Projective Equivariant Network via Second-order Fundamental Differential Invariants

Anonymous Author(s)
Affiliation
Address
email

Abstract

Equivariant networks enhance model efficiency and generalization by embedding symmetry priors into their architectures. However, most existing methods, primarily based on group convolutions and steerable convolutions, face significant limitations when dealing with complex transformation groups, particularly the projective group, which plays a crucial role in vision. In this work, we tackle the challenge by constructing projective equivariant networks based on differential invariants. Using the moving frame method with a carefully selected cross section tailored for multi-dimensional functions, we derive a complete and concise set of second-order fundamental differential invariants of the projective group. We provide a rigorous analysis of the properties and transformation relationships of their underlying components, yielding a further simplified and unified set of fundamental differential invariants, which facilitates both theoretical analysis and practical applications. Building on this foundation, we develop the first deep projective equivariant networks, **PDINet**, which achieve full projective equivariance without discretizing or sampling the group. Empirical results on projectively transformed STL-10 and Imagenette datasets show that PDINet achieves improvements of 11.39% and 5.66% in accuracy over baseline results, respectively, demonstrating strong generalization to complex geometric transformations under out-of-distribution settings.

1 Introduction

Incorporating symmetry as an inductive bias into neural networks has emerged as a powerful approach to enhance model efficiency and generalization. Convolutional neural networks (CNNs) [Simonyan and Zisserman, 2014, He et al., 2016, Chen et al., 2017, Krizhevsky et al., 2012], which are among the most widely used architectures in deep learning, owe much of their success to their inherent translational equivariance. Building on this idea, Cohen and Welling [2016a] proposed Group Equivariant CNNs (G-CNNs), which generalize equivariance to broader transformations like rotations and reflections. Equivariant networks achieve symmetry incorporation by constructing network layers whose outputs transform in a predictable pattern under group actions applied to the inputs.

The development of equivariant networks began with G-CNNs, whose feature map can be seen as a function on a group. Although G-CNNs have proven effective in various tasks [Worrall and Brostow, 2018, Esteves et al., 2019, Lafarge et al., 2021, Shamsolmoali et al., 2021], they are less suited to continuous groups, as handling such groups typically requires group sampling or discretization, which introduces approximation errors and computational complexity. Then, Steerable CNNs [Cohen and Welling, 2016b, Weiler and Cesa, 2019] were proposed to overcome these limitations by viewing features as fields that transform according to specified group representations. In this framework, G-CNNs can be interpreted as a special case where the group representation is chosen to be the regular representation. Steerable CNNs are capable of handling continuous groups such as $\mathrm{SO}(2)$ and $\mathrm{SO}(3)$ directly, thereby significantly broadening the scope of equivariant networks to include real-world

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

symmetries beyond discrete groups [Weiler et al., 2018, Wang and Walters, 2022, Wang et al., 2020]. However, for more complex non-compact Lie groups such as the projective group, deriving closed-form steerable basis filters becomes intractable, inherently limiting the applicability of steerable CNNs. To address this, MacDonald et al. [2022] enabled group convolutions over finite-dimensional Lie groups by computing the integral on the Lie algebra, thus introducing a projective equivariant model, homConv. However, this method still relies on group sampling, which results in exponential memory growth with increasing network depth, thereby hindering scalability to deeper architectures. Mironenco and Forré [2024] improved sampling efficiency via group decompositions, but focused solely on affine subgroups like $\mathrm{GL}^+(n, \mathbb{R})$ and $\mathrm{SL}(n, \mathbb{R})$, without addressing more complex groups such as the projective group. Recently, Li et al. [2024, 2025] proposed InvarLayer and Steerable EquivarLayer that construct affine equivariant networks based on invariants, enabling closed-form and sampling-free affine equivariance. However, these works remain specialized to the affine group and do not yet generalize to more complex non-compact groups like the projective group.

Actually, projective transformations play a fundamental role in computer vision [Hartley and Zisserman, 2003, Birchfield, 1998, Mohr and Triggs, 1996], as they capture the relationships between objects and their images under perspective projections. Achieving equivariance on projective transformations is especially critical in practical applications such as mobile robot navigation, 3D scene analysis, and camera pose estimation, where accurately handling perspective effects and viewpoint changes can significantly enhance model robustness and accuracy. Early on, Suk and Flusser [2004] proposed a projective invariant feature extraction method based on the projective moment invariants. Nevertheless, the invariants are formulated as infinite series of moment products, leading to significant computational overhead and intractable error analysis in practical implementations. To overcome this limitation, Li et al. [2018] proposed an alternative framework that constructs projective invariants using finite combinations of weighted moments, where the weights are derived from relative projective differential invariants. Note that moment-based projective invariants are essentially global image descriptors, which makes them inappropriate for constructing equivariant operators that act locally on feature fields for capturing fine-grained spatial patterns. Instead, differential invariants inherently hold the property of acting locally at each spatial position, which makes them a natural foundation for building equivariant operators [Sangalli et al., 2022, 2023, Li et al., 2024, 2025]. While Olver [2023] has proposed a systematic framework for the computation of projective differential invariants via the moving frames method, it requires at least third-order derivatives because the projective action is not free at second order for scalar functions. Besides, they only consider single-channel cases, and the extension to multi-dimensional cases needs more complex expressions involving high-order derivatives, which limits their practicality in computation and applications on color images.

In this work, we construct projective equivariant networks based on differential invariants, achieving full projective equivariance without relying on group discretization or sampling. This overcomes the depth limitations of homConv [MacDonald et al., 2022] and enables effective scaling to deeper architectures. A core challenge lies in deriving concise and practical projective differential invariants. To support color images and multi-channel intermediate features in modern neural networks, we focus on the differential invariants for multi-dimensional functions. In this case, the projective group acts freely on the second-order jet space, allowing us to derive a complete set of second-order fundamental differential invariants using the moving frame method [Olver, 2015], which can express any second-order invariant of the projective group. However, the choice of cross section used to define the moving frame significantly affects the form of the resulting invariants. While a direct extension of the cross section in [Olver, 2023] to the multi-dimensional case is theoretically valid, it leads to prohibitively long expressions with hundreds of terms, rendering them impractical. Instead, we propose a new cross section tailored to the multi-dimensional structure, which involves up to second-order derivatives, yielding a much more concise set of fundamental differential invariants. Further analysis reveals that these fundamental invariants are composed of a set of simpler components. By exploring the algebraic properties and transformation relationships of these components, we further simplify invariants into a unified set of fundamental invariants, facilitating both practical use and theoretical analysis. Based upon these simplified invariants, we design learnable equivariant operators by combining them with parameterized multi-layer perceptrons (MLPs), and embed the operators into standard neural network backbones to build **PDINet**, the first deep projective equivariant network free from group sampling. Empirical evaluations under challenging out-of-distribution settings demonstrate the strong generalization ability of our model to complex geometric transformations.

We summarize our main contributions as follows:

- We employ the moving frame method to derive a complete set of second-order fundamental differential invariants of the projective group for multi-dimensional functions, enabling support for color images and multi-channel features.
- We conduct an in-depth analysis of the algebraic structure and transformation properties of these invariants, resulting in a further simplified and unified set of fundamental invariants that facilitate both theoretical understanding and practical computation.
- We develop **PDINet** based on second-order projective differential invariants. It is the first deep projective equivariant network that achieves full projective equivariance without relying on group discretization or sampling, thus allowing effective scaling to deeper architectures.
- Numerical experiments on projectively deformed STL-10 and Imagenette[1]. under out-of-distribution settings demonstrate the effectiveness of our model, with improvements of 11.39% and 5.66% over baseline results, showcasing its strong generalization capability under complex geometric transformations.[2]

2 Method

2.1 Basic concepts and notations

To begin with, we introduce some basic concepts and notations necessary for our formulation. An image can be viewed as a continuous function $\mathbf{u}(x, y)$ defined on a 2D plane. For example, an RGB image corresponds to a three-dimensional function. Likewise, intermediate features in neural networks can also be interpreted as functions, and each layer can be seen as an operator that maps one function to another.

A central concept in this work is equivariance. If the output of an operator undergoes a corresponding transformation when the input is transformed, it is referred to as equivariance. The formal definition is as follows:

Definition 1 *An operator $\psi : \mathcal{F}_1 \to \mathcal{F}_2$ is said to be* ***equivariant*** *with respect to a group G if*

$$g \cdot \psi(\mathbf{u}) = \psi(g \cdot \mathbf{u}), \quad \forall g \in G, \mathbf{u} \in \mathcal{F}_1, \tag{1}$$

where $\mathcal{F}_1$ and $\mathcal{F}_2$ are the input and output function spaces, respectively.

Let X denote the domain, $U = \mathbb{R}^n$ be the range of a function, and $U^{(d)} = U \times U_1 \times \cdots \times U_d$ be the derivative space up to order d. A group action $g \cdot \mathbf{x}$ on the domain naturally induces an action on functions, defined as $(g \cdot \mathbf{u})(\mathbf{x}) = \mathbf{u}(g^{-1} \cdot \mathbf{x})$, which models how geometric transformations deform images. This action further extends to derivatives through prolongation to the jet space $X \times U^{(d)}$. For example, the first-order prolongation of the action on $X \times U^{(1)}$ can be expressed as: $(\mathbf{x}, \mathbf{u}(\mathbf{x}), \nabla\mathbf{u}(\mathbf{x})) \mapsto \Big(g \cdot \mathbf{x}, (g \cdot \mathbf{u})|_{g\cdot\mathbf{x}}, \nabla(g \cdot \mathbf{u})\big|_{g\cdot\mathbf{x}}\Big)$, where $(g \cdot \mathbf{u})|_{g\cdot\mathbf{x}} = \mathbf{u}(\mathbf{x})$ holds by definition.

A differential invariant is a quantity that remains unchanged under the prolonged group action. The definition is given below.

Definition 2 *Given a group G acting on X, a d-th order* ***differential invariant*** *is a function $\mathcal{I} : X \times U^{(d)} \to \mathbb{R}$ such that*

$$\mathcal{I}(g \cdot (\mathbf{x}, \mathbf{u}^{(d)})) = \mathcal{I}(\mathbf{x}, \mathbf{u}^{(d)}), \quad \forall g \in G, (\mathbf{x}, \mathbf{u}^{(d)}) \in X \times U^{(d)}, \tag{2}$$

where $g \cdot (\mathbf{x}, \mathbf{u}^{(d)})$ denotes the prolonged group action on the jet space $X \times U^{(d)}$.

The definition can be extended to the multi-dimensional case. Specifically, we call $\boldsymbol{\mathcal{I}} = (\mathcal{I}_1, \ldots, \mathcal{I}_k)$ an **k-dimensional differential invariant**. In addition, we define **relative differential invariants**, which may transform with a weight function under the group action:

$$\mathcal{R}(g \cdot (\mathbf{x}, \mathbf{u}^{(d)})) = w(g, \mathbf{x}) \cdot \mathcal{R}(\mathbf{x}, \mathbf{u}^{(d)}), \tag{3}$$

[1]Imagenette is a publicly available dataset downloaded from https://github.com/fastai/imagenette.

[2]Our code will be released upon acceptance.

where $w(g, \mathbf{x})$ is a scalar weight depending on the group element g and the point $\mathbf{x}$. Notably, differential invariants are closely tied to equivariance, as a (multi-dimensional) differential invariant $\boldsymbol{\mathcal{I}}$ yields an equivariant operator $\hat{\boldsymbol{\mathcal{I}}}(\mathbf{u})(\mathbf{x}) \triangleq \boldsymbol{\mathcal{I}}(\mathbf{x}, \mathbf{u}^{(d)})$ satisfying $\hat{\boldsymbol{\mathcal{I}}}(g \cdot \mathbf{u}) = g \cdot \hat{\boldsymbol{\mathcal{I}}}(\mathbf{u})$.

In this work, we focus on constructing such differential invariants and using them to build projective equivariant operators for neural networks.

2.2 Method of moving frames

The method of moving frames is a powerful technique for deriving differential invariants [Olver, 2003, 2015]. We begin with the definition of a moving frame.

Definition 3 *[Olver, 2015] Let G be a Lie group acting on a manifold $\mathcal{M}$. A* ***moving frame*** *is a map $\eta : \mathcal{M} \to G$ such that*

$$\eta(g \cdot z) = \eta(z) \cdot g^{-1}, \quad g \in G, z \in \mathcal{M}. \tag{4}$$

Given a moving frame, the **invariantization** of a function $F : \mathcal{M} \to \mathbb{R}$ is defined as

$$\iota(F)(z) \triangleq F(\eta(z) \cdot z), \tag{5}$$

which converts an arbitrary function F into a group-invariant function satisfying $\iota(F)(g \cdot z) = \iota(F)(z)$. More generally, we can define an invariant as $\mathcal{I}(g \cdot z) = \mathcal{I}(z), g \in G, z \in \mathcal{M}$. In our context, the manifold of interest is the jet space $\mathcal{M} = X \times U^{(n)}$ and we focus on differential invariants.

A necessary and sufficient condition for the existence of a moving frame is that the group G acts freely and regularly on the manifold $\mathcal{M}$. Under this condition, a moving frame can be constructed via a cross section, as described below:

Theorem 4 *[Olver, 2015] Let G be a r-dimensional Lie group acting freely and regularly on a m-dimensional manifold $\mathcal{M}$. Given local coordinates $z = (z_1, \ldots, z_m)$ on $\mathcal{M}$, let $\mathcal{K}$ be a cross section of the form $\mathcal{K} = \{z_1 = c_1, z_2 = c_2, \ldots, z_r = c_r\} \subset \mathcal{M}$, where c_i are constants. Then for $z \in \mathcal{M}$, there exists a unique $g \in G$ such that $g \cdot z \in K$. Defining $\eta(z) = g$, namely $\eta(z) \cdot z \in \mathcal{K}$, yields a map $\eta : \mathcal{M} \to G$, which is a moving frame.*

Here, the group action is said to be **free** if for any $g \in G$, $g \cdot z = z$ implies $g = e$, where e is the identity element of the group. Usually, the group action can be made free by increasing the order of the jet space. The action is **regular** if the orbits form a regular foliation, which is typically satisfied in common groups.

With a moving frame obtained from Theorem 4, we can construct a complete set of fundamental invariants, meaning any invariant can be expressed as a combination of these fundamental invariants.

Theorem 5 *[Olver, 2015] Let $\eta : \mathcal{M} \to G$ be a moving frame from Theorem 4 and define $w(g, z) \triangleq g \cdot z$. Then*

$$w(\eta(z), z) = (c_1, c_2, \ldots, c_r, w_{r+1}(\eta(z), z), \ldots, w_m(\eta(z), z)), \tag{6}$$

where $I_1(z) \triangleq w_{r+1}(\eta(z), z)$, ..., $I_{m-r}(z) \triangleq w_m(\eta(z), z)$ constitute a complete system of functionally independent invariants, called ***fundamental invariants****.*

This theorem provides a method to construct fundamental invariants via the moving frame and indicates that the number of fundamental invariants is $m - r$. In the following sections, we will leverage these results to derive projective differential invariants.

2.3 Projective transformation

Projective transformations are ubiquitous in the visual world as two different views of the same planar object can be related by a 2D projective transformation. A standard projective group action is described by the projective special linear group $\mathrm{PSL}(3, \mathbb{R})$ acting on the 2D projective plane $\mathbb{RP}^2$, which can be interpreted as the set of equivalence classes of points $(x, y, p) \sim (cx, cy, cp)$ for any $c \neq 0$. Points in $\mathbb{RP}^2$ with $p \neq 0$ can be represented in inhomogeneous coordinates as

(x, y), corresponding to the homogeneous coordinate $(x, y, 1)$. Thus, the action of a projective transformation on 2D coordinates can be written as

$$x' = \frac{\alpha x + \beta y + \gamma}{\rho x + \sigma y + \tau}, y' = \frac{\lambda x + \mu y + \nu}{\rho x + \sigma y + \tau}, \tag{7}$$

where the transformation is parameterized by the coefficient matrix

$$P = \begin{pmatrix} \alpha & \beta & \gamma \\ \lambda & \mu & \nu \\ \rho & \sigma & \tau \end{pmatrix}. \tag{8}$$

Since the transformation is defined up to a nonzero scaling factor, we can normalize by requiring the determinant of P to be 1, i.e., $\Delta = \det(P) = 1$. Thus, there are 8 independent degrees of freedom. The transformation reduces to an affine transformation when $\rho = \sigma = 0$, while a pure projective transformation, characterized by $\rho^2 + \sigma^2 \neq 0$, exhibits nonlinear behavior. Thus, projective transformations represent a more general and complex class of geometric transformations.

For an n-dimensional function $\mathbf{u}(x, y)$, the projective transformation of coordinates induces a natural action on the function, $\mathbf{u}'(x', y') = \mathbf{u}(x, y)$, which can be further prolonged to its derivatives. We denote the derivatives of the i-th component function $u^{[i]}$ as

$$u^{[i]}_{jk} \triangleq D_x^j D_y^k u^{[i]}, \tag{9}$$

where D_x and D_y are the differentiation operators with respect to x and y, respectively. Under a projective transformation, these derivatives transform as

$$u^{[i]}_{jk} \mapsto u^{[i]\prime}_{jk} = D_{x'}^j D_{y'}^k u^{[i]\prime}, \tag{10}$$

with the transformed differential operators given by

$$D_{x'} = \frac{\rho x + \sigma y + \tau}{\Delta} \left(((\mu\rho - \lambda\sigma)x + \mu\tau - \nu\sigma)D_x + ((\mu\rho - \lambda\sigma)y - \lambda\tau + \nu\rho)D_y \right), \tag{11}$$

$$D_{y'} = \frac{\rho x + \sigma y + \tau}{\Delta} \left(((\alpha\sigma - \beta\rho)x - \beta\tau + \gamma\sigma)D_x + ((\alpha\sigma - \beta\rho)y + \alpha\tau - \gamma\rho)D_y \right). \tag{12}$$

2.4 Projective differential invariants of multi-dimensional functions

The projective group action is not free on the second-order jet space for scalar functions, requiring prolongation to the third-order jet space to achieve freeness. This leads to complex formulations [Olver, 2023], which may limit the practicality of the resulting invariants due to their complexity and computational cost. Moreover, in practice, third-order derivatives are harder to estimate reliably from data than lower-order ones. In this work, we focus on multi-dimensional functions, which naturally align with applications such as color image processing. In this setting, the group action is free on the second-order jet space, allowing the existence of second-order differential invariants. Using the method of moving frames, we can derive these invariants, where the choice of cross section significantly influences the simplicity of the resulting expressions. Although the cross section proposed by Olver [2023] can be extended to the multi-dimensional case, the resulting invariants tend to be lengthy, typically involving hundreds of terms, which makes them less practical. Instead, we propose an alternative cross section that leverages multiple dimensions while relying only on derivatives up to second order, yielding invariants with significantly more concise and tractable forms.

Specifically, we choose the following cross section:

$$\mathcal{K} = \{x = y = 0, u^{[1]}_x = 1, u^{[1]}_y = 0, u^{[2]}_x = 0, u^{[2]}_y = 1, u^{[1]}_{xx} = u^{[1]}_{xy} = 0\}. \tag{13}$$

This defines 8 normalization equations, which, together with the constraint $\Delta = 1$, determine all group parameters, thereby establishing the moving frame η. The detailed derivation of the moving frame is provided in the Appendix.

With the moving frame η constructed, we can then apply the invariantization process according to Theorem 5 to obtain a complete set of fundamental differential invariants. For the coordinates involved in the cross section, we have

$$\iota(x) = 0, \iota(y) = 0, \iota(u^{[1]}_x) = 1, \iota(u^{[1]}_y) = 0,$$
$$\iota(u^{[2]}_x) = 0, \iota(u^{[2]}_y) = 1, \iota(u^{[1]}_{xx}) = 0, \iota(u^{[1]}_{xy}) = 0.$$

The remaining coordinates of the second-order jet space yield the following differential invariants:

$$\iota(u_{yy}^{[1]}) = \frac{\mathcal{T}_{111}}{\mathcal{J}_{12}^2}, \tag{14}$$

$$\iota(u_{xx}^{[2]}) = \frac{\mathcal{T}_{222}}{\mathcal{J}_{12}^2}, \tag{15}$$

$$\iota(u_{xy}^{[2]}) = -\frac{\mathcal{T}_{212} + 2\mathcal{T}_{122}}{2\mathcal{J}_{12}^2}, \tag{16}$$

$$\iota(u_{yy}^{[2]}) = \frac{\mathcal{T}_{121} + 2\mathcal{T}_{112}}{\mathcal{J}_{12}^2}, \tag{17}$$

$$\iota(u_{x}^{[i]}) = -\frac{\mathcal{J}_{2i}}{\mathcal{J}_{12}}, \quad 3 \leq i \leq n, \tag{18}$$

$$\iota(u_{y}^{[i]}) = \frac{\mathcal{J}_{1i}}{\mathcal{J}_{12}}, \quad 3 \leq i \leq n, \tag{19}$$

$$\iota(u_{xx}^{[i]}) = \frac{\mathcal{J}_{12}\mathcal{T}_{2i2} + \mathcal{J}_{2i}\mathcal{T}_{212}}{\mathcal{J}_{12}^3}, \quad 3 \leq i \leq n, \tag{20}$$

$$\iota(u_{xy}^{[i]}) = -\frac{2\mathcal{J}_{12}\mathcal{T}_{1i2} + 2\mathcal{J}_{12}\mathcal{T}_{21i} + 3\mathcal{J}_{2i}\mathcal{T}_{112}}{\mathcal{J}_{12}^3}, \quad 3 \leq i \leq n, \tag{21}$$

$$\iota(u_{yy}^{[i]}) = \frac{\mathcal{J}_{12}\mathcal{T}_{1i1} + 2\mathcal{J}_{1i}\mathcal{T}_{112}}{\mathcal{J}_{12}^3}, \quad 3 \leq i \leq n, \tag{22}$$

where $\mathcal{J}_{ij}$ and $\mathcal{T}_{ijk}$ are two key quantities defined as:

$$\mathcal{J}_{ij} \triangleq u_x^{[i]}u_y^{[j]} - u_x^{[j]}u_y^{[i]}, \tag{23}$$

$$\mathcal{T}_{ijk} \triangleq u_{xx}^{[i]}u_y^{[j]}u_y^{[k]} + u_{yy}^{[i]}u_x^{[j]}u_x^{[k]} - u_{xy}^{[i]}(u_x^{[j]}u_y^{[k]} + u_x^{[k]}u_y^{[j]}), \tag{24}$$

satisfying $\mathcal{J}_{ii} = 0$, $\mathcal{J}_{ij} = -\mathcal{J}_{ji}$, and $\mathcal{T}_{ijk} = \mathcal{T}_{kji}$.

The invariants (14)-(22), together with the obvious invariants $S_0 = \{u^{[i]} \mid 1 \leq i \leq n\}$, form a complete set of second-order fundamental differential invariants of the projective group. Compared to projective invariants for scalar functions [Olver, 2023], our results involve up to second-order derivatives and are expressed in a more concise form.

2.5 Fundamental components of projective differential invariants

In the previous subsection, we have derived a complete set of second-order fundamental differential invariants. While relatively concise, their expressions are asymmetric and depend on the specific choice of the first two dimensions used in the cross section. To obtain a simpler, more unified, and elegant formulation, we conduct a deeper analysis of the fundamental components of these invariants. This enables us to further simplify their structure while preserving completeness.

Note that the numerators and denominators in (14)-(22) are all relative invariants. Thus, we focus on the properties of these relative invariants, as absolute invariants can be obtained by taking the ratio of two relative invariants with the same weight. Moreover, since the expressions are built from the basic quantities $\mathcal{J}_{ij}$ and $\mathcal{T}_{ijk}$, we will delve into their transformation properties and algebraic relationships.

We first present three classes of simplified relative differential invariants of the projective group.

Theorem 6 *Let $W = \frac{(\rho x + \sigma y + \tau)^3}{\Delta}$. Then the following quantities are relative differential invariants of the projective group:*

- *For $i \neq j$, $\mathcal{J}_{ij}$ is a relative differential invariant of weight W.*
- *For $1 \leq i \leq n$, $\mathcal{T}_{iii}$ is a relative differential invariant of weight W^2.*
- *For $1 \leq i, j \leq n$, $\mathcal{T}_{iji} + 2\mathcal{T}_{iij}$ is a relative differential invariant of weight W^2.*

These relative invariants are not functionally independent; rather, they can be transformed into one another. Given that there are $6n - 6$ second-order fundamental differential invariants according to

Subsection 2.4, we expect a complete and independent set of relative invariants to contain $6n-5$ elements. To this end, we investigate the transformation rules among the relative invariants and aim to identify a minimal generating set sufficient to express all fundamental differential invariants. We start with the transformation properties of $\mathcal{J}_{ij}$.

Theorem 7 *For any indices $1 \leq i_1, i_2, i_3, i_4 \leq n$, the following equation holds:*

$$\mathcal{J}_{i_1 i_2} \cdot \mathcal{J}_{i_3 i_4} + \mathcal{J}_{i_1 i_3} \cdot \mathcal{J}_{i_4 i_2} + \mathcal{J}_{i_1 i_4} \cdot \mathcal{J}_{i_2 i_3} = 0. \tag{25}$$

This implies that for any four distinct indices i_1, i_2, i_3, i_4, the six pairwise combinations of $\mathcal{J}_{ij}$ are dependent such that once any five are known, the remaining one can be determined. Based on Theorem 7, we can construct a subset of $\{\mathcal{J}_{ij} \mid i \neq j\}$ that is sufficient to express all $\mathcal{J}_{ij}$.

Theorem 8 *Let $S_1 \triangleq \{\mathcal{J}_{12}, \mathcal{J}_{23}, \ldots, \mathcal{J}_{n-1,n}\}$ and $S_2 \triangleq \{\mathcal{J}_{13}, \mathcal{J}_{24}, \ldots, \mathcal{J}_{n-2,n}\}$. Then $S_1 \cup S_2$ is a generating set for the collection $\{\mathcal{J}_{ij} \mid i \neq j\}$, meaning that any $\mathcal{J}_{ij}$ can be expressed as a functional combination of these elements.*

According to Theorem 7, $\mathcal{J}_{i,i+3}$ can be written in terms of $\mathcal{J}_{i,i+1}, \mathcal{J}_{i+1,i+2}, \mathcal{J}_{i+2,i+3}, \mathcal{J}_{i,i+2}$, and $\mathcal{J}_{i+1,i+3}$, all of which belong to $S_1 \cup S_2$. By induction, any $\mathcal{J}_{ij}$ can thus be recovered from the generating set. A complete and rigorous proof is provided in the Appendix.

Before establishing the transformation relationships for $\mathcal{T}_{ijk}$, we provide a more compact representation of $\mathcal{J}_{ij}$ and $\mathcal{T}_{ijk}$ to clarify their structural relationships:

$$\mathcal{J}_{ij} = \mathbf{g}_i^\top \mathbf{Q} \mathbf{g}_j, \tag{26}$$

$$\mathcal{T}_{ijk} = \mathbf{g}_i^\top \mathbf{Q}^\top \mathbf{H}_j \mathbf{Q} \mathbf{g}_k, \tag{27}$$

where $\mathbf{g}_i \triangleq \left(u_x^{[i]}, u_y^{[i]}\right)^\top$ is the gradient of the i-th component function, $\mathbf{H}_j$ is the Hessian matrix of the j-th component function, and $\mathbf{Q}$ is a fixed orthogonal matrix defined as:

$$\mathbf{Q} \triangleq \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

Notably, $\{\mathcal{J}_{ij}\}$ can be used to establish the transformation relationships between the gradients.

Lemma 9 *For any distinct indices i, j, k, the gradient $\mathbf{g}_k$ can be expressed in terms of $\mathbf{g}_i$ and $\mathbf{g}_j$ as:*

$$\mathbf{g}_k = \frac{\mathcal{J}_{kj}}{\mathcal{J}_{ij}} \mathbf{g}_i + \frac{\mathcal{J}_{ki}}{\mathcal{J}_{ji}} \mathbf{g}_j. \tag{28}$$

Using Lemma 9 and the form of $\mathcal{T}_{ijk}$ in (27), we derive the transformation rule for $\mathcal{T}_{ijk}$ given $\{\mathcal{J}_{ij}\}$.

Theorem 10 *Let $i \neq k$, and let i', j, k' be arbitrary indices. Then $\mathcal{T}_{i'jk'}$ can be expressed as:*

$$\mathcal{T}_{i'jk'} = \frac{\mathcal{J}_{i'k}\mathcal{J}_{k'k}}{\mathcal{J}_{ik}^2} \mathcal{T}_{iji} - \frac{\mathcal{J}_{i'k}\mathcal{J}_{k'i} + \mathcal{J}_{i'i}\mathcal{J}_{k'k}}{\mathcal{J}_{ik}^2} \mathcal{T}_{ijk} + \frac{\mathcal{J}_{i'i}\mathcal{J}_{k'i}}{\mathcal{J}_{ik}^2} \mathcal{T}_{kjk}. \tag{29}$$

This result implies that, for a fixed j, the triplet $\{\mathcal{T}_{iji}, \mathcal{T}_{ijk}, \mathcal{T}_{kjk}\}$ serves as a generating set for $\{\mathcal{T}_{i'jk'} \mid 1 \leq i', k' \leq n\}$, provided $\{\mathcal{J}_{ij}\}$ is known.

With the transformation relationships for $\mathcal{J}_{ij}$ and $\mathcal{T}_{ijk}$ established, we can now construct a minimal set of relative invariants that suffices to express all the relative invariants in Theorem 6.

Theorem 11 *Define the following sets of relative invariants:*

$$S_3 \triangleq \{\mathcal{T}_{111}, \mathcal{T}_{222}, \ldots, \mathcal{T}_{nnn}\}, \tag{30}$$

$$S_4 \triangleq \{\mathcal{T}_{121} + 2\mathcal{T}_{112}, \mathcal{T}_{131} + 2\mathcal{T}_{113}, \ldots, \mathcal{T}_{n-1,n,n-1} + 2\mathcal{T}_{n-1,n-1,n}\}, \tag{31}$$

$$S_5 \triangleq \{\mathcal{T}_{212} + 2\mathcal{T}_{221}, \mathcal{T}_{323} + 2\mathcal{T}_{332}, \ldots, \mathcal{T}_{n,n-1,n} + 2\mathcal{T}_{n,n,n-1}\}. \tag{32}$$

Then $S_1 \cup S_2 \cup S_3 \cup S_4 \cup S_5$ can express all the relative invariants in Theorem 6.

It can be shown (see the Appendix) that the fundamental differential invariants (14)-(22) can be expressed via the relative invariants in Theorem 6. Therefore, we arrive at the following conclusion:

Theorem 12 *The union $S \triangleq S_0 \cup S_1 \cup S_2 \cup S_3 \cup S_4 \cup S_5$ forms a complete set of relative differential invariants to express all second-order differential invariants of the projective group.*

This set contains exactly $6n - 5$ elements, matching the expected minimal number needed to express all second-order fundamental differential invariants. Compared to the invariants derived in Subsection 2.4, the current formulation is further simplified and independent of the specific choice of the first two dimensions in the cross section, exhibiting a more unified structure. Moreover, while the original invariants involve polynomials of degree up to five, the present set only contains at most cubic expressions, resulting in lower computational complexity.

2.6 Projective equivariant network

As discussed before, with a set of relative invariants obtained, we can convert them into absolute invariants by dividing each by another relative invariant with the same weight. We apply this procedure to relative invariants in S to construct a complete set of fundamental differential invariants.

Specifically, we select $\mathcal{R}_0 = \frac{1}{n}(\mathcal{J}_{12} + \mathcal{J}_{23} + \ldots + \mathcal{J}_{n1})$ as the denominator, which is a relative invariant of weight W. We keep the elements in S_0 unchanged, divide the elements in $S_1 \cup S_2$ by $\mathcal{R}_0$, and divide those in $S_3 \cup S_4 \cup S_5$ by $\mathcal{R}_0^2$. This yields a set of differential invariants sufficient to express all second-order fundamental differential invariants. In fact, this set with $6n - 5$ invariants may contain one redundant element, but completeness is of greater concern. To avoid division by zero, we add a positive constant ϵ to the denominator during division, enhancing numerical stability.

Let $\boldsymbol{\mathcal{I}} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)^\top$ denote the set of differential invariants we obtained, which naturally induces an equivariant operator $\hat{\boldsymbol{\mathcal{I}}}$. Theoretically, the invariants $I_1, \ldots, I_N$ are sufficient to express all second-order differential invariants. In practice, we leverage the expressive power of neural networks to combine $I_1, \ldots, I_N$ using a two-layer MLP to produce the output [Li et al., 2024, 2025]. This leads to a learnable equivariant operator:

$$\mathbf{u}_{out} = h_\theta \circ \boldsymbol{\mathcal{I}}(\mathbf{u}_{in}), \tag{33}$$

where h_θ is an MLP parameterized by θ. By integrating this operator into standard network architectures, we can build projective equivariant models. We refer to the resulting model as the **Projective Differential Invariant Network** (**PDINet**), as illustrated in Figure 1.

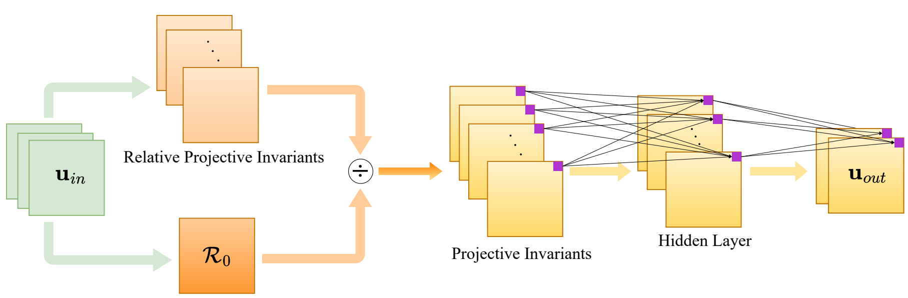

Figure 1: A single layer of PDINet. (1) Compute projective invariants from relative projective invariants. (2) Combine these invariants with an MLP to produce equivariant outputs.

3 Experiments

For empirical evaluation, we conduct image classification tasks under out-of-distribution settings, where models are trained on the original dataset and tested on images deformed by projective transformations. We adopt ResNet-18 [He et al., 2016] as the backbone and replace its convolutional layers with our equivariant operators defined in (33) to construct the projective equivariant network, PDINet. As the main baseline, we consider homConv, a projective equivariant model proposed by MacDonald et al. [2022]. The same backbone is used for a fair comparison. However, homConv relies on group sampling, which leads to exponential memory growth with the number of layers, resulting in out-of-memory (OOM) issues.

3.1 Proj-STL-10

STL-10 [Coates et al., 2011] is a dataset containing 5000 training images and 8000 test images. Each image has a resolution of 96×96 with RGB channels. We apply random projective transformations to the test set to generate the Proj-STL-10 dataset. We train models on the original STL-10 dataset and evaluate them on Proj-STL-10. This evaluation protocol constitutes a challenging out-of-distribution setting, which tests the model's ability to generalize beyond the training distribution.

Each experiment is repeated five times with different random seeds, and we report the average accuracy and standard deviation in Table 1. The results show that PDINet performs much better than ResNet-18 on the transformed test set, achieving a 11.39% improvement.

Table 1: Test accuracy (%) on Proj-STL-10 after training on STL-10.

Model	Accuracy	# Params
ResNet-18	$39.73_{\pm 0.33}$	11.18M
homConv	OOM	-
PDINet (ours)	$\mathbf{51.12}_{\pm 0.47}$	8.90M

3.2 Proj-Imagenette

Imagenette is a ten-class subset of the ImageNet dataset [Deng et al., 2009], consisting of 9469 training images and 3925 test images. All images are adapted to a uniform resolution of 256×256 for model input. We apply random projective transformations to the test set to generate the Proj-Imagenette dataset, while keeping the training set unchanged. This setup simulates an out-of-distribution scenario and evaluates the model's ability to generalize to geometric transformations.

Each experiment is repeated five times with different random seeds, and we report the mean $\pm$ standard deviation of test accuracy in Table 2. Results demonstrate that PDINet retains strong performance under distribution shift, outperforming ResNet-18 by 5.66%. This confirms the effectiveness of the projective equivariance of our model in enhancing out-of-distribution generalization.

Detailed experimental settings and implementation details can be found in the Appendix.

Table 2: Test accuracy (%) on Proj-Imagenette after training on Imagenette.

Model	Accuracy	# Params
ResNet-18	$65.64_{\pm 0.39}$	11.18M
homConv	OOM	-
PDINet (ours)	$\mathbf{71.30}_{\pm 0.45}$	8.90M

4 Conclusion

In this work, we propose a projective equivariant network, **PDINet**, based on second-order differential invariants of the projective group. Our method overcomes the exponential memory growth encountered by homConv [MacDonald et al., 2022], enabling effective scaling to deeper networks. Leveraging the moving frame method and a carefully chosen cross section tailored to multi-dimensional functions, we derive a complete and concise set of second-order projective fundamental differential invariants. Further analysis reveals transformation relationships among projective invariants, allowing us to obtain a unified and simplified formulation that enhances both theoretical clarity and computational efficiency. Building upon these invariants, we design a learnable projective equivariant operator that can be seamlessly integrated into various network architectures. It is the first time to achieve full projective equivariance in deep networks without group sampling or discretization. Experiments under out-of-distribution settings demonstrate the strong generalization ability of our model. With the prevalence and significance of projective transformations in vision, PDINet holds promising potential for broader applications in computer vision.

One limitation of our approach is that the second-order invariants we derive vanish in the one-dimensional case, preventing the direct application of PDINet to grayscale images. In addition, this work focuses on group actions on scalar fields and does not yet cover more general cases involving arbitrary group representations, which we consider a valuable direction for future research.

References

Stan Birchfield. An introduction to projective geometry (for computer vision). *Unpublished note, Stanford university*, 14, 1998.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 834–848, 2017.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

Taco S Cohen and Max Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning*, 2016a.

Taco S Cohen and Max Welling. Steerable CNNs. In *International Conference on Learning Representations*, 2016b.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

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*, 2019.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Maxime W Lafarge, Erik J Bekkers, Josien PW Pluim, Remco Duits, and Mitko Veta. Roto-translation equivariant convolutional networks: Application to histopathology image analysis. *Medical Image Analysis*, 68:101849, 2021.

Erbo Li, Hanlin Mo, Dong Xu, and Hua Li. Image projective invariants. *IEEE transactions on pattern analysis and machine intelligence*, 41(5):1144–1157, 2018.

Yikang Li, Yeqing Qiu, Yuxuan Chen, Lingshen He, and Zhouchen Lin. Affine equivariant networks based on differential invariants. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Yikang Li, Yeqing Qiu, Yuxuan Chen, and Zhouchen Lin. Affine steerable equivariant layer for canonicalization of neural networks. In *The Thirteenth International Conference on Learning Representations*, 2025.

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*, 2022.

Mircea Mironenco and Patrick Forré. Lie group decompositions for equivariant neural networks. In *International Conference on Learning Representations*, 2024.

Roger Mohr and Bill Triggs. Projective geometry for image analysis. In *XVIIIth International Symposium on Photogrammetry & Remote Sensing (ISPRS'96)*, 1996.

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

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

Peter J Olver. Projective invariants of images. *European Journal of Applied Mathematics*, 34(5): 936–946, 2023.

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.

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.

Pourya Shamsolmoali, Masoumeh Zareapoor, Jocelyn Chanussot, Huiyu Zhou, and Jie Yang. Rotation equivariant feature image pyramid network for object detection in optical remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 60, 2021.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Tomas Suk and Jan Flusser. Projective moment invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1364–1367, 2004.

Dian Wang and Robin Walters. SO (2) equivariant reinforcement learning. In *International Conference on Learning Representations*, 2022.

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

Maurice Weiler and Gabriele Cesa. General E(2)-equivariant steerable CNNs. In *Advances in Neural Information Processing Systems*, 2019.

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

Daniel Worrall and Gabriel Brostow. Cubenet: Equivariance to 3D rotation and translation. In *Proceedings of the European Conference on Computer Vision*, 2018.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction strictly follow the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the work in Section 4.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: All theorems in Section 2 are presented with the full set of assumptions. We also provide the complete proofs in the Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all the information for experimental reproduction Section 3 and the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will make the data and code publicly available upon acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so ?No? is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide the experimental setup and implementation details in Section 3 and the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Error bars are reported for all results in Section 3.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide sufficient information on the computer resources in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our paper focuses on the research of equivariant deep learning algorithms, and we do not foresee immediate positive or negative societal outcomes.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: For the code and dataset used in the paper, we respect the license and terms of use, and cite the relevant papers properly.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.