNet	Partial CEResNet w/ InstaAug	VP CEResNet*	VP CEResNet†
#Params (↓)	×1.0000	×1.9628	×1.0000	×1.9814	×1.2416	×1.3211
FLOPs (↓)	×1.0000	×0.3161	×1.0000	×1.1581	×1.0006	×1.0007

As observed in Table 5, while the number of parameters slightly increases due to the encoder r_ϕ utilizing only 1D convolutions, the additional FLOPs are negligible compared to those of CEResNet and Partial CEResNet.

F. Additional Experiments

F.1. Evaluation in CIFAR-100

We conducted experiments using the partially $SE(2)$-equivariant ResNet on the CIFAR100 dataset. As you can see in Table 6, our model still performs competitively on CIFAR100. However, the improvement is not surprising compared to Partial G-CNN because CIFAR100 is not necessarily trained to be aware of the level of rotation invariance for each data point. Furthermore, our performance remains constrained at around 50% as we utilized the same model architecture as in the experiments conducted in the Partial G-CNN paper, resulting in weaker performance compared to contemporary models.

Table 6. Test accuracy in CIFAR100.

	$SE(2)$-ResNet	Partial $SE(2)$-ResNet	VP $SE(2)$-ResNet (ours)
Test Accuracy (%;↑)	52.40	57.02	**57.67**

F.2. Comparison with AdaAug

We compared our method with AdaAug (Cheung & Yeung, 2022) on Flowers102. AdaAug is a method for learning adaptive data augmentation policies in a class-dependent and potentially instance-dependent manner. It trains the policy network that determines the augmentations via the validation loss evaluated by the augmented validation data and the classifier, while training the classifier with the augmented training data. Here is the test accuracy comparison with different baselines used with AdaAug that generates adaptive color-shift augmentations as shown in Table 7.

Table 7. Test accuracy comparison with AdaAug in Flowers102.

	ResNet w/ AdaAug	CEResNet w/ AdaAug	Partial CEResNet w/ AdaAug	VP CEResNet (ours)
Test Accuracy (%;↑)	64.70	63.51	68.45	**69.40**

Although AdaAug is fairly a good method, our method still outperforms it due to its architectural inductive bias. Furthermore, our method does not demand the validation dataset.

F.3. Stability of Proposed Discrete Distribution

We compared two discrete distributions for $p(u|f)$ over training time: the Gumbel-Softmax of Partial G-CNN and the Novel Distribution of VP G-CNN. For the same architecture based on VP CEResNet in the Flowers102 task, we only altered the distribution and compared them. That is, the Gumbel-Softmax distribution is also designed to be input-aware by predicting the parameters from the encoder r_ϕ as depicted in Fig. 9.

In each plot, every point represents the probability of each group element u_1, u_2, u_3 sampled from $p(u|f)$, and the x-axis denotes the training epochs. For Gumbel-Softmax, the probabilities of each group element frequently vary even at the end of training, while the novel distribution exhibits converged probability distributions (1/3,1/3,1/3) after 300 epochs with minor variations at 575 epochs.

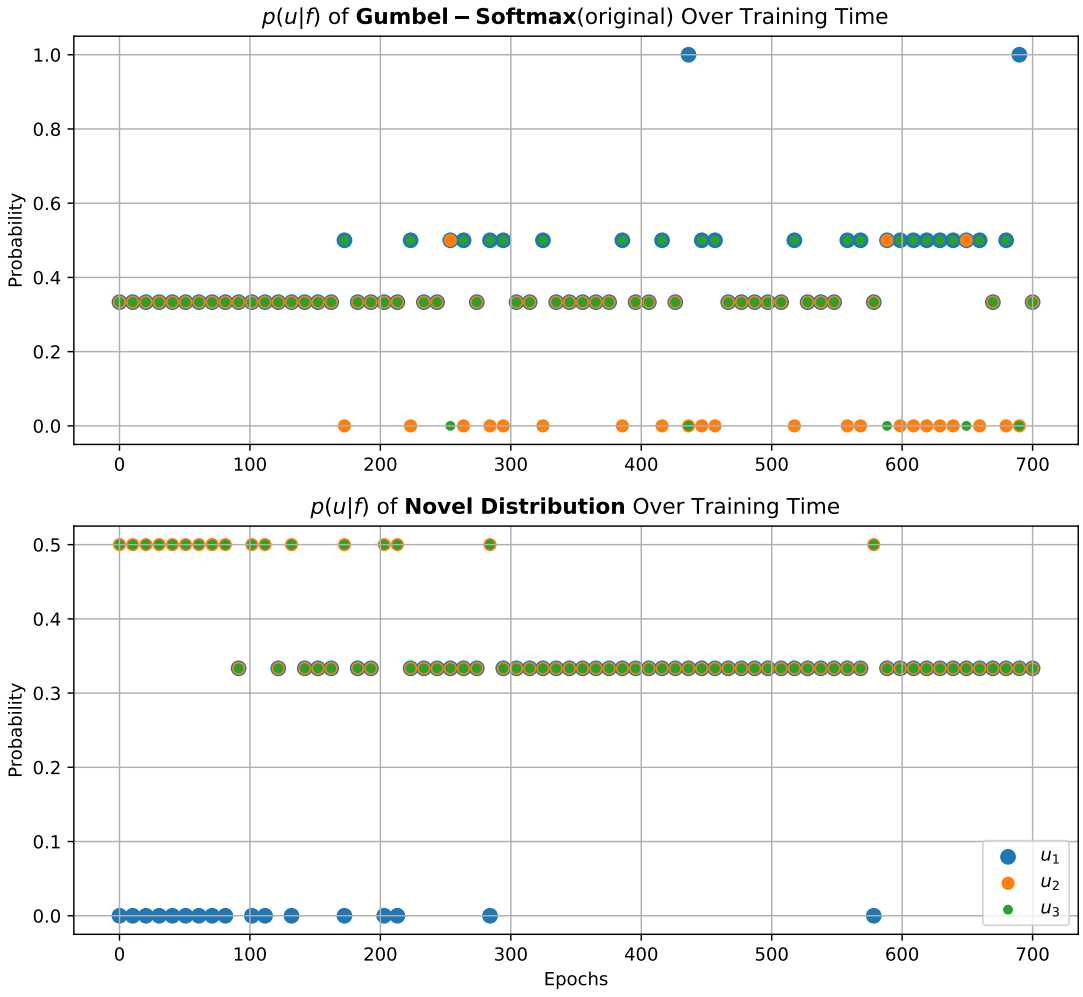


Figure 9. Gumbel-Softmax (top) vs. Novel Distribution (bottom)