Affine Equivariant Networks Based on Differential Invariants

Anonymous CVPR submission

Paper ID 2786

Abstract

Convolutional neural networks benefit from translation equivariance, achieving tremendous success. Equivariant networks further extend this property to other transformation groups. However, most existing methods require discretization or sampling of groups, leading to increased model sizes for larger groups, such as the affine group. In this paper, we build affine equivariant networks based on differential invariants from the viewpoint of symmetric PDEs, without discretizing or sampling the group. To address the division-by-zero issue arising from fractional differential invariants of the affine group, we construct a new kind of affine invariants by normalizing polynomial relative differential invariants to replace classical differential invariants. For further flexibility, we design an equivariant layer, which can be directly integrated into convolutional networks of various architectures. Moreover, our framework for the affine group is also applicable to its continuous subgroups. We implement equivariant networks for the scale group, the rotation-scale group, and the affine group. Numerical experiments demonstrate the outstanding performance of our framework across classification tasks involving transformations of these groups. Remarkably, under the out-of-distribution setting, our model achieves a 3.37% improvement in accuracy over the main counterpart affConv on the affNIST dataset.

1. Introduction

The success of convolutional neural networks (CNNs) can be attributed to their utilization of translation symmetry. This profound insight emphasizes the significance of incorporating symmetry priors into the design of models. With this insight, equivariant networks extend the exploitation of more symmetries, leading to great improvement on performance and efficiency. Development of equivariant networks begins with the approach of group convolutions, which views feature maps as functions defined on a group and conducts convolution operation over the group [3, 63]. Further advancements in equivariant networks effectively achieve equivariance on the Euclidean group and its subgroups [4, 11, 12, 61, 62, 67, 68]. However, existing methods have certain limitations when dealing with more complicated groups. One representative group is the affine group. While group convolutions typically require discretization of continuous groups, it becomes impractical for the affine group due to its high dimension. Finzi et al. [16] conduct group convolutions by sampling from Haar measure on the group. But it relies on easy access to Haar measure, which is unsuitable for the affine group. Recently, MacDonald et al. [37] overcome the limitation by computing the integral on the Lie algebra, thereby obtaining the affine equivariant model, affConv. Nevertheless, this approach still requires sampling from the group and encounters an exponential growth in memory requirements as the number of convolutional layers increases.

In another branch, some works adopt partial differential operators (PDOs) to design equivariant networks [25, 27, 49]. They achieve equivariance on Euclidean groups by imposing constraints on the weights of PDOs. In fact, specific functional combinations of partial derivatives remain constant under group actions — a concept known as “differential invariants.” Under the guidance of the differential invariant theory, Liu et al. [33, 34] design a shift and rotationally equivariant system of learnable partial differential equations (PDEs) with linear combinations of differential invariants. The evolution process of PDEs can be used to solve multiple vision problems. Subsequent works extend the approach to more tasks and further develop the models [14, 46, 48, 77]. However, these efforts also concentrate on equivariance of Euclidean groups, and the full potential of differential invariants in handling more general groups has yet to be explored.

In this paper, we construct affine equivariant networks based on differential invariants from the viewpoint of symmetric PDEs, without discretizing or sampling the group. Inspired by learnable PDEs [33, 34], we regard image data as smooth functions on the 2D plane and model the equivariant inference process of feature extraction as an evolving system governed by symmetric PDEs. The differential invariant theory reveals that, given a group G, a PDE ad-

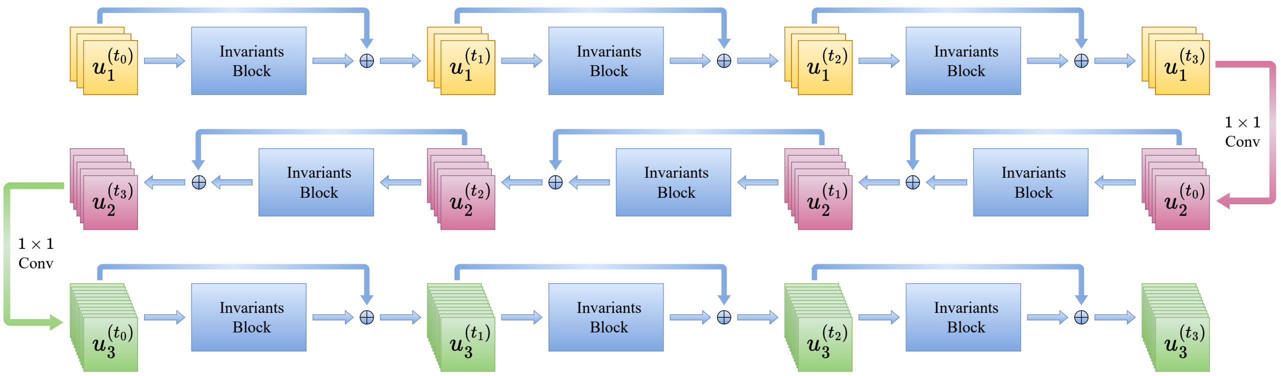

Figure 1. InvarPDEs-Net consists of iterative processes of multiple symmetric PDEs constructed with invariants. We link them by linearly combining the output of one PDE to match the dimension of the subsequent one, which can be implemented with 1×1 convolutions.

mits G as a symmetry group if and only if the PDE consists of fundamental differential invariants of the group G [41]. To construct learnable symmetric PDEs, we can precompute a complete set of fundamental differential invariants of the given group, and then employ multilayer perceptrons (MLPs) to combine them into equations, leveraging the universal approximation capability of neural networks. However, differential invariants of the affine group may take the form of fractional polynomials, potentially leading to the division-by-zero issue in practice. Nonetheless, we notice that affine differential invariants can be represented by polynomial relative differential invariants. Building on this observation, we propose a technique to construct a new kind of affine invariants by normalizing polynomial relative differential invariants with a special norm, thus replacing the fundamental differential invariants. These new invariants not only avoid the division-by-zero issue but also retain more information. To discretize the symmetric PDE (not the affine group), we approximate the temporal derivatives by forward difference and approximate the spatial derivatives by Gaussian derivatives, resulting in an iterative process that can be viewed as a feed-forward deep equivariant network.

To equip our network with adaptability to varying channel numbers, similar to other modern networks, we sequentially stack iterative processes of multiple learnable symmetric PDEs with different dimensions. We connect them by linearly combining the output channels of one PDE to match channel numbers of the subsequent PDE. Thus, the output of one PDE can serve as the input of the subsequent one. This approach allows us to create an equivariant network with varying channel numbers, which consists of multiple symmetric PDEs constructed with invariants. We name it *InvarPDEs-Net* (see Figure 1). For further flexibility, we extract a block from the iterative process and modify it into an equivariant layer, offering the freedom to specify input and output channel numbers. The layer can serve as a drop-in replacement for convolutional networks of various architectures. We name it *InvarLayer*. Our framework for constructing equivariant networks of the affine group is also applicable to its continuous subgroups. We implement equivariant networks for the scale group, the rotation-scale group and the affine group. Empirical experiments on classification tasks involving transformations of these groups demonstrate the outstanding performance of our method.

We summarize our main contributions as follows:

- From the viewpoint of symmetric PDEs, we construct affine equivariant networks based on differential invariants. It is the first time that affine equivariance for networks is achieved without discretizing or sampling the group. Consequently, we overcome the limitation on network depth encountered by affConv [37].
- We propose a technique to construct a new kind of affine invariants by normalizing polynomial relative differential invariants with a special norm, which can be incorporated into our networks and enhance numerical stability.
- For further flexibility, we also design an equivariant layer, InvarLayer, which serves as a drop-in replacement for convolutional networks of various architectures.
- Our framework for constructing affine equivariant networks is also applicable to its continuous subgroups. We implement equivariant networks for three non-Euclidean groups: the scale group, the rotation-scale group and the affine group. Extensive experiments demonstrate the outstanding performance of our framework. Particularly, we achieve a 3.37% improvement in accuracy compared with affConv [37] on the public affNIST[1] dataset under the out-of-distribution setting.[2]

2. Related works

Currently there are two mainstream methods for constructing group equivariant networks. One approach stems from Cohen and Welling [3], which treats feature maps as functions defined on a group. Some works extend this approach to subgroups of Euclidean groups on various do-

[1] https://www.cs.toronto.edu/ tijmen/affNIST/
[2] Our code will be publicly available upon acceptance.

mains, such as rotation on the 2D plane [1, 24, 31, 32], rotation over the 3D space [11, 12, 67, 68], symmetries on spheres [5, 9, 10] and surfaces [6, 7]. Besides, with some proper approximations, some works utilize this approach to handle non-compact groups, such as the scale group [47, 53, 66, 69, 78] and Lie groups [16, 37]. The other approach follows the steerable CNNs framework [4, 61–63], which views feature maps as vector fields. This approach has also been further applied to subgroups of Euclidean groups on the 2D plane [20, 57, 58, 70, 75], the 3D space [17, 62], and spheres [13, 64]. In addition, similar to the first approach, this approach has also been extended to the scale group [19, 40, 52] and the rotation-scale group [18, 55, 74].

Besides the above approaches, some works utilize PDOs with learnable coefficients to design equivariant neural networks on 2D plane [25, 27, 49]. Besides, PDOs can also be applied to spheres [28, 50], volumetric data [51] and surfaces [65]. Differential invariants, as a specialized form of partial differential operators, hold a distinctive role in the field of image processing [22, 39, 44, 56, 59]. The equivariant method of moving frames offers an elegant tool to derive differential invariants of a given group [15, 42, 43]. Wang et al. [59] provide a practical and simplified approach for deriving relative affine differential invariants. Theoretical links reveal that differential invariants are closely intertwined with symmetric PDEs [41]. Building upon this connection, Liu et al. [33, 34] design a shift and rotationally equivariant system comprised of learnable PDEs with linear combinations of fundamental differential invariants. Subsequently, some works apply learnable PDEs to feature learning and extensive vision tasks [14, 77]. Some works further develop the approach and create equivariant networks on Euclidean groups [46, 48]. Additionally, some researchers draw inspiration from PDEs to design deep convolutional networks [35, 36].

3. Theoretical framework

In this section, we propose a new framework based on differential invariants to achieve equivariance of the affine group. We also describe some extensions of the framework and how they can be implemented.

3.1. Basic concepts and notations

To explicitly present the proposed method and theoretical derivation in the following, we first give a preliminary introduction to concepts involved and notations used.

Inputs and intermediate feature maps of neural networks can be modeled as vector functions defined on a continuous domain, *e.g.* the 2D plane for image data. Each layer of the network thereby can be regarded as an operator. In this paper, we study $\mathcal{F} = \{\mathbf{u} \big| \mathbf{u} : X \to \mathbb{R}^n\}$ as the set of bounded smooth functions defined on $X = \mathbb{R}^2$. Given a group G acting on X, it naturally induces a group action on $\mathcal{F}$, *i.e.* $(g \cdot \mathbf{u})(\mathbf{x}) = \mathbf{u}(g^{-1} \cdot \mathbf{x})$, where $g \in G, \mathbf{x} \in X, \mathbf{u} \in \mathcal{F}$.

Equivariance indicates that the output of a mapping transforms in accordance with transformation of the input.

Definition 1 *Let G be a group acting on function sets $\mathcal{F}$ and $\mathcal{F}'$. An operator $\Psi : \mathcal{F} \to \mathcal{F}'$ is said to be **equivariant** with respect to G, if $\Psi[g \cdot \mathbf{u}] = g \cdot \Psi[\mathbf{u}], \forall g \in G, \mathbf{u} \in \mathcal{F}$.*

Transitivity is an important property of equivariance. As a result, when equivariant operators are composed together, they still possess equivariance.

The concept of invariants is crucial and widely applied in various fields. Invariants extract some symmetric information and remain constant on the orbits of group actions. Here we give the definition of invariants below.

Definition 2 *Let G be a group acting on X, and $\mathcal{F} = \{\mathbf{u} \big| \mathbf{u} : X \to \mathbb{R}^n\}$ be a function set defined on X. An **invariant** of G is a map $\mathcal{I} : X \times \mathcal{F} \to \mathbb{R}$ such that $\forall \mathbf{u} \in \mathcal{F}, \mathbf{x} \in X, g \in G$,*

$$\mathcal{I}(g \cdot \mathbf{x}, g \cdot \mathbf{u}) = \mathcal{I}(\mathbf{x}, \mathbf{u}). \tag{1}$$

We call $\boldsymbol{\mathcal{I}} \triangleq (\mathcal{I}_1, ..., \mathcal{I}_k)^\top$ a k-dimensional invariant of G, if $\mathcal{I}_1, ..., \mathcal{I}_k$ are invariants of G.

Invariants under operation of postcomposition maintain the property of invariance, which can be formulated as follows:

Proposition 3 *Let $\boldsymbol{\mathcal{I}} : X \times \mathcal{F} \to \mathbb{R}^k$ be a k-dimensional invariant, and $\mathbf{h} : \mathbb{R}^k \to \mathbb{R}^{k'}$ be a k'-dimensional vector function. Then $\mathbf{h} \circ \boldsymbol{\mathcal{I}}$ is a k'-dimensional invariant.*

Intuitively, invariants and equivariant operators somehow both imply symmetry of group G. In fact, we can construct an equivariant operator with an invariant.

Proposition 4 *Let $\boldsymbol{\mathcal{I}} : X \times \mathcal{F} \to \mathbb{R}^k$ be a k-dimensional invariant of G, where $\mathcal{F} = \{\mathbf{u} \big| \mathbf{u} : X \to \mathbb{R}^n\}$. View $\boldsymbol{\mathcal{I}}(\cdot, \mathbf{u})$ as a k-dimensional function in $\mathcal{F}' = \{\mathbf{v} \big| \mathbf{v} : X \to \mathbb{R}^k\}$, and define an operator $\hat{\boldsymbol{\mathcal{I}}} : \mathcal{F} \to \mathcal{F}'$ such that*

$$\hat{\boldsymbol{\mathcal{I}}}[\mathbf{u}] \triangleq \boldsymbol{\mathcal{I}}(\cdot, \mathbf{u}). \tag{2}$$

Then $\hat{\boldsymbol{\mathcal{I}}}$ is equivariant.

Proof. $\forall \mathbf{u} \in \mathcal{F}, g \in G, x \in X$, we have

$$\begin{aligned}
\hat{\boldsymbol{\mathcal{I}}}[g \cdot \mathbf{u}](x) &= \boldsymbol{\mathcal{I}}(x, g \cdot \mathbf{u}) \\
&= \boldsymbol{\mathcal{I}}(g^{-1} \cdot x, \mathbf{u}) \\
&= \hat{\boldsymbol{\mathcal{I}}}[\mathbf{u}](g^{-1} \cdot x) \\
&= (g \cdot \hat{\boldsymbol{\mathcal{I}}}[\mathbf{u}])(x).
\end{aligned}$$

Therefore, $\hat{\boldsymbol{\mathcal{I}}}[g \cdot \mathbf{u}] = g \cdot \hat{\boldsymbol{\mathcal{I}}}[\mathbf{u}]$. □

It is worth noting that the equivariant operator $\hat{\boldsymbol{\mathcal{I}}}$ composed with a function $\mathbf{h}$ remains equivariant. Specifically, the operator $\mathbf{u} \mapsto \mathbf{h} \circ \hat{\boldsymbol{\mathcal{I}}}[\mathbf{u}]$ is actually equivalent to $\mathbf{u} \mapsto \hat{\boldsymbol{\mathcal{I}}}_{\mathbf{h}}[\mathbf{u}]$, where $\boldsymbol{\mathcal{I}}_{\mathbf{h}} \triangleq \mathbf{h} \circ \boldsymbol{\mathcal{I}}$ is still an invariant according to Proposition 3.

As a special type of invariants, a differential invariant is a quantity involving the derivatives of functions that remains unchanged under the prolongation of group actions.

Definition 5 *Let f be a smooth function and $\mathcal{I}(\mathbf{x}, \mathbf{u}) \triangleq f(\mathbf{x}, \mathbf{u}(\mathbf{x}), \nabla\mathbf{u}(\mathbf{x}), ..., \nabla^d\mathbf{u}(\mathbf{x})))$. If $\mathcal{I}$ is an invariant, we call $\mathcal{I}$ a d-th order* ***differential invariant****.*

As we always require translation invariance by default, it is sufficient to consider differential invariants in the form $\mathcal{I}(\mathbf{x}, \mathbf{u}) \triangleq f(\mathbf{u}(\mathbf{x}), \nabla\mathbf{u}(\mathbf{x}), ..., \nabla^d\mathbf{u}(\mathbf{x})))$, omitting the term $\mathbf{x}$ [34, 60]. According to the differential invariant theory [41], there are finite independent differential invariants up to the d-th order such that any d-th order differential invariant can be expressed by these differential invariants. We call them **fundamental differential invariants**.

In this paper, we focus on the affine group, which is ubiquitous in computer vision. The affine group consists of translation and invertible linear transformations. Denote the affine group as G, and any element $g \in G$ can be represented as $g = (\mathbf{A}, \mathbf{b})$, where $\mathbf{A} \in \mathbb{R}^{2\times 2}$ is invertible and $\mathbf{b} \in \mathbb{R}^2$. Then $g \in G$ acts on $\mathbb{R}^2$ via the following way: $g \cdot \mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{b}, \forall \mathbf{x} \in \mathbb{R}^2$.

3.2. From symmetric PDE to equivariant network

Inspired by learnable PDEs, we model the process of feature extraction as the evolution process governed by PDEs [14, 33–36, 77]. If the utilized PDE exhibits symmetry, the resultant feature extraction process will inherently possess equivariance [14, 33, 34, 77].

Let $\tilde{\mathcal{F}} = \{\tilde{\mathbf{u}} | \tilde{\mathbf{u}} : [0, T] \times \mathbb{R}^2 \to \mathbb{R}^n\}$ be a set of bounded smooth functions involving a temporal variable $t \in [0, T]$ and a spatial variable $\mathbf{x} \in \mathbb{R}^2$. We focus on high-dimensional evolutionary PDEs in the following form:

$$\frac{\partial \tilde{\mathbf{u}}}{\partial t} = \mathbf{F}\left(t, \mathbf{x}, \tilde{\mathbf{u}}, \nabla_{\mathbf{x}}\tilde{\mathbf{u}}, ..., \nabla^d_{\mathbf{x}}\tilde{\mathbf{u}}\right), \tag{3}$$

where $\mathbf{F}$ is a smooth function. We can view $\mathbf{u}^{(t)} \triangleq \tilde{\mathbf{u}}(t, \cdot)$ as a function in $\mathcal{F} = \{\mathbf{u} | \mathbf{u} : \mathbb{R}^2 \to \mathbb{R}^n\}$, and consider the group action of G on $\tilde{\mathbf{u}} \in \tilde{\mathcal{F}}$ following the same way of the group action on $\mathbf{u}^{(t)} \in \mathcal{F}$, *i.e.* $(g \cdot \tilde{\mathbf{u}})(t, \mathbf{x}) = \tilde{\mathbf{u}}(t, g^{-1} \cdot \mathbf{x})$. For a given symmetry group G, a PDE in the form (3) is called G-**symmetric** as long as if $\tilde{\mathbf{u}}$ is a solution, then $g \cdot \tilde{\mathbf{u}}$ is also a solution, for any $g \in G$.

According to the differential invariant theory [41], the PDE (3) is G-symmetric if and only if the right side of (3) is a function of differential invariants. Additionally, any differential invariant can be expressed as a function of fundamental differential invariants. Therefore, any G-symmetric PDE in the form (3) can be written as:

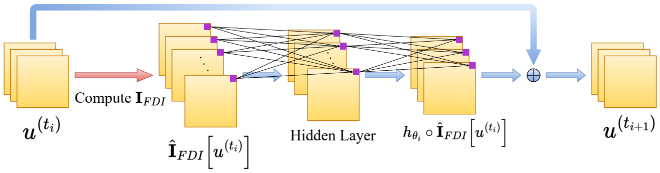
Compute $\mathbf{I}_{FDI}$
$\boldsymbol{u}^{(t_i)}$
$\hat{\mathbf{I}}_{FDI}\left[u^{(t_i)}\right]$
Hidden Layer
$h_{\theta_i} \circ \hat{\mathbf{I}}_{FDI}\left[u^{(t_i)}\right]$
$\boldsymbol{u}^{(t_{i+1})}$

Figure 2. Each iteration of the evolutionary PDE can be viewed as a layer of the network.

$$\frac{\partial \tilde{\mathbf{u}}}{\partial t}(t, \mathbf{x}) = \mathbf{H}\left(t, \mathcal{I}_1(\mathbf{x}, \mathbf{u}^{(t)}), ..., \mathcal{I}_k(\mathbf{x}, \mathbf{u}^{(t)})\right), \tag{4}$$

where $\mathbf{H}$ is a smooth function and $\mathcal{I}_i (i = 1, 2, ..., k)$ form a complete set of fundamental differential invariants. Denote $\boldsymbol{\mathcal{I}}_{FDI}$ as the concatenation of fundamental differential invariants, *i.e.* $\boldsymbol{\mathcal{I}}_{FDI} \triangleq (\mathcal{I}_1, ..., \mathcal{I}_k)^\top$. We can present (4) in a more compact form:

$$\frac{\partial \tilde{\mathbf{u}}}{\partial t} = \mathbf{H}^{(t)} \circ \hat{\boldsymbol{\mathcal{I}}}_{FDI}[\mathbf{u}^{(t)}], \tag{5}$$

where $\mathbf{H}^{(t)} \triangleq \mathbf{H}(t, \cdot)$ is a smooth function indexed by t with input dimension k and output dimension n, and the definition of the operator $\hat{\boldsymbol{\mathcal{I}}}_{FDI}$ is given in (2).

Consider a PDE system consisting of a G-symmetric PDE in the form (5) with an initial condition,

$$\begin{cases} \frac{\partial \tilde{\mathbf{u}}}{\partial t} = \mathbf{H}^{(t)} \circ \hat{\boldsymbol{\mathcal{I}}}_{FDI}[\mathbf{u}^{(t)}], \\ \tilde{\mathbf{u}}(0, \mathbf{x}) = \mathbf{u}_0(\mathbf{x}). \end{cases} \tag{6}$$

We approximate the temporal derivative by forward difference to discretize the PDE and formally solve the PDE system (6) by iteration. Let $0 = t_0 < t_1 < ... < t_N = T$ be a partition of the interval $[0, T]$, and the forward scheme is shown as follows:

$$\mathbf{u}^{(t_0)} = \mathbf{u}_0, \tag{7}$$

$$\mathbf{u}^{(t_{i+1})} = \mathbf{u}^{(t_i)} + \Delta t_i \cdot \mathbf{H}^{(t_i)} \circ \hat{\boldsymbol{\mathcal{I}}}_{FDI}[\mathbf{u}^{(t_i)}], \tag{8}$$

where $\Delta t_i \triangleq t_{i+1} - t_i$, $\mathbf{u}^{(t_i)} \triangleq \tilde{\mathbf{u}}(t_i, \cdot)$. As is well known, neural networks have universal approximation capabilities. Theoretically, if we choose $\mathbf{H}^{(t)}$ to be a neural network, we can represent any differential invariant. In practice, we introduce a series of parameterized multilayer perceptrons (MLPs), $\{\mathbf{h}_{\theta_i}, 0 \leq i \leq N - 1\}$, whose input dimension matches $\boldsymbol{\mathcal{I}}_{FDI}$ and output dimension matches $\mathbf{u}_0$. Consequently, we have the iterative process:

$$\mathbf{u}^{(t_{i+1})} = \mathbf{u}^{(t_i)} + \Delta t_i \cdot \mathbf{h}_{\theta_i} \circ \hat{\boldsymbol{\mathcal{I}}}_{FDI}[\mathbf{u}^{(t_i)}]. \tag{9}$$

We regard each iteration as an operator $\Psi_i : \mathbf{u}^{(t_i)} \mapsto \mathbf{u}^{(t_{i+1})}$ (see Figure 2), which is equivariant. Note that equivariance of Ψ_i does not rely on the existence and uniqueness of the solution to the original PDE system (6). Furthermore, if we replace $\boldsymbol{\mathcal{I}}_{FDI}$ with general invariants, the operator Ψ_i is still equivariant.

Utilizing transitivity of equivariance, we stack these equivariant operators together to get a feed-forward deep equivariant network, *i.e.* $\Psi \triangleq \Psi_{N-1} \circ \cdots \circ \Psi_1 \circ \Psi_0$. The number of layers corresponds to the number N of iterations, and the number of channels corresponds to the dimension of $\tilde{\mathbf{u}}$ in the PDE. The network takes $\mathbf{u}^{(t_0)} = \mathbf{u}_0$ as inputs, and produces $\mathbf{u}^{(t_N)}$ as the output features. Inference of the network aligns with the evolution process of the PDE. Furthermore, the network naturally incorporates the skip connection structure [23], which is renowned for its advantageous impact on network optimization. Inspired by PDEs based on invariants, we call the network *InvarPDE-Net*.

3.3. SupNorm normalized differential invariants

The basic version of InvarPDE-Net provides an approach to create equivariant networks without discretizing or sampling groups. However, for the affine group, its fundamental differential invariants are in the form of fractional polynomials, potentially leading to the division-by-zero issue in practice.

We notice that differential invariants of the affine group can be expressed by polynomial relative differential invariants. Building upon the observation, we propose a technique to construct a new type of affine invariants by normalizing polynomial relative differential invariants with a special norm, which not only avoid the division-by-zero issue but also exhibit better expressive power than classical differential invariants. To start with, we give a definition of polynomial relative differential invariants.

Definition 6 *Let G be the affine group acting on $X = \mathbb{R}^2$, $\mathcal{F} = \{\mathbf{u} | \mathbf{u} : X \to \mathbb{R}^n\}$ be the set of bounded smooth functions, $w : G \to \mathbb{R}^+$ be a positive multiplier, and P be a m-degree homogeneous polynomial. Define $\mathcal{J} : X \times \mathcal{F} \to \mathbb{R}$ as follows $\mathcal{J}(\mathbf{x}, \mathbf{u}) \triangleq P(\mathbf{u}(\mathbf{x}), \nabla\mathbf{u}(\mathbf{x}), ..., \nabla^d\mathbf{u}(\mathbf{x}))$. We call $\mathcal{J}$ a d-th order* ***(polynomial) relative differential invariant*** *of G with weight w and degree m, if $\forall \mathbf{u} \in \mathcal{F}, \mathbf{x} \in X, g \in G$, we have*

$$\mathcal{J}(g \cdot \mathbf{x}, g \cdot \mathbf{u}) = w(g)\mathcal{J}(\mathbf{x}, \mathbf{u}). \tag{10}$$

Low-order relative differential invariants of the affine group for scalar functions on $\mathbb{R}^2$ are shown in Table 1. Unless the weight $w \equiv 1$, a relative differential invariant is generally not an invariant. However, it is not hard to find that the result of dividing two relative differential invariants with the same weight is a differential invariant. But fractional polynomials may suffer from the division-by-zero issue and the drawback of causing substantial information loss.

Next, we present a technique to construct invariants based on relative differential invariants via normalization.

Relative Differential Invariants	Weight	Degree
u	1	1
$u_{xx}u_{yy} - u_{xy}^2$	$1/(\det \mathbf{A})^2$	2
$u_y^2 u_{xx} - 2u_x u_y u_{xy} + u_x^2 u_{yy}$	$1/(\det \mathbf{A})^2$	4

Table 1. We present relative differential invariants of the affine group for scalar functions up to order 2. Note that any element g in the affine group can be represented as $g = (\mathbf{A}, \mathbf{b})$.

Theorem 7 *Let G be the affine group acting on $X = \mathbb{R}^2$. Let $\mathcal{F} = \{\mathbf{u} | \mathbf{u} : X \to \mathbb{R}^n\}$ and $\mathcal{F}' = \{\mathbf{v} | \mathbf{v} : X \to \mathbb{R}^k\}$ be sets of bounded smooth functions on X. Define a norm on $\mathcal{F}'$ called* ***SupNorm****, $\|\mathbf{v}\|_{\sup} \triangleq \sup_{\mathbf{x}\in X} \|\mathbf{v}(\mathbf{x})\|_\infty$. Given a collection of relative differential invariants of G with weight w, denoted as $\mathcal{J}_i : X \times \mathcal{F} \to \mathbb{R} (i = 1, 2, ..., k)$. Define $\boldsymbol{\mathcal{J}} : X \times \mathcal{F} \to \mathbb{R}^k$ as $\boldsymbol{\mathcal{J}} \triangleq (\mathcal{J}_1, ..., \mathcal{J}_k)^\top$, and $\boldsymbol{\mathcal{J}}(\cdot, \mathbf{u})$ can be viewed as an element in $\mathcal{F}'$. Define $\boldsymbol{\mathcal{I}} : X \times \mathcal{F} \to \mathbb{R}^k$ as follows:*

$$\boldsymbol{\mathcal{I}}(\mathbf{x}, \mathbf{u}) \triangleq \frac{1}{\|\boldsymbol{\mathcal{J}}(\cdot, \mathbf{u})\|_{\sup}} \cdot \boldsymbol{\mathcal{J}}(\mathbf{x}, \mathbf{u}). \tag{11}$$

Then $\boldsymbol{\mathcal{I}}$ is a k-dimensional invariant of G.

The key to the proof of Theorem 7 is that for any $g \in G$, we have $\|\boldsymbol{\mathcal{J}}(\cdot, g \cdot \mathbf{u})\|_{\sup} = w(g)\|\boldsymbol{\mathcal{J}}(\cdot, \mathbf{u})\|_{\sup}$. A detailed proof is provided in Supplementary Material. We call the invariant constructed in (11) a **SupNorm normalized differential invariant** (SNDI). As the invariant involves global spatial information of derivatives, it is no longer a classical differential invariant. It may contain information beyond fundamental differential invariants.

To construct SNDIs, we start from a collection of polynomial relative differential invariants. Although the selection of relative differential invariants does not affect invariance, a recommended practice is to encompass those that sufficiently represent fundamental differential invariants, thus capturing adequate information. Next, we can normalize each relative differential invariant individually, as Theorem 7 also holds when $k = 1$. Alternatively, we can normalize all relative differential invariants with the same weight and the same degree together, which preserves more information between relative differential invariants. An additional benefit is that SNDIs derived in this way exhibit illumination invariance, *i.e.* $\boldsymbol{\mathcal{I}}(\mathbf{x}, c \cdot \mathbf{u}) = \boldsymbol{\mathcal{I}}(\mathbf{x}, \mathbf{u}), \forall c > 0$.

Here is an example to have a glimpse of the advantage in expressive power of SNDIs compared with that of classical differential invariants. Assuming u a bounded scalar function on $\mathbb{R}^2$, $\frac{u_{xx}u_{yy} - u_{xy}^2}{u_y^2 u_{xx} - 2u_x u_y u_{xy} + u_x^2 u_{yy}}$ is the only fundamental differential invariant of the affine group up to second order apart from the trivial one u itself. Through the newly proposed method of normalization, we can obtain two SNDIs $\frac{u_{xx}u_{yy} - u_{xy}^2}{\|u_{xx}u_{yy} - u_{xy}^2\|_{\sup}}$ and $\frac{u_y^2 u_{xx} - 2u_x u_y u_{xy} + u_x^2 u_{yy}}{\|u_y^2 u_{xx} - 2u_x u_y u_{xy} + u_x^2 u_{yy}\|_{\sup}}$. It is not hard to find that we can express the fundamental differential invariant as the quotient of two SNDIs up to a constant multiple, but not vice versa. From another perspective, we need to discretize functions by sampling on grid points in implementation. Given k polynomial relative differential invariants, each one can be viewed as an $M \times M$ matrix. Obtaining differential invariants through division would lead to the loss of at least M^2 degrees of freedom, while normalization only sacrifices at most k degrees of freedom.

In summary, SNDIs not only avoid the division-by-zero issue but also exhibit better expressive power than classical differential invariants. Given a collection of polynomial relative differential invariants, we construct SNDIs via normalization, and concatenate them together, resulting in a higher dimensional invariant $\boldsymbol{\mathcal{I}}_{SNDI}$. With theoretical guarantee of invariance, we can directly employ $\boldsymbol{\mathcal{I}}_{SNDI}$ to replace fundamental differential invariants $\boldsymbol{\mathcal{I}}_{FDI}$ in (9). Thus, each layer of InvarPDE-Net is adjusted to:

$$\mathbf{u}^{(t_{i+1})} = \mathbf{u}^{(t_i)} + \Delta t_i \cdot \mathbf{h}_{\theta_i} \circ \hat{\boldsymbol{\mathcal{I}}}_{SNDI}[\mathbf{u}^{(t_i)}]. \tag{12}$$

3.4. Extensions of network architecture

The equivariant network InvarPDE-net derived from a symmetric PDE requires the dimension of features, namely the number of channels, to be consistent across each layer. This is not the case for the majority of conventional networks. Hence, we generalize the network to accommodate varying channel numbers while maintaining equivariance.

Note that we can stack several PDEs of the same dimension sequentially, with the output of one PDE serving as the input of the subsequent one. Furthermore, when dealing with PDEs of different dimensions, we can linearly combine the output channels of one PDE to match the number of channels in the subsequent PDE. Since linear combinations of invariants remain invariants, the process does not affect equivariance. This extension allows us to create an equivariant network composed of multiple PDEs with varying channel numbers. We name the network *InvarPDEs-Net* (see Figure 1), including InvarPDE-Net as a special case.

In addition, we aim to design an equivariant layer that can be directly integrated into convolutional networks of various architectures by replacing convolutional layers. Such an equivariant layer will offer enhanced flexibility in its applications. A key aspect lies in the ability to freely specify input and output channel numbers, similar to a convolutional layer. By observing the iterative process in (12), we can adjust the output dimension of $\mathbf{h}_{\theta_i}$, and directly employ $\mathbf{h}_{\theta_i} \circ \hat{\boldsymbol{\mathcal{I}}}_{SNDI}[\mathbf{u}^{(t_i)}]$ as the output of this layer, which is still equivariant. Given input and output channel numbers, C_1 and C_2, we formulate the equivariant layer as follows:

$$\mathbf{u}_{out} = \mathbf{h}_\theta \circ \hat{\boldsymbol{\mathcal{I}}}_{SNDI}[\mathbf{u}_{in}], \tag{13}$$

where $\boldsymbol{\mathcal{I}}_{SNDI}$ is a k-dimensional SNDI, and $\mathbf{h}_\theta$ is an MLP with input dimension k and output dimension C_2. We name the equivariant layer *InvarLayer* (see Figure 3). It has a similar structure to the PDE iteration process (see Figure 2) but without the skip connection, allowing different input and output channel numbers.

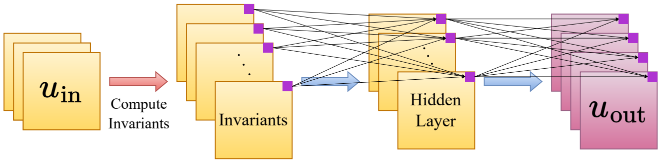

Figure 3. InvarLayer is an equivariant layer extracted and adapted from the iterative process of a symmetric PDE, which allows for free specification of input and output channel numbers.

3.5. Implementation

We establish a theoretical foundation in the continuous setting. When it comes to implementation, in the context of processing image data, discretization on 2D grids becomes necessary. We employ Gaussian derivatives to estimate derivatives by applying derivatives of a Gaussian kernel [25, 27]. For example, $f_x(x_0) \approx \sum_{n=1}^{N} \partial_x G(x_n; \sigma) f(x_n + x_0)$, where $G(x; \sigma)$ is a Gaussian kernel with standard deviation σ centered around 0, and x_n are grid points around 0. In the case of 2D grid points, it can be implemented using convolutions with specific kernels.

It is important to highlight that common network components are compatible with our approach. Proposition 3 guarantees that invariants under the operation of post-composition maintain their invariance property. This means BatchNorm [26], pointwise nonlinearities, 1×1 Convolution, and DropOut [54] can all be seamlessly integrated into our models without compromising equivariance. Pooling can also be incorporated into the models, though it introduces equivariance error to some extent. Specially, when using global pooling, we obtain invariant features.

In the following, we will discuss the input and ultimate output in InvarPDEs-Net, with a specific focus on image classification tasks. Currently, we simply replicate the image data along the channel dimension multiple times until the given number of channels is reached, which serves as the input of the network, *i.e.* $\mathbf{u}^{(t_0)} = \mathbf{u}_0$. For the final output of equivariant features of the network $\mathbf{u}^{(t_N)}$, we perform spatial global pooling to extract a set of invariant features, matching the number of channels. Subsequently, we apply two fully connected layers to acquire the ultimate classification result.

As for MLPs used for combining invariants in our networks, we apply two layer perceptrons in practice. Since MLPs operate on the vector $\boldsymbol{\mathcal{I}}(\mathbf{x}, \mathbf{u})$ for each point $\mathbf{x}$ and share weights spatially, they can be effectively implemented using 1×1 convolutions with the ReLU activation function. Likewise, connections between PDEs of different dimensions in InvarPDEs-Net can also be realized using 1×1

convolutions. As for the computation of SupNorm in constructing SNDIs, it can be easily implemented by applying global Max-Pooling over the channels corresponding to relative differential invariants that are normalized together.

3.6. Discussion

Unlike existing methods for designing equivariant networks, our framework does *not* apply discretization or sampling to the group. The number of channels is independent of the dimension of the group. When the group is larger, the number of fundamental differential invariants is bounded by the number of derivatives, and the same holds true for polynomial relative differential invariants. Therefore, the model size does not increase as the group becomes larger. That is why our framework can handle affine equivariance.

Moreover, our framework can be extended to continuous subgroups of the affine group. Common examples include the scale group, the shearing group, the rotation group, the rotation-scale group and the equi-affine group. To construct equivariant networks for these groups, we simply compute corresponding differential invariants and incorporate them into InvarPDE-Net. If the differential invariants involve fractions, the normalization technique is also applicable. The network structures, InvarPDEs-Net and InvarLayer, are compatible with these groups, and the implementation process remains the same. Therefore, it is a unified framework for the affine group and its continuous subgroups.

4. Experiments

For empirical validation, we implement InvarPDEs-Net and InvarLayer for three non-Euclidean groups: the scale group, the rotation-scale group, and the affine group. We conduct classification experiments on image datasets with different group transformations, and refrain from using data augmentation to emphasize the innate equivariance of networks.

4.1. Scale equivariance

Following previous works on scale equivariance [29, 40, 52, 78], we conduct experiments on datasets with scale variations, specifically Scale-MNIST and Scale-Fashion. We build Scale-MNIST and Scale-Fashion by rescaling the images of the MNIST [30] dataset and the Fashion-MNIST [71] dataset with the scaling factor randomly selected from $[0.3, 1]$. Then we reshape them back to the original size 28×28 by zero paddings. For both datasets, we use 10k samples for training and 50k for testing. In line with prior works [40, 52, 78], we integrate InvarLayer into a CNN with three convolution layers and two fully connected layers, and ensure that both InvarPDEs-Net and InvarLayer have fewer than 500k trainable parameters. For more details on the models and experiments, please refer to Supplementary Material.

Models	Scale-MNIST	Scale-Fashion
SiCNN [73]	97.53 ± 0.12	85.32 ± 0.22
SI-ConvNet [29]	97.56 ± 0.13	85.16 ± 0.14
SEVF [38]	97.28 ± 0.16	84.73 ± 0.11
DSS [69]	97.34 ± 0.13	84.50 ± 0.51
SS-CNN [19]	97.68 ± 0.15	85.39 ± 0.32
SESN [52]	97.92 ± 0.09	85.93 ± 0.28
ScDCFNet [78]	97.91 ± 0.08	86.19 ± 0.15
SE-CNN [40]	97.16	87.48
InvarPDEs-Net (Ours)	$\mathbf{98.30 \pm 0.06}$	$\mathbf{89.62 \pm 0.26}$
InvarLayer (Ours)	97.75 ± 0.05	89.50 ± 0.15

Table 2. Test accuracy (%) on Scale-MNIST and Scale-Fashion. All models have approximately 500k trainable parameters.

Experiments are repeated for six times using datasets generated with independent seeds. We report the mean ± std of the test accuracy of our models in Table 2. The results of SE-CNN on both datasets and SESN on Scale-MNIST come from the original papers [40, 52], and the others come from [78] under the same settings. On Scale-MNIST, InvarLayer achieves comparable results with other models and InvarPDEs-Net delivers the best performance. On Scale-Fashion, InvarPDEs-Net and InvarLayer outperform other models significantly.

4.2. Rotation-Scale equivariance

Models	RS-MNIST	RS-Fashion
SFCNN [63]	89.69 ± 0.40	75.80 ± 0.11
RDCF [2]	90.46 ± 0.33	73.96 ± 0.19
SEVF [38]	90.29 ± 0.37	71.03 ± 0.31
SESN [52]	90.19 ± 0.39	72.19 ± 0.05
ScDCFNet [78]	90.40 ± 0.09	72.24 ± 0.23
RST-CNN [18]	93.19 ± 0.29	78.64 ± 0.60
InvarPDEs-Net (Ours)	$\mathbf{95.80} \pm \mathbf{0.09}$	$\mathbf{79.48} \pm \mathbf{0.31}$
InvarLayer (Ours)	93.15 ± 0.21	74.51 ± 0.71

Table 3. Test accuracy (%) on RS-MNIST and RS-Fashion. All models have approximately 500k trainable parameters. RST-CNN is a rotation-scale equivariant network, while other compared models are only equivariant to rotation (SFCNN and RDCF) or scaling (SEVF, SESN, and ScDCFNet).

Gao et al. [18] first presented a rotation-scale equivariant network, RST-CNN. Following [18], we generate datasets RS-MNIST and RS-Fashion for evalutation. With the same procedure, we apply rotation (uniformly in $[0, 2\pi]$) and rescaling (uniformly in $[0.3, 1]$) to the images of MNIST and Fashion-MNIST, and zero-pad them back to the original size followed by upsizing images to 56×56. For both datasets, we use 5k samples for training and 50k for testing. Consistent with [18], we integrate InvarLayer into a CNN with three convolution layers and two fully connected layers, and keep the number of trainable parameters below 500k for both InvarPDEs-Net and InvarLayer.

The mean ± std of the test accuracy over six independent trials are reported in Table 3. Compared models include RST-CNN [18] and other models that are equivariant to either rotation (SFCNN [63] and RDCF [2]) or scaling (SEVF [38], SESN [52], and ScDCFNet [78]). The results of these models are obtained from [18] under the same settings. On RS-MNIST, InvarPDEs-Net significantly outperforms other models and InvarLayer exhibits comparable results with RST-CNN. On RS-Fashion, InvarPDEs-Net remains the top-performing model, while InvarLayer delivers relatively modest results. With minor adjustments of hyperparameters, InvarLayer lifts the accuracy to 93.40% on RS-MNIST and to 76.08% on RS-Fashion. More details about models and experiments are provided in Supplementary Material.

4.3. Affine equivariance

Models	Accuracy	Parameters
CapsNet [45]	79	8.1M
GE CapsNet [31]	89.10	235K
affine CapsNet [21]	93.21	–
RU CapsNet [8]	97.69	> 580K
affConv [37]	95.08 ± 0.31	373K
InvarPDEs-Net (Ours)	95.72 ± 0.12	340K
InvarLayer (Ours)	$\mathbf{98.45 \pm 0.15}$	365K

Table 4. Test accuracy (%) on affNIST after training on MNIST. The first four models are Capsule Networks that demonstrate robustness to affine transformations but admit few rigorous mathematical guarantees, while affConv is an affine equivariant network.

As for affine equivariance, the main counterpart we compare with is the affine equivariant model, affConv [37]. Following [37], we evaluate our models on the public dataset affNIST under the out-of-distribution setting. Specifically, we train our models on 50k non-transformed MNIST images (padded to 40×40) and test them on 320k affine-perturbed MNIST (affNIST) images with size 40×40. As mentioned before, it is impractical to apply affConv to deep networks, while InvarLayer overcomes the limitation. We use the structure of ResNet-32 for InvarLayer. For a fair comparison, we ensure that InvarPDEs-Net and InvarLayer both have fewer parameters than affConv (373k). Additional details can be found in Supplementary Material.

We present the mean ± std of test accuracy over six training runs with different random seeds in Table 4. Besides affConv, we also list the results under the same setup from some Capsule Networks [8, 21, 31, 45], which may lack rigorous theoretical guarantees of invariance. Although RU CapsNet performs better than affConv, which could not be well understood according to [37], our InvarLayer beats it by a margin of 0.76%. Moreover, our InvarPDE-net also outperforms affConv.

Additionally, we go back to the most common task, training on affNIST and testing on affNIST. The results are shown in Table 5. Compared with results of other models under the same setting, InvarLayer performs the best and achieves state-of-the-art results again.

Models	Accuracy	Parameters
ResNet-32 [23]	95.76	464K
ITN [76]	98.91	–
DE-CNNs [72]	99.08	> 2.5M
InvarPDEs-Net (Ours)	97.13 ± 0.05	373K
InvarLayer (Ours)	$\mathbf{99.25 \pm 0.08}$	365K

Table 5. Test accuracy (%) on affNIST after training on affNIST.

5. Conclusion

In this paper, we propose a new framework to achieve affine equivariance, a long-standing challenge in the field of equivariant networks. Within our framework, we construct a PDE-inspired equivariant network, InvarPDEs-Net, which showcases strong performance across extensive experiments. Furthermore, for more flexibility, we introduce an equivariant layer, InvarLayer, which can serve as a drop-in replacement for convolutional networks of various architectures. When combined with a ResNet structure, InvarLayer retains state-of-the-art results on the affNIST dataset. While the performance of InvarLayer exhibits some variability in certain setups, we recognize its immense potential. We believe that further refinement of the layer design based on our paradigm will elevate its capabilities to a higher level.

Our framework is quite promising and merits further extension. It is known that differential invariants exist for Lie groups satisfying certain regular conditions [41]. We concentrate on the affine group and make differential invariants applicable through the normalization technique, which is also suitable for its subgroups. How to adapt differential invariants of more general Lie groups into equivariant networks remains a future research. Additionally, besides the 2D planes considered in our work, it is worthwhile to study the extension of our framework to other manifolds, such as spheres and 3D spaces. Moreover, while our experiments involve image classification tasks, applications to a broader range of tasks in real world can be further explored.

References

[1] Erik J Bekkers, Maxime W Lafarge, Mitko Veta, Koen AJ Eppenhof, Josien PW Pluim, and Remco Duits. Roto-translation covariant convolutional networks for medical image analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2018. 3

[2] Xiuyuan Cheng, Qiang Qiu, Robert Calderbank, and Guillermo Sapiro. Rotdcf: Decomposition of convolutional filters for rotation-equivariant deep networks. *arXiv preprint arXiv:1805.06846*, 2018. 7, 8

[3] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning*, pages 2990–2999. PMLR, 2016. 1, 2

[4] Taco S Cohen and Max Welling. Steerable CNNs. In *International Conference on Learning Representations*, 2016. 1, 3

[5] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *International Conference on Learning Representations*, 2018. 3

[6] Taco S Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral CNN. In *International Conference on Machine Learning*, 2019. 3

[7] Pim De Haan, Maurice Weiler, Taco S Cohen, and Max Welling. Gauge equivariant mesh CNNs: Anisotropic convolutions on geometric graphs. In *International Conference on Learning Representations*, 2021. 3

[8] Fabio De Sousa Ribeiro, Georgios Leontidis, and Stefanos Kollias. Introducing routing uncertainty in capsule networks. *Advances in Neural Information Processing Systems*, 33: 6490–6502, 2020. 8

[9] Michaël Defferrard, Martino Milani, Frédérick Gusset, and Nathanaël Perraudin. Deepsphere: a graph-based spherical CNN. In *International Conference on Learning Representations*, 2019. 3

[10] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning SO(3) equivariant representations with spherical CNNs. In *Proceedings of the European Conference on Computer Vision*, 2018. 3

[11] Carlos Esteves, Avneesh Sud, Zhengyi Luo, Kostas Daniilidis, and Ameesh Makadia. Cross-domain 3D equivariant image embeddings. In *International Conference on Machine Learning*, 2019. 1, 3

[12] Carlos Esteves, Yinshuang Xu, Christine Allen-Blanchette, and Kostas Daniilidis. Equivariant multi-view networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1568–1577, 2019. 1, 3

[13] Carlos Esteves, Ameesh Makadia, and Kostas Daniilidis. Spin-weighted spherical CNNs. *Advances in Neural Information Processing Systems*, 33:8614–8625, 2020. 3

[14] Cong Fang, Zhenyu Zhao, Pan Zhou, and Zhouchen Lin. Feature learning via partial differential equation with applications to face recognition. *Pattern Recognition*, 69:14–25, 2017. 1, 3, 4

[15] Mark Fels and Peter J Olver. Moving coframes: II. regularization and theoretical foundations. *Acta Applicandae Mathematica*, 55:127–208, 1999. 3

[16] Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to Lie groups on arbitrary continuous data. In *International Conference on Machine Learning*, pages 3165–3176. PMLR, 2020. 1, 3

[17] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. SE(3)-transformers: 3D roto-translation equivariant attention networks. *Advances in Neural Information Processing Systems*, 33:1970–1981, 2020. 3

[18] L Gao, G Lin, and W Zhu. Deformation robust roto-scale-translation equivariant CNNs. *Transactions on Machine Learning Research*, 2022. 3, 7, 8

[19] Rohan Ghosh and Anupam K Gupta. Scale steerable filters for locally scale-invariant convolutional neural networks. *arXiv preprint arXiv:1906.03861*, 2019. 3, 7

[20] Simon Graham, David Epstein, and Nasir Rajpoot. Dense steerable filter CNNs for exploiting rotational symmetry in histology images. *IEEE Transactions on Medical Imaging*, 2020. 3

[21] Jindong Gu and Volker Tresp. Improving the robustness of capsule networks to image affine transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7285–7293, 2020. 8

[22] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003. 3

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 5, 8

[24] Lingshen He, Yuxuan Chen, Yiming Dong, Yisen Wang, Zhouchen Lin, et al. Efficient equivariant network. *Advances in Neural Information Processing Systems*, 34:5290–5302, 2021. 3

[25] Lingshen He, Yuxuan Chen, Zhengyang Shen, Yibo Yang, and Zhouchen Lin. Neural ePDOs: Spatially adaptive equivariant partial differential operator based networks. In *International Conference on Learning Representations*, 2022. 1, 3, 6

[26] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. pmlr, 2015. 6

[27] Erik Jenner and Maurice Weiler. Steerable partial differential operators for equivariant neural networks. In *International Conference on Learning Representations*, 2021. 1, 3, 6

[28] Chiyu Max Jiang, Jingwei Huang, Karthik Kashinath, Philip Marcus, Matthias Niessner, et al. Spherical CNNs on unstructured grids. In *International Conference on Learning Representations*, 2018. 3

[29] Angjoo Kanazawa, Abhishek Sharma, and David Jacobs. Locally scale-invariant convolutional neural networks. *arXiv preprint arXiv:1412.5104*, 2014. 7

[30] Yann LeCun. The MNIST database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998. 7

[31] Jan Eric Lenssen, Matthias Fey, and Pascal Libuschewski. Group equivariant capsule networks. *Advances in Neural Information Processing Systems*, 31, 2018. 3, 8

[32] Junying Li, Zichen Yang, Haifeng Liu, and Deng Cai. Deep rotation equivariant network. *Neurocomputing*, 2018. 3

[33] Risheng Liu, Zhouchen Lin, Wei Zhang, and Zhixun Su. Learning PDEs for image restoration via optimal control. In *Proceedings of the European Conference on Computer Vision*, pages 115–128. Springer, 2010. 1, 3, 4

[34] Risheng Liu, Zhouchen Lin, Wei Zhang, Kewei Tang, and Zhixun Su. Toward designing intelligent PDEs for computer vision: an optimal control approach. *Image and Vision Computing*, 31(1):43–56, 2013. 1, 3, 4

[35] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. PDE-net: Learning PDEs from data. In *International Conference on Machine Learning*, pages 3208–3216. PMLR, 2018. 3

[36] Zichao Long, Yiping Lu, and Bin Dong. PDE-net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network. *Journal of Computational Physics*, 399:108925, 2019. 3, 4

[37] Lachlan E MacDonald, Sameera Ramasinghe, and Simon Lucey. Enabling equivariance for arbitrary Lie groups. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8183–8192, 2022. 1, 2, 3, 8

[38] Diego Marcos, Benjamin Kellenberger, Sylvain Lobry, and Devis Tuia. Scale equivariance in CNNs with vector fields. *arXiv preprint arXiv:1807.11783*, 2018. 7, 8

[39] Joseph L Mundy and Andrew Zisserman. *Geometric invariance in computer vision*. MIT Press, 1992. 3

[40] Hanieh Naderi, Leili Goli, and Shohreh Kasaei. Scale equivariant CNNs with scale steerable filters. In *2020 International Conference on Machine Vision and Image Processing (MVIP)*, pages 1–5. IEEE, 2020. 3, 7

[41] Peter J Olver. *Applications of Lie groups to differential equations*. Springer Science & Business Media, 1993. 2, 3, 4, 8

[42] Peter J Olver. Moving frames. *Journal of Symbolic Computation*, 36(3-4):501–512, 2003. 3

[43] Peter J Olver. Modern developments in the theory and applications of moving frames. *London Math. Soc. Impact150 Stories*, 1:14–50, 2015. 3

[44] Peter J Olver, Guillermo Sapiro, and Allen Tannenbaum. Affine invariant detection: edge maps, anisotropic diffusion, and active contours. *Acta Applicandae Mathematica*, 59:45–77, 1999. 3

[45] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *Advances in Neural Information Processing Systems*, 30, 2017. 8

[46] Mateus Sangalli, Samy Blusseau, Santiago Velasco-Forero, and Jesús Angulo. Differential invariants for SE(2)-equivariant networks. In *2022 IEEE International Conference on Image Processing*, pages 2216–2220. IEEE, 2022. 1, 3

[47] Mateus Sangalli, Samy Blusseau, Santiago Velasco-Forero, and Jesus Angulo. Scale equivariant U-Net. In *33rd British Machine Vision Conference*, 2022. 3

[48] Mateus Sangalli, Samy Blusseau, Santiago Velasco-Forero, and Jesus Angulo. Moving frame net: SE(3)-equivariant network for volumes. In *NeurIPS Workshop on Symmetry and Geometry in Neural Representations*, pages 81–97. PMLR, 2023. 1, 3

[49] Zhengyang Shen, Lingshen He, Zhouchen Lin, and Jinwen Ma. PDO-eConvs: Partial differential operator based equivariant convolutions. In *International Conference on Machine Learning*, pages 8697–8706. PMLR, 2020. 1, 3

[50] Zhengyang Shen, Tiancheng Shen, Zhouchen Lin, and Jinwen Ma. PDO-eS2CNNs: Partial differential operator based equivariant spherical CNNs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9585–9593, 2021. 3

[51] Zhengyang Shen, Tao Hong, Qi She, Jinwen Ma, and Zhouchen Lin. PDO-s3DCNNs: Partial differential operator based steerable 3D CNNs. In *International Conference on Machine Learning*, pages 19827–19846. PMLR, 2022. 3

[52] Ivan Sosnovik, Michał Szmaja, and Arnold Smeulders. Scale-equivariant steerable networks. In *International Conference on Learning Representations*, 2019. 3, 7, 8

[53] Ivan Sosnovik, Artem Moskalev, and Arnold Smeulders. How to transform kernels for scale-convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1092–1097, 2021. 3

[54] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 6

[55] Zikai Sun and Thierry Blu. Empowering networks with scale and rotation equivariance using a similarity convolution. In *International Conference on Learning Representations*, 2022. 3

[56] Stanley L Tuznik, Peter J Olver, and Allen Tannenbaum. Affine differential invariants for invariant feature point detection. *arXiv preprint arXiv:1803.01669*, 2018. 3

[57] Dian Wang, Robin Walters, Xupeng Zhu, and Robert Platt. Equivariant q learning in spatial action spaces. In *Conference on Robot Learning*, pages 1713–1723. PMLR, 2022. 3

[58] Rui Wang, Robin Walters, and Rose Yu. Incorporating symmetry into deep dynamics models for improved generalization. In *International Conference on Learning Representations*, 2020. 3

[59] Yuanbin Wang, Bin Zhang, and Tianshun Yao. Projective invariants of co-moments of 2D images. *Pattern Recognition*, 43(10):3233–3242, 2010. 3

[60] Yuanbin Wang, Xingwei Wang, Bin Zhang, et al. Affine differential invariants of functions on the plane. *Journal of Applied Mathematics*, 2013, 2013. 4

[61] Maurice Weiler and Gabriele Cesa. General E(2)-equivariant steerable CNNs. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 3

[62] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco S Cohen. 3D steerable CNNs: Learning rotationally equivariant features in volumetric data. *Advances in Neural Information Processing Systems*, 31, 2018. 1, 3

[63] Maurice Weiler, Fred A Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant CNNs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 849–858, 2018. 1, 3, 7, 8

[64] Ruben Wiersma, Elmar Eisemann, and Klaus Hildebrandt. CNNs on surfaces using rotation-equivariant features. *ACM Transactions on Graphics*, 39(4):92–1, 2020. 3

[65] Ruben Wiersma, Ahmad Nasikun, Elmar Eisemann, and Klaus Hildebrandt. DeltaConv: anisotropic operators for geometric deep learning on point clouds. *ACM Transactions on Graphics*, 41(4):1–10, 2022. 3

[66] Thomas Wimmer, Vladimir Golkov, Hoai Nam Dang, Moritz Zaiss, Andreas Maier, and Daniel Cremers. Scale-equivariant deep learning for 3D data. *arXiv preprint arXiv:2304.05864*, 2023. 3

[67] Marysia Winkels and Taco S Cohen. Pulmonary nodule detection in CT scans with equivariant CNNs. *Medical Image Analysis*, 2019. 1, 3

[68] Daniel Worrall and Gabriel Brostow. Cubenet: Equivariance to 3D rotation and translation. In *Proceedings of the European Conference on Computer Vision*, pages 567–584, 2018. 1, 3

[69] Daniel Worrall and Max Welling. Deep scale-spaces: Equivariance over scale. *Advances in Neural Information Processing Systems*, 32, 2019. 3, 7

[70] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

[71] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 7

[72] Wenju Xu, Guanghui Wang, Alan Sullivan, and Ziming Zhang. Towards learning affine-invariant representations via data-efficient CNNs. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 904–913, 2020. 8

[73] Yichong Xu, Tianjun Xiao, Jiaxing Zhang, Kuiyuan Yang, and Zheng Zhang. Scale-invariant convolutional neural networks. *arXiv preprint arXiv:1411.6369*, 2014. 7

[74] Yilong Yang, Srinandan Dasmahapatra, and Sasan Mahmoodi. Rotation-scale equivariant steerable filters. In *Medical Imaging with Deep Learning*, 2023. 3

[75] Linfeng Zhao, Xupeng Zhu, Lingzhi Kong, Robin Walters, and Lawson LS Wong. Integrating symmetry into differentiable planning with steerable convolutions. In *International Conference on Learning Representations*, 2022. 3

[76] Yunhan Zhao, Ye Tian, Charless Fowlkes, Wei Shen, and Alan Yuille. Resisting large data variations via introspective transformation network. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3080–3089, 2020. 8

[77] Zhenyu Zhao, Zhouchen Lin, and Yi Wu. A fast alternating time-splitting approach for learning partial differential equations. *Neurocomputing*, 185:171–182, 2016. 1, 3, 4

[78] Wei Zhu, Qiang Qiu, Robert Calderbank, Guillermo Sapiro, and Xiuyuan Cheng. Scaling-translation-equivariant networks with decomposed convolutional filters. *The Journal of Machine Learning Research*, 23(1):2958–3002, 2022. 3, 7, 8