Equivariance in the Era of Large Pretrained Models

Siba Smarak Panigrahi

School of Computer Science
McGill University
Montreal, Canada

August 2024

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Science.

Abstract

Recent advancements in artificial intelligence have highlighted the strengths of large pretrained models across diverse applications. However, these models often struggle with transformations of input data, which is crucial for tasks demanding invariance or equivariance. Redesigning each component of existing architectures to achieve equivariance is difficult and computationally expensive. We investigate the canonicalization framework for designing equivariant architecture and propose a novel prior regularization to align the canonical inputs with orientations present in pretraining datasets. Experimental results indicate that our regularization approach effectively preserves performance while ensuring the robustness of large pretrained and foundation models. Furthermore, to reduce the canonicalization time and tackle the expressivity limitations in equivariant canonicalization networks, we introduce a novel approach of employing non-equivariant pretrained models as canonicalization functions. To facilitate the widespread implementation of our findings, we develop an open-source Python package, equiadapt, that enables retrofitting existing models with equivariance capabilities. This study addresses the challenges in designing equivariant architectures and offers a practical way toward more generalizable and efficient adaptation of AI systems.

Sommaire

Les récents progrès en matière d'intelligence artificielle ont mis en évidence les atouts des grands modèles pré-entraînés dans diverses applications. Cependant, ces modèles ont souvent du mal à gérer les transformations des données d'entrée, ce qui est crucial pour les tâches exigeant l'invariance ou l'équivariance. Repenser chaque composant des architectures existantes pour atteindre l'équivariance est difficile et coûteux en termes de calcul. Nous étudions le cadre de canonisation pour la conception d'une architecture équivariante et proposons une nouvelle régularisation préalable pour aligner les entrées canoniques avec les orientations présentes dans les ensembles de données de pré-entraînement. Les résultats expérimentaux indiquent que notre approche de régularisation préserve efficacement les performances tout en garantissant la robustesse des grands modèles pré-entraînés et des modèles de fondation. De plus, pour réduire le temps de canonisation et résoudre les limitations d'expressivité dans les réseaux de canonisation équivariants, nous introduisons une nouvelle approche consistant à utiliser des modèles pré-entraînés non équivariants comme fonctions de canonisation. Pour faciliter la mise en œuvre généralisée de nos résultats, nous développons une librairie Python en code source ouvert, equiadapt, qui permet de moderniser les modèles existants avec des capacités d'équivariance. Cette étude aborde les défis liés à la conception d'architectures équivariantes et offre un moyen pratique de parvenir à une adaptation plus généralisable et plus efficace des systèmes d'IA.

Previously Published Material

This thesis is based on the equivariant adaptation of large pretrained and foundational models. The motivation for this work is presented in Chapter 1, and an in-depth description is provided in Chapter 3. The outcomes of this research have been published at the 37th Conference on Neural Information Processing Systems as Mondal et al. (2023). As co-first author, I contributed to the entire scope of the research process, from initial identification of the problem, experiment setup and evaluation, particularly for the image domain, and code implementation to writing the publications.

All co-first authors have consented in writing to use the previously published material in this thesis. The extension of Mondal et al. (2023) to EquiOptAdapt and the use of pretrained models as canonicalization networks, described in Section 3.3, is the sole creation of Siba Smarak Panigrahi. Finally, the authors developed and released an open-source Python library, EquiAdapt, available on the Python Package Index (PyPI). Siba Smarak Panigrahi is an active contributor and maintainer of the library, ensuring it stays up-to-date and functional while also interacting with the user community. The repository is hosted on GitHub at `https://github.com/arnab39/equiadapt`.

Dedicated to my parents Snigdha Panda and Ramakanta Panigrahi

Acknowledgments

Firstly, I extend my sincere gratitude to my supervisor, Dr. Siamak Ravanbakhsh, for his steadfast support, mentorship, and flexibility, which allowed me to explore various research problems before finalizing my Master's thesis topic. Two years ago, when I was searching for a research-based Master's program to develop my research skills and portfolio, he offered me the opportunity to join McGill University, the Mila community, and his lab, and I could not have asked for a better experience. His unwavering commitment to fostering my skill development and research proficiency has been invaluable. I am immensely grateful for his dedication to nurturing my skills and fostering my growth in a research environment.

I am deeply grateful to my research collaborator, Arnab Kumar Mondal, whose insightful discussions and guidance on research problems and his personal advice regarding research have been immensely beneficial. His exceptional work ethic, commitment to research ideas, and collaborative spirit have been a constant source of inspiration.

My appreciation extends to my lab colleagues for their continual support in brainstorming innovative ideas and the enjoyable interactions within and beyond lab meetings. Special thanks are due to Victor Livernoche for his assistance in translating this thesis's abstract into French.

I also acknowledge the McGill Computer Science Graduate Society (CSGS) and the Mila-Quebec AI Institute. Mila, especially, became a second home where I formed lasting friendships through foosball, board games, cultural events, and social gatherings. The exceptional computing resources and technical support from Mila's skilled IT staff were crucial to this research.

Special thanks go to my family for their unwavering support and my friends, both in Montreal and abroad, who have made the past two years enjoyable and memorable.

Lastly, I pray the Almighty to bless me with the strength and wisdom to continue striving for excellence in all my endeavours.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
GPT	Generative Pretrained Transformer
SAM	Segment-Anything Model
C_n	Cyclic Group of order n
D_n	Dihedral Group of order n
$SO(n)$	Special Orthogonal Group in dimension n
$SE(n)$	Special Euclidean Group in dimension n
$O(n)$	Orthogonal Group in dimension n
$E(n)$	Euclidean Group in dimension n
$GL(n)$	General Linear Group in dimension n
2D	2-Dimensional
3D	3-Dimensional
CNN	Convolutional Neural Network
G-CNN	Group Convolutional Neural Network
Irreps	Irreducible Representations
Unif	Uniform
CIFAR	Canadian Institute For Advanced Research
MNIST	Modified National Institute of Standards and Technology
KL	Kullback-Leibler
COCO	Common Objects in Context

STL	Self-Taught Learning
ResNet	Residual Network
ViT	Vision Transformer
MaskRCNN	Mask Region-based Convolutional Neural Network
WRN	Wide Residual Network or WideResNet
ReLU	Rectified Linear Unit
LC	Learned Canonicalization
VN	Vector Neurons
DGCNN	Deep Graph Convolutional Neural Network

Chapter 1

Introduction

Large pretrained models have gained a pivotal role in the artificial intelligence (AI) community, showcasing remarkable capabilities across a broad spectrum of tasks. These models leverage vast amounts of data, use massive amounts of computing resources to learn complex patterns, and have fundamentally transformed our approach to solving several computational problems. A few examples include language generation (Achiam et al., 2023; Le Scao et al., 2023), document understanding (Liu et al., 2024b; Hu et al., 2024), image classification and segmentation (Dosovitskiy et al., 2020; Radford et al., 2021; Kirillov et al., 2023), image and video generation (Rombach et al., 2022; Saharia et al., 2022; Singer et al., 2022; Hu et al., 2023), weather forecasting (Lam et al., 2022; Ravuri et al., 2021), material generation (Gruver et al., 2023b; Jiao et al., 2023; Merchant et al., 2023; Zeni et al., 2023), protein structure prediction (Abramson et al., 2024) and molecular docking (Corso et al., 2022).

Crucially, these applications often require that the models recognize and adapt to the inherent symmetries and variations in their inputs and tasks, ensuring robustness against diverse transformations, such as rotation, translation, and scaling. For instance, deep learning models tailored for image classification require an object to be invariably recognized, such as a cat, even when the image undergoes Euclidean transformations. Further,

models developed for predicting crystal properties are expected to treat each crystal with uniformity, irrespective of its spatial orientation, since the intrinsic properties of the crystal remain unchanged. Such tasks are referred to as invariant tasks. On the other hand, an example of an equivariant task is image segmentation, wherein the model is required to segment specific objects. Consequently, it must adjust its output in response to any transformations in the input.

1.1 Motivation

1.1.1 Equivariant networks

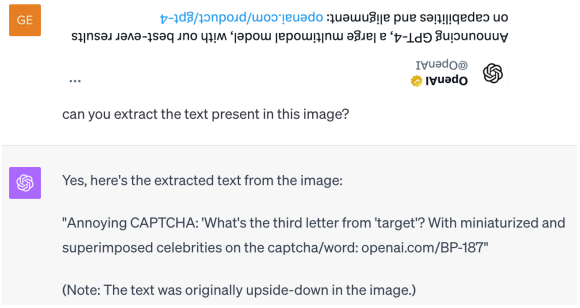
Equivariant and invariant networks are specially crafted to tackle these equivariant and invariant tasks. The architecture of these networks inherently incorporates the principle of equivariance to a certain group of transformations. The most popular way is to incorporate equivariance into the architecture design (Weiler and Cesa, 2019; Cesa et al., 2022). Each layer of such a neural network is designed to be equivariant, and the composition of those layers guarantees the equivariance of the whole model. Equivariant networks also offer additional significant advantages, including:

1. **Enhanced Generalization**: These networks can generalize across various transformations of input data, facilitating robust model performance under these diverse transformations (Gordon et al., 2020; Elesedy and Zaidi, 2021; Mao et al., 2023).

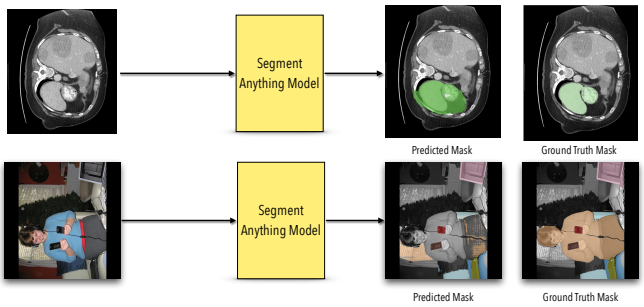
2. **Efficient Learning**: Learning from a single data orientation is equivalent to learning from all possible orientations. This attribute significantly reduces the data required for training (Wang et al., 2021; Batzner et al., 2022; Mondal et al., 2022).

3. **Parameter Efficiency**: Equivariant networks promote parameter efficiency by sharing weights across different transformed states of the input.

1.1.2 Equivariance of large pretrained models



(a) GPT-4 (Achiam et al., 2023) cannot perform text detection from inverted input images.



(b) MedSAM (Ma et al., 2024) and SAM (Kirillov et al., 2023)) are not robust to rotations for segmentation tasks. Segmentation masks for medical images are generated with MedSAM demo (Link).

Figure 1.1 Issues with large pretrained models on transformed inputs.

Existing large pretrained and foundation models (Bommasani et al., 2021) are typically not equivariant to most transformations. However, their wide applications across several domains require them to be robust to unseen transformations and orientations of images. For instance, Vision Language Models (or VLMs) are designed on top of large language models (LLMs) to take a text prompt along with images for structured document understanding (Hu et al., 2024; Tong et al., 2024; Dong et al., 2024; Liu et al., 2024b,a). Amongst several tasks, VLMs attempt to solve question-answering, parsing, and information extraction by leveraging the image inputs. In the real world, users capture input document images at various camera angles and then provide the image to these systems. Thus, it necessitates that these models be robust to understand transformed inputs, but Figure 1.1a demonstrates that a recent, popular, and one of the best LLMs, GPT-4 (Achiam et al., 2023), is not robust to trivial rotations.

Another set of examples comes from the systems built on top of the Segment-Anything Model or SAM (Kirillov et al., 2023). Some systems are expected to aid diagnosis in the medical domain (Ma et al., 2024; Zhu et al., 2024). Thus, such systems should be able to accurately detect or segment portions of X-ray, CT Scan, or MRI images in several

orientations and failure would lead to severe consequences. Again, in the real world, users capture images from any camera angle, and SAM would be required to segment instances irrespective of the input orientation. However, in Figure 1.1b, we observe that both MedSAM and SAM are robust to rotated image inputs.

This motivates us to build equivariant large pretrained models.

1.1.3 Towards equivariant large pretrained models

One straightforward strategy involves replacing individual layers of large pretrained networks with equivariant counterparts. While this modification ensures equivariance, it introduces considerable challenges:

1. **Resource intensive Re-training**: Re-training these large models from scratch with massive amounts of data demands considerable computational time and financial resources, potentially costing millions of dollars, with additional logistical and environmental impacts.

2. **Complex Design Requirements**: Crafting equivariant versions of complex operations presents non-trivial technical difficulties. For instance, the design of equivariant transformer architecture [1] poses a non-trivial effort (Romero and Cordonnier, 2020; Xu et al., 2023).

3. **Parameter Increase**: Large number of parameters in pretrained models leads to expensive equivariant counterparts with several parameters, resulting in more costly models that require longer inference times (Kaba et al., 2023).

These challenges highlight the need for an architecture-agnostic solution to integrate equivariance into large pretrained models.

[1] commonly uses self-attention mechanism (Vaswani et al., 2017)

1.2 Statement of Contributions

In this study, we investigate the adaptation of learning canonicalization functions for predicting canonical input orientations (Kaba et al., 2023) to enhance the robustness of existing large pretrained models, resolving the issues with equivariance in Section 1.1.3 while preserving both performance and inference speeds.

Our initial findings reveal that integrating equivariant canonicalization functions with a large pretrained model and applying a single task-specific loss diminishes the model's downstream performance. To counter this, we introduce a novel prior regularization loss that aligns the canonical outputs with the orientations in the pretrained model's original training dataset. This approach effectively maintains the pretrained model's original performance and ensures equivariance. We substantiate our claims through experimental setups that employ both standard testing datasets and their transformed counterparts.

Additionally, we recognize a significant challenge when using equivariant networks as canonicalization functions. Primarily, these networks require a high level of expressiveness to accurately map inputs to canonical orientations across extensive datasets, which leads to longer input canonicalization times. To address this, we demonstrate that non-equivariant pretrained models can effectively serve as canonicalization functions. This approach ensures equivariance while alleviating the previously mentioned processing time delays.

Finally, we release a Python package, `equiadapt`, for users to build on and easily integrate the idea of canonicalization with their existing models to convert them into equivariant models. The code for the same can be found in GitHub[2].

[2]https://github.com/arnab39/equiadapt

1.3 Organisation of This Work

In Chapter 2, we give an overview of relevant mathematical concepts, such as group theory and equivariance, along with related literature. We describe the problems with the existing canonicalization framework and introduce a well-motivated novel prior regularization and non-equivariant canonicalization networks in Chapter 3. We detail our experiments in Chapter 4 to support our claims and demonstrate the effectiveness of our proposed methods. Finally, we summarize our work and provide potential future directions in Chapter 5.

Chapter 2

Background

This chapter provides a detailed review of key mathematical concepts and relevant literature on equivariance, focusing on the design of equivariant architectures. We explore traditional methods for composing equivariant layers to construct equivariant networks and architecture-agnostic strategies for developing equivariant networks. Finally, we identify and discuss the limitations inherent in these approaches when adapting large pretrained models to achieve equivariance.

2.1 Group Theory and Equivariance

2.1.1 Groups, group actions, and group representations

A group is defined as a set of elements that includes an identity element e under an associative binary operation $\circ$. This operation is closed within the group $\mathcal{G}$, meaning for any two elements $a, b \in \mathcal{G}$, the result of the operation, $a \circ b$, also belongs to $\mathcal{G}$. Furthermore, every element $g \in \mathcal{G}$ has a unique inverse g^{-1}, satisfying $g \circ g^{-1} = e$. Similarly, a subgroup $\mathcal{H}$ is a subset of elements in $\mathcal{G}$ satisfying the above properties. A subgroup's size (or order) is always a divisor of the size of the group. Common examples of discrete groups encompass permutation groups, denoted as S_n, which represent all permutations of the

set $\{1, 2, \ldots, n\}$, cyclic groups C_n representing n discrete rotational symmetries, and dihedral groups D_n, comprising n discrete rotations coupled with reflections. Examples of continuous groups include $SO(n)$, which describes continuous rotations in n dimensions, $O(n)$, extending $SO(n)$ to include reflections. The Euclidean group $E(n)$ incorporates roto-reflections and translations in n dimensions, i.e., Euclidean transformations that preserve the Euclidean distance between two points.

Groups play a crucial role in understanding symmetry transformations via group actions. A group action enables an element g from the group $\mathcal{G}$ to act on an element w in a set Ω and defined as a mapping $(g, w) \rightarrow g.w$ where $g.w \in \Omega$. This is typically also noted as left group action. The action of the *identity* group element is an identity action which leaves w unchanged, i.e., $e.w = w$. Similarly, $w.g$ is the right group action. The concept of group action can be extended to spaces defined by signals on the underlying set Ω such as $X(\Omega)$ with $(g.x)(w) = x(g^{-1}.w)$. Note that, in both cases, the group action has the compositional property, i.e., $g.(h.w) = (g \circ h).w$ and $(g.(h.x))(w) = ((g \circ h).x)(w)$, where $g, h \in \mathcal{G}$. A few concepts are related to group action, widely used in the literature: assume a group $\mathcal{G}$ acts on Ω, then,

- *fixed point* of g is an element $w \in \Omega$ such that $g.w = w$
- *stabilizer* $\mathcal{G}_w$ of $w \in \Omega$ is the subgroup $\{g\}$ of $\mathcal{G}$ such that w is a fixed point of g
- *orbit* of an element w is the set of elements $u \in \Omega$, such that $g.w = u$ for some $g \in \mathcal{G}$.

Linear group actions, often termed *group representations*, map each group element g to an invertible matrix $\rho(g)$. This mapping is described by $\rho(g) : \mathcal{G} \rightarrow GL(\Omega)$, where $GL(\Omega)$ represents the set of all invertible linear transformations applicable to the space Ω. For example, the permutation group S_n representation is given by permutation matrices that rearrange standard basis vectors. Similarly, for the group $SO(2)$, one of the group representations involves 2D rotation matrices, which can act on an image to rotate around

a specific center point. Concretely, in the case of images, Ω signifies the underlying 2D discrete grid, and $X(\Omega)$ indicates a signal defined on this grid, which is the image itself. An element $g \in SO(2)$ can therefore rotate both the grid (Ω) and the image ($X(\Omega)$).

2.1.2 Equivariance and invariance

A function $f : X(\Omega) \to \mathcal{Y}$ is *equivariant* to $\mathcal{G}$ or $\mathcal{G}$*-equivariant* if $f(\rho_{X(\Omega)}(g)x) = \rho_{\mathcal{Y}}(g)(f(x))$ $\forall g \in \mathcal{G}$ and $x \in X(\Omega)$. This condition implies that applying the function f commutes with the action of the group $\mathcal{G}$; the order in which the group action and the function are applied does not alter the result. Since the group representation may differ in the spaces $X(\Omega)$ and $\mathcal{Y}$, they are denoted with $\rho_{X(\Omega)} : \mathcal{G} \to GL(X(\Omega))$ and $\rho_{\mathcal{Y}} : \mathcal{G} \to GL(\mathcal{Y})$. $\mathcal{G}$*-invariance* is a special case of $\mathcal{G}$*-equivariance* when $\rho_{\mathcal{Y}}$ is trivial, $\rho_{\mathcal{Y}}(g) = 1 \forall g \in \mathcal{G}$. Therefore, a function $f : X(\Omega) \to \mathcal{Y}$ is *invariant* to $\mathcal{G}$ or $\mathcal{G}$*-invariant* if $f(\rho_{X(\Omega)}(g)x) = f(x) \forall g \in \mathcal{G}$ and $x \in X(\Omega)$.

These concepts of equivariance and invariance are critical in many deep learning applications, where they help define the changes in a function's output against specific input transformations. For instance, in the context of image classification, translation (or shift) invariance is a desired property where the identification of an object in an image should not depend on its position. This implies that the function f, responsible for classifying images, should be translation invariant. Similarly, translation equivariance is vital in instance segmentation and object detection tasks. This property ensures that the predicted segmentation maps or bounding boxes adjust accordingly if the input image is shifted. Thus, the functions applied in these applications must demonstrate translation equivariance to track accurately and segment objects as their positions change. These characteristics are essential for robust machine learning models that generalize well across varying inputs.

2.2 Equivariant Networks from Composition of Equivariant Layers

Several methods exist to build equivariant networks, such as parameter sharing (Ravanbakhsh et al., 2017) and self-supervision (Dangovski et al., 2021). However, in this section, we review one of the most widely-used approaches in the form of group convolution or linear equivariant layers.

Traditional Convolutional Neural Networks (or CNNs) excel in processing grid-based signals, such as images (LeCun et al., 1995; Ren et al., 2015; He et al., 2016; Redmon et al., 2016). They are structured with multiple convolutional layers, each employing convolution kernels or filters. Mathematically, consider a set of N feature maps arranged on a discrete grid, denoted by $F : \mathbb{Z}^2 \to \mathbb{R}^N$. In a convolutional layer containing L convolution kernels, each represented as $\psi^i : \mathbb{Z}^2 \to \mathbb{R}^N$, the convolution operation is defined by the equation: $[F * \psi^i](x) = \sum_{y\in\mathbb{Z}^2} \sum_{n=1}^{N} F_n(y)\psi_n^i(y - x)$. This operation allows the layer to extract and leverage spatial and temporal relationships within the data. The inherent structure of CNNs promotes translation equivariance, meaning the convolution operation remains consistent under shifts (i.e., commutes with translation) in the input data.

Discrete Group Equivariant CNNs (Discrete G-CNNs), introduced in
(Cohen and Welling, 2016a), extend conventional CNNs to discrete groups such as $p4m$ and $p4$, which encompass translations coupled with rotations by 90 degrees, with and without and flips, respectively. This architecture incorporates two specialized convolution operations: lifting convolution and group convolution:

- The **lifting convolution** is the initial layer, which takes a 2D feature map and convolves it across the specified group. This is mathematically represented as $[F * \psi](g) = \sum_{y\in\mathbb{Z}^2} \sum_n F_n(y)\psi_n(g^{-1}y)$, where both F and ψ operate over $\mathbb{Z}^2$, but the outcome, $F * \psi$, is mapped onto the group $\mathcal{G}$.

- Subsequent layers utilize **group convolutions**, where the convolution kernels themselves are defined on the group $\mathcal{G}$, i.e., $[F * \psi](g) = \sum_{h\in\mathcal{G}} \sum_n F_n(h)\psi_n(g^{-1}h)$.

This design is followed by projection or pooling layers to adapt to the specific demands of the target task. In summary, the G-CNNs are a more general version of CNNs where the underlying equivariance is extended to a group $\mathcal{G}$ instead of only translation (Kondor and Trivedi, 2018; Cohen et al., 2019b; Bekkers, 2020).

Subsequent research has broadened the scope of discrete group equivariant CNNs to other 2D and 3D transformations, addressing complex symmetries. Notably, Hoogeboom et al. (2018) introduced a model incorporating 6-fold planar rotational symmetry through hexagonal tiling. Additionally, extensions to three-dimensional transformations include CubeNet (Worrall and Brostow, 2018), which integrates translations and rotations at right angles in three-dimensional grids, and 3D roto-translation groups for enhancing pulmonary nodule detection (Winkels and Cohen, 2022). Lenssen et al. (2018) extend the equivariance idea to capsule networks (Hinton et al., 2011) and Marcos et al. (2017) to vector field networks.

Continuous rotation G-CNNs have been further developed, albeit with regular discretizations featuring $SE(2)$-group convolution layers. These layers have been successfully applied in tasks such as histopathology image analysis and retinal imaging (Bekkers et al., 2017, 2018; Lafarge et al., 2021), demonstrating improved sample efficiency and superior performance compared to traditional CNNs augmented with data rotation techniques.

Worrall et al. (2017) introduce Harmonic networks as CNNs for equivariance to $SO(2)$ by replacing regular convolution kernels with circular harmonics. Steerable CNNs (Cohen and Welling, 2016b) presented another framework to design continuous group equivariant networks. In this framework, 2D signals are bundles of feature fibres instead of traditional stacks of feature maps. Each fibre contains feature vectors. They describe the construction of $\mathcal{H}$-equivariant kernels with the help of a predefined basis that maps an input signal (bundles of feature fibres) to feature vectors. The framework extends to $\mathcal{G}$-steerable networks (e.g., $p4m$), with $\mathcal{H}$ (e.g., D_4) being a discrete subgroup of $\mathcal{G}$ with induced group

representation. The paper demonstrates the effectiveness of Steerable CNNs in image classification tasks. Building on this work, (Weiler and Cesa, 2019) proposed a general theory with irreducible representations to design $E(2)$-equivariant steerable CNNs. This theory applies to the $E(2)$ group and subgroups. The key here is defining and solving the kernel constraint under irreducible representations of the group, which is crucial for designing effective kernels in these networks. The approach also leverages irreducible representations (*irreps*) to facilitate a change of basis. This is particularly useful in managing different types of representations found in neural network layers, such as trivial, regular, and quotient representations. Finally, (Cesa et al., 2022) extended these concepts to develop a framework for building $E(n)$-equivariant steerable CNNs.

While these methods are fundamental for developing equivariant neural networks, they present notable challenges, particularly when considering large pretrained and foundation models. Firstly, constructing such large networks from scratch is a non-trivial task. Secondly, retraining these extensive models from scratch demands significant data and computational resources, which entail additional consequences. We have discussed these issues in detail in Section 1.1.3. Consequently, our attention turns to architecture-agnostic strategies that guarantee equivariance with minimal or no modifications to the existing architecture.

2.3 Equivariant Networks from Model Agnostic Approaches

2.3.1 Symmetry regularization

Shakerinava et al. (2022) explore a complex scenario with unknown and potentially non-linear group actions. Equivariance is obtained through objective functions rather than architectural design. The objective function comprises injectivity promoting and invariance of the considered group. They show that an injective function $f : \mathcal{X} \to \mathcal{Z}$, enforced with logarithmic barrier function or hinge loss, can behave as an equivariant map to any transformation function $t_{\mathcal{X}}$ in the input space $\mathcal{X}$, and transformation function $t_{\mathcal{Z}}$ in the embedding space $\mathcal{Z}$. Here, $t_{\mathcal{X}}$ is unknown, whereas $t_{\mathcal{Z}}$ is inherently complex and can be pushed towards a simple (linear) action of our choice. The authors ensure the regularization of the latent embedding within the constraints of $E(n)$ or $O(n)$ groups, aiming to preserve either the Euclidean distance or the orthogonal distance (inner product), which aligns with the group invariants. This framework is termed symmetry regularization.

(Gupta et al., 2023) extended this work to propose an equivariant contrastive learning framework where an additional invariance loss was used along with the above objective functions to ensure minimal shifts in the embedding space in response to slight augmentations or transformations in the input space.

Despite this approach's potential, its application to large pretrained models faces practical constraints. Specifically, the process requires multiple expensive forward passes through the large pretrained model f. Furthermore, we know apriori, the exact transformation group to which the model's equivariance should be adapted.

2.3.2 Symmetrization and frame averaging

Symmetrization refers to the fact that any arbitrary function can be made $\mathcal{G}$-equivariant or $\mathcal{G}$-invariant by averaging over the group (Yarotsky, 2022). For example, given function $\Phi : V \to W$, then

$$\Psi(X) = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} g.\Phi(g^{-1}.X) \tag{2.1}$$

is $\mathcal{G}$-equivariant. Similarly, invariant functions can be designed when the action of g on Φ is trivial. Murphy et al. (2018) used this approach to design permutation-invariant functions.

Symmetrization in its vanilla form becomes computationally infeasible over larger groups due to the forward passes with all transformations in $\mathcal{G}$ of the input. Therefore, several works have limited their analysis to small finite groups. Basu et al. (2023b) proposes equi-tuning as a fine-tuning method to convert pretrained non-equivariant models to equivariant models through symmetrization and support their claims across a diverse set of tasks, including image and natural language tasks. Equi-tuning employs an equal weighting scheme for outputs from the pretrained model, whereas Basu et al. (2023a) devises a learnable weighting scheme to perform a weighted average of transformed outputs.

A key observation in Puny et al. (2021) is that the group average over the entire group can be replaced with averaging over an input-dependent subset of group elements, i.e., $\mathcal{F}(X) \subset \mathcal{G}$, where $|\mathcal{F}(X)|$ is typically small and $\mathcal{F}(X)$ is $\mathcal{G}$-equivariant. The main findings highlight that the frame-averaging framework can be used for designing universal functions. This approach also retains the expressive power of the original backbone function. In contrast, equivariant networks generally are computationally expensive (Morris et al., 2019; Kim et al., 2021; He et al., 2021; Kaba et al., 2023) and lack expressivity (Xu et al., 2018; Maron et al., 2019; Azizian et al., 2020; Zhang et al., 2022; Joshi et al., 2023) due to additional constraints on network design to guarantee equivariance. The effectiveness of this approach to build equivariant network has been demonstrated across several applications, particularly in shape space learning (Atzmon et al., 2022), materials modelling (Duval et al., 2023), point cloud processing (Finkelshtein et al., 2022), and 3D shape anal-

ysis (Li et al., 2023).

In principle, Equation 2.1 can be alternatively written as an expectation over a uniform distribution defined on the group $\mathcal{G}$, i.e.,

$$\Psi(X) = \mathbb{E}_{g \sim Unif(\mathcal{G})}[g.\Phi(g^{-1}.X)] \tag{2.2}$$

Therefore, a sampling-based average can estimate Equation 2.2 (Murphy et al., 2018, 2019). Kim et al. (2023) demonstrate that the uniform distribution $Unif(\mathcal{G})$, can be replaced with a parameterized distribution $p_\theta(g|x)$ and the whole framework is equivariant as long as $p_\theta(g|x)$ is probabilistically equivariant (Bloem-Reddy et al., 2020). The authors use reparameterization (Kingma and Welling, 2013) to replace $p_\theta(g|x)$ with a combination of invariant noise ϵ and equivariant network q_θ.

While symmetrization is an approach for adapting large pretrained models to exhibit equivariance, it faces challenges due to the requirement for multiple, computationally intensive forward passes through large pretrained models during fine-tuning and inference. This arises from the necessity of averaging the outputs across various input transformations, thereby substantially increasing computational overhead. Consequently, scaling this method to large-scale foundation models, such as the Segment-Anything Model (SAM, Kirillov et al. (2023)) and GPT-4V (Achiam et al., 2023), proves challenging. The ideal solution for equivariant adaptation would seamlessly integrate with these models as a plug-and-play module, reducing the number of necessary forward passes to one, thus enhancing feasibility and efficiency.

2.3.3 Canonicalization

In canonicalization, the function Φ only processes one canonical (or standard) orientation. For the invariance task, mapping all members in the orbit of an input to a canonical input from the orbit before the function application is enough. For equivariance, elements are

mapped to a canonical sample and following function application, the outputs are transformed back according to their original position in orbit. The setup can be formulated as

$$\Psi(x) = c'(x).\Phi(c(x)^{-1}.x) \tag{2.3}$$

where $c : X \to \mathcal{G}$ is a canonicalization function, $\Phi : X \to Y$ can be any prediction function and $\Psi : X \to Y$ its equivariant version, $c(x)^{-1}.x$ is the canonical input in the orbit of x, and $c'(x)$ is the counterpart of $c(x)$ on the output. It is implicit that $c(x)$ and $c'(x)$ act on x and output of Φ with the group representation ρ_X and ρ_Y respectively. We define canonicalizer as a function which outputs the canonical input, i.e., $\mathcal{C}(x) = c(x)^{-1}.x$.

Several earlier works leverage hand-engineered heuristics to canonicalize inputs (Lowe, 2004) with poor generalization. Spatial transformer (Jaderberg et al., 2015) learns input transformations across layers, thus requiring modifications in the architecture for easier processing in downstream vision tasks. Its equivariant version (Esteves et al., 2018b; Tai et al., 2019) fixes a canonical coordinate but fails to handle groups larger in dimension than the size of the underlying grid.

Kaba et al. (2023) demonstrate that $c(x)$ can be learned and it is enough for $c(x)$ to be continuous and $\mathcal{G}$-equivariant i.e., $c(\rho_X(g).x) = \rho_{\mathcal{G}}(g).c(x)$ for Ψ to be a universal approximator of $\mathcal{G}$-equivariant functions. Additionally, we could impose certain symmetries to $\Phi(x)$ and perform partial canonicalization for additional symmetries with $c(x)$[1]. For instance, a CNN could be made $E(2)$-equivariant with an $O(2)$-equivariant $c(x)$, and $p4$-equivariant with C_4-equivariant $c(x)$.

They jointly train an equivariant canonicalization network $c(x)$ with relatively small parameters and a prediction network $\Phi(x)$ of choice from scratch, where the canonicalization network learns to output a suitable canonical orientation. The approach's effectiveness is demonstrated with larger discrete groups where it maintains the expressivity of Φ at a much lower parameter cost than G-CNNs (Cohen and Welling, 2016a).

[1]Theorem 3.1 and 3.2 in Kaba et al. (2023)

Since these approaches require single forward pass through prediction models due to a single (canonical) orientation while being model-agnostic, we focus on learned canonicalization (Kaba et al., 2023) as a suitable approach for equivariant adaptation of large pretrained models. We first analyze the problems with trivially using the framework for our goal and propose appropriate solutions and improvements.

Finally, since we demonstrate the use of our proposed approach on pretrained models for image-based applications, in the above literature review, we concentrated on equivariant networks designed for image processing that address global symmetries, excluding local gauge transformations (Cohen et al., 2019a). Additionally, we do not cover equivariant networks tailored for processing sets (Zaheer et al., 2017; Qi et al., 2017), meshes (De Haan et al., 2020), general graphs (Maron et al., 2018; Gasteiger et al., 2019; Atz et al., 2021; Brandstetter et al., 2021; Satorras et al., 2021), and arbitrary manifolds (Weiler et al., 2021).

Chapter 3

Equivariant Adaptation of Large Pretrained Model

In this chapter, we first introduce the process of integrating the canonicalization framework with large pretrained models, aiming to develop equivariant large pretrained models and also provide critical insights that underscore the adoption of a novel prior regularization within this framework. Subsequently, we address concerns about the expressivity and canonicalization time of equivariant canonicalization networks and explore non-equivariant pretrained models as alternatives for these networks. Effectively, we propose a pipeline that utilizes non-equivariant networks to achieve equivariance. Lastly, we detail our experimental setup and methodologies to validate and support our claims.

3.1 Designing Equivariant Canonicalization Networks

As motivated in Section 2.3.3, we build upon the learned canonicalization framework to achieve equivariant adaptation of large pretrained models. The framework requires learning an equivariant canonicalization network $c(x)$ simultaneously with a prediction network $\Phi(x)$. Two approaches were proposed in Kaba et al. (2023) for designing $c(x)$:

direct approach and *optimization approach*. We interchange network with function in the text.

3.1.1 Direct approach

Existing equivariant networks (refer to Section 2.2) can be selected as canonicalization networks and trained to output group elements. This is referred to as *direct approach*.

As an example, consider using a C_4-equivariant G-CNN (Cohen and Welling, 2016a) as a canonicalization network, which outputs a group element $g \in C_4$. Due to the design, each layer operation in $c(x)$ is equivariant to C_4. Thus, if input x was transformed with $g_1 \in C_4$, then the output group element will be $\rho_{\mathcal{G}}(g_1).c(x)$.

3.1.2 Optimization approach

Another method suggests that $c(x)$ can be defined as the set of elements that minimize an arbitrary function $q : \rho(\mathcal{G}) \times X \to \mathbb{R}$, if $q(\rho(g), x)$ satisfies the following constraints (here, q can be a neural network):

1. *Equivariance*
$$q(\rho(g), \rho(g_1).x) = q(\rho(g_1)^{-1}\rho(g), x), \forall g_1 \in \mathcal{G} \tag{3.1}$$

2. *Uniqueness* up to input symmetry. The outcome of the optimization process can lead to non-unique canonical orientations in a single orbit. In this case, we can select one of them as the canonical input (Kaba and Ravanbakhsh, 2023).

Equivariant networks trivially satisfy these constraints. However, the interesting alternative is a non-equivariant network that represents a distance function (or energy), i.e., $q(\rho(g), x) = E(\rho(g)^{-1}.x)$ and is minimized at canonical inputs through optimization. We build upon this idea and propose a contrastive loss to train existing non-equivariant architectures and obtain equivariant canonicalization networks in Section 3.3.

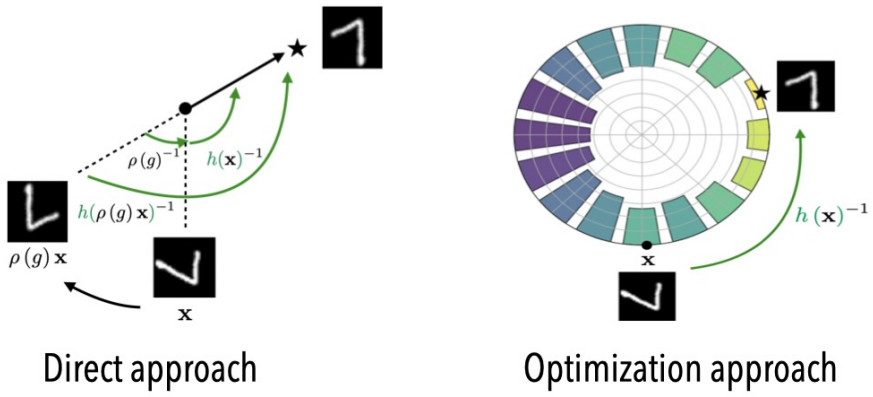


Figure 3.1 Direct approach and optimization approach for canonicalization. In both methods, our goal is to predict the group element(s) that can be used to canonicalize. Figures adapted from Kaba et al. (2023).

3.2 EquiAdapt

The canonicalization framework allows equivariant adaptation of existing large pretrained models. In our initial investigation, we train an equivariant canonicalization function $c(x)$ while optionally finetuning the large pretrained model $\Phi(x)$ using the same task objective. We consider both zero-shot and finetuning setups to derive insights regarding the training dynamics and propose a novel prior regularization.

3.2.1 Augmentation and alignment

Alignment. The performance of the above combination requires the *alignment* of canonicalization and prediction function. For example, suppose the canonicalization network produces images in a less preferred orientation (e.g., upside-down) compared to those the pretrained network expects. In that case, the overall performance is significantly degraded.

Augmentation. In addition to alignment, there is an *augmentation* effect that further adds complexity: during its training, the canonicalization network performs data augmentation with respect to the group $\mathcal{G}$, where the canonicalization network is $\mathcal{G}$-equivariant.

Suppose a prediction network is engineered to be equivariant to group $\mathcal{G}$ through canonicalization. At the start of training, the randomly initialized weights of the canonicalization function produce random canonical orientations for each data point. This mimics the data augmentation effect with transformations through elements of group $\mathcal{G}$ for the prediction network. As the training progresses, the canonical orientations for visually similar images gradually converge, as illustrated in Figure 3.2, thereby reducing the augmentation effect. Consequently, canonicalization ensures equivariance and offers an additional augmentation effect to the prediction network.

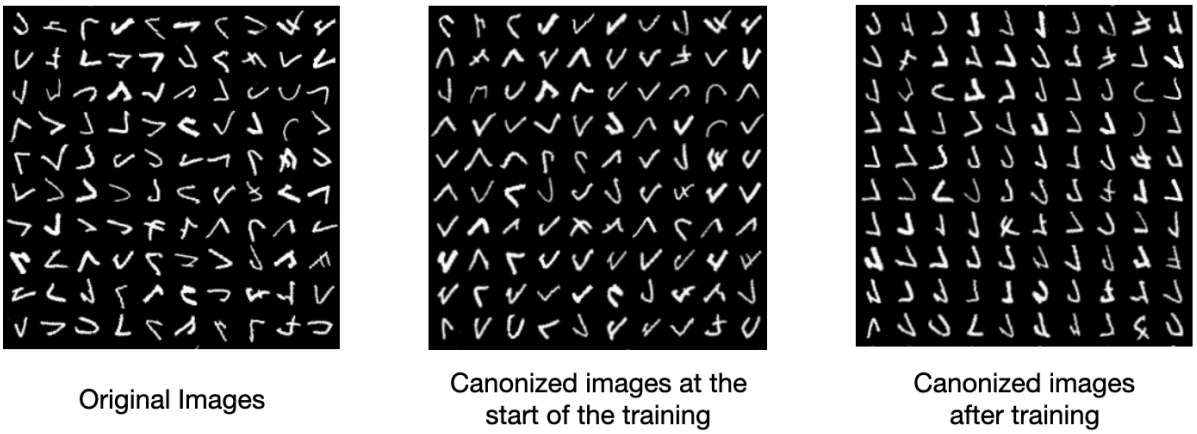


Figure 3.2 Visualization of the diminishing augmentation effect introduced by learning canonicalization (Kaba et al., 2023) during training for the Rotated-MNIST dataset (Larochelle et al., 2007). In this visualization, the leftmost image represents the original training images. Moving towards the center, we present the canonicalized images at the beginning of the training process. Finally, the rightmost image unveils the transformation of the canonicalized images after training the model for 100 epochs.

However, there are two scenarios where the augmentation provided by learning the canonicalization function can be detrimental:

First, augmentation with the full range of transformations in a group $\mathcal{G}$ can hamper the early stages of *training* or *fine-tuning* if the training benefits from smaller changes to the input. This is particularly pronounced in natural image datasets like CIFAR datasets (Krizhevsky et al., 2009), where image transformations with small rotation angles (ranging from $-10°$ to $+10°$) are typically advantageous. In contrast, a canonicalization function might introduce rotations spanning from $-180°$ to $+180°$ at the beginning of training. Further, such extensive input modifications can destabilize the prediction network's training, leading to a diminished performance by exposing it to data that deviates significantly from the training distribution. This phenomenon is illustrated in Table 4.1, where we compare the impact of different rotational augmentations — including those introduced by learned canonicalization (Kaba et al., 2023) — on the performance of prediction networks trained with CIFAR datasets. The decrease in performance tends to be more pronounced in networks trained from scratch on more complex datasets featuring a higher number of labels. This scenario exemplifies the *variance-invariance tradeoff* described by Chen et al. (2020a). Training with arbitrary augmentations biases the prediction function since the test distribution is not perfectly symmetric under rotations.

Second, although Kaba et al. (2023) advocates relying solely on the task loss objective to learn the canonicalization network, we observe that it is insufficient to learn the correct orientation when used with large pretrained models. To support this hypothesis, in Figure 4.2a, we plot the distribution over the canonical orientations during inference and notice that the canonical orientations from the canonicalization function are inconsistent with the desired canonical outputs for the prediction network, impacting its performance. This could be due to the small size of the finetuning dataset compared to the pretraining dataset. For instance, on the CIFAR10 dataset (Krizhevsky et al., 2009) without any augmentations, we expect the canonical orientation for every data point to be identical after training. However, from Figure 4.2a, we can see that the canonical orientations for the test set are distributed uniformly from $-180°$ to $+180°$ after training until convergence of

the task objective. Consequently, during inference, the prediction network will receive images with different orientations and underperform. This issue arises because the prediction networks are not inherently robust to these transformations.

Therefore, when analyzing the performance of this type of equivariant network, both alignment and augmentation effects must be considered.

- When both networks are trained together from scratch, the alignment is a non-issue, and (unwanted) augmentation can degrade or improve performance, depending on the extent of symmetry in the dataset, e.g., in Kaba et al. (2023), augmentation effect enabled equivariant networks in canonicalization framework to achieve higher performance on Rotated-MNIST dataset (Larochelle et al., 2007) while as shown in Table 4.1, the performance degrades due to unwanted augmentation in both CIFAR10 and CIFAR100 datasets (Krizhevsky et al., 2009).

- However, when dealing with pretrained prediction networks, one must consider the alignment effect. One could then think of freezing the pretrained prediction network, avoiding unwanted augmentation and backpropagating the task loss to align the canonicalization network. However, task loss alone is not sufficient, and this can become computationally expensive for large pretrained models, such as Segment-Anything Model (SAM, Kirillov et al. (2023)) considered in this work.

Therefore, we propose an alternative: directly regularizing the canonicalization network to produce canonical forms consistent with the (pre)training data, aligning with the prediction network.

3.2.2 Prior regularization

In Section 3.2, we discussed how the canonicalization networks can deviate canonical orientations away from those present in the pretraining datasets. To address this issue, we introduce canonicalization prior, a regularizer that will encourage the canonicalization

network, along with the task-specific loss, to align inputs in an orientation favourable for the prediction network. This approach is motivated by the observation that the orientations of inputs in the fine-tuning dataset, such as images or point clouds, hold valuable information. These orientations are presumed to be analogous to those in the pretraining dataset. Consequently, we assume that the canonicalization process should align inputs to mirror the distribution of orientations within the fine-tuning dataset as accurately as possible.

Deriving the regularizer. The task of the canonicalization network is to map each input data point to a distribution over the group $\mathcal{G}$ of transformations. We introduce two important distributions below:

1. $\mathbb{P}_{c(x)}$ is the distribution induced by the canonicalization function $c(.)$ over $\mathcal{G}$ for a given input data point x, i.e., the predicted distribution over transformations by canonicalization function.

2. $\mathbb{P}_{\mathcal{D}}$ is the canonicalization prior which is a distribution over $\mathcal{G}$ associated with a dataset $\mathcal{D}$, i.e., the distribution over transformations present in the dataset $\mathcal{D}$. More generally, the canonicalization prior can be defined as $\mathbb{P}_{\mathcal{D}}(x)$, i.e., the canonical orientation depends on input x. However, for pretraining datasets, we assume the same prior for all x.

We enforce the prior regularization to minimize the Kullback-Leibler (KL) divergence between $\mathbb{P}_{\mathcal{D}}$ and $\mathbb{P}_{c(x)}$ over the entire dataset $\mathcal{D}$ that is

$$\mathcal{L}_{\text{prior}} = \mathbb{E}_{x\sim\mathcal{D}}[D_{KL}(\mathbb{P}_{\mathcal{D}} \parallel \mathbb{P}_{c(x)})]$$

To simplify further, define the prior to follow a probability density function $q(\mathbf{R})$, where $\mathbf{R}$ is selected as a placeholder, such as rotations. Then, the prediction from canonicalization function c can be defined as $\mathbb{P}_{c(x)} = p(\mathbf{R}|c(x))$. Since the prior distribution is kept

constant, minimizing the KL divergence is equivalent to minimizing the cross-entropy, and $\mathcal{L}_{prior}$ simplifies to:

$$\mathcal{L}_{\text{prior}} = -\mathbb{E}_{x\sim\mathcal{D}}\, \mathbb{E}_{R\sim q(\mathbf{R})}[\log p(\mathbf{R}|c(x))] \tag{3.2}$$

We separately derive the prior regularization for the discrete and continuous rotation groups.

Discrete Rotations

The group of 2D discrete rotations (cyclic group C_n) can be seen as a discrete approximation of its continuous rotation group counterpart $SO(2)$. In this case, we consider the canonicalization prior to be a categorical distribution over group elements, with the prior distribution having a probability mass of 1 for the identity element. Then

$$\mathbb{P}_{\mathcal{D}}(\mathbf{R}) = \delta_{\mathbf{R},\mathbf{I}}$$

, where $\delta_{\mathbf{R},\mathbf{I}}$ is the Kronecker delta function and Equation 3.2 becomes

$$\mathcal{L}_{prior} = -\mathbb{E}_{x\sim\mathcal{D}} \log p(\mathbf{I}|c(x))$$

.

In other words, the regularization loss is simply the negative logarithm of the probability assigned by the canonicalization function to the identity element **I** of the group. Details on practical implementations can be found in Section 4.1.1.

Continuous rotations

We use the matrix Fisher distribution (Downs, 1972) in the case of canonicalization with continuous rotations. It is the analogue of the Gaussian distribution on the $SO(n)$ mani-

fold and is defined as

$$p(\mathbf{R}|\mathbf{F}) = \frac{1}{n(\mathbf{F})}\exp(\mathbf{Tr}[\mathbf{F}^T\mathbf{R}]) \tag{3.3}$$

where $\mathbf{F} \in \mathbb{R}^{n\times n}$ is the parameter of the distribution and $n(\mathbf{F})$ is a normalization constant. Interpretation of the parameter $\mathbf{F}$ and useful properties of the distribution are provided in (Khamsi and Kirk, 2011; Lee, 2018; Mohlin et al., 2020). In particular, considering the proper singular value decomposition $\mathbf{F} = \mathbf{USV}^T$, we find that $\hat{\mathbf{R}} \equiv \mathbf{UV}^T$ is the mode of the distribution and the singular values $\mathbf{S}$ can be interpreted as concentration parameters in the different axes. We therefore set $\mathbf{S} = s\mathbf{I}$ to obtain the isotropic version of the distribution,

$$p(\mathbf{R} \mid \hat{\mathbf{R}}, s) = \frac{1}{n(s)}\exp(s\mathbf{Tr}[\hat{\mathbf{R}}^T\mathbf{R}]) \tag{3.4}$$

where the normalization constant only depends on s (Theorem 2.1 of Lee (2018)). Note that this becomes the Von-Mises distribution, as expected, on $SO(2)$.

Proposition 1. *Let p and q be matrix Fisher distributions of $\boldsymbol{R}$*

$$p(\boldsymbol{R} \mid \hat{\boldsymbol{R}}, s_p) = \frac{1}{n(s_p)}exp(s_p\boldsymbol{Tr}[\hat{\boldsymbol{R}}_p^T\boldsymbol{R}]), \qquad q(\boldsymbol{R} \mid \hat{\boldsymbol{R}}, s_q) = \frac{1}{n(s_q)}exp(s_q\boldsymbol{Tr}[\hat{\boldsymbol{R}}_q^T\boldsymbol{R}]).$$

The cross-entropy is given by

$$\mathbb{E}_{\boldsymbol{R}\sim q}\left[\log p(\boldsymbol{R} \mid \hat{\boldsymbol{R}}_p, s_p)\right] = N(s_q)s_p\boldsymbol{Tr}(\hat{\boldsymbol{R}}_p^T\hat{\boldsymbol{R}}_q) + \log c(s_p) \tag{3.5}$$

Proof. The cross-entropy is given by

$$\mathbb{E}_{\mathbf{R}\sim q}\left[\log p(\mathbf{R} \mid \hat{\mathbf{R}}_p, s_p)\right] = \int_{SO(n)} q(\mathbf{R} \mid \hat{\mathbf{R}}_q, s_q)\log p(\mathbf{R} \mid \hat{\mathbf{R}}_p, s_p)d\mathbf{R}, \tag{3.6}$$

where $d\mathbf{R}$ is the invariant Haar measure on $SO(n)$. Here, we assume that it is scaled such that $\int_{SO(n)} d\mathbf{R} = 1$.

We obtain

$$\mathbb{E}_{\mathbf{R}\sim q}\left[\log p(\mathbf{R} \mid \hat{\mathbf{R}}_p, s_p)\right] = \int_{SO(n)} q(\mathbf{R} \mid \hat{\mathbf{R}}_q, s_q)(s_p \mathbf{Tr}[\hat{\mathbf{R}}_p^T \mathbf{R}] - \log c(s_p))d\mathbf{R}, \tag{3.7}$$

$$\mathbb{E}_{\mathbf{R}\sim q}\left[\log p(\mathbf{R} \mid \hat{\mathbf{R}}_p, s_p)\right] = s_p \mathbf{Tr}(\hat{\mathbf{R}}_p^T \, \mathbb{E}_{\mathbf{R}\sim q}[\mathbf{R}]) - \log c(s_p). \tag{3.8}$$

From Theorem 2.2 and Lemma 2.2 of Lee (2018), we have

$$\mathbb{E}_{\mathbf{R}\sim q}[\mathbf{R}] = \frac{d \log c(s_q)}{d s_q} \hat{\mathbf{R}}_q. \tag{3.9}$$

Therefore, we find

$$\mathbb{E}_{\mathbf{R}\sim q}\left[\log p(\mathbf{R} \mid \hat{\mathbf{R}}_p, s_p)\right] = \frac{d \log c(s_q)}{d s_q} s_p \hat{\mathbf{R}}_q - \log c(s_p), \tag{3.10}$$

which completes the proof.

□

Setting the location parameters of the estimated and prior distributions as $\mathbf{R}_{c(x)}$ and $\hat{\mathbf{R}}_q = \mathbf{I}$ respectively, we find that the canonicalization prior Equation 3.2 is given by

$$\mathcal{L}_{prior} = -\lambda \mathbf{Tr}(\mathbf{R}_{c(x)}) = \frac{\lambda}{2} ||\mathbf{R}_{c(x)} - \mathbf{I}||_F \tag{3.11}$$

where we have eliminated terms that do not depend on $\mathbf{R}_{c(x)}$ and $\lambda = N(s_q)s_p$. Following intuition, the strength of the regularization is determined by the concentrations of the distributions around their mode. Details on practical implementations can be found in Section 4.1.2.

3.2.3 Training and inference with prior regularization

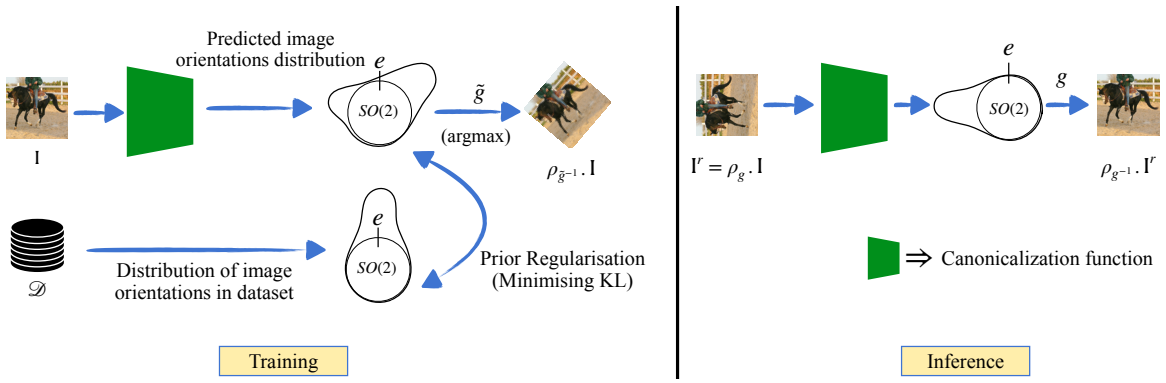


Figure 3.3 Training and inference with our proposed prior regularized canonicalization method. The canonicalization function outputs a distribution over image orientations, and a group element is sampled from this distribution to canonicalize the input image. Additionally, this predicted distribution is regularized during training to match the orientations seen by the large pretrained model in the pretraining dataset.

Training The pipeline consists of two networks - a canonicalization network $c(.)$ followed by a prediction network $\Phi(.)$, which is a large pretrained model. The overall pipeline is represented in Equation 2.3. Depending on the task, we must modify the prediction network output, i.e., for an invariant task, the output of $\Phi(.)$ is the final output, while for an equivariant task, the outputs of $\Phi(.)$ must be transformed to correspond to input orientations. There are two ways to train this pipeline:

- simultaneously training the canonicalization network with prior-regularization loss $\mathcal{L}_{prior}$ and fine-tuning the prediction network with a task-specific loss $\mathcal{L}_{task}$, i.e., $\mathcal{L}_{total} = \mathcal{L}_{task} + \beta.\mathcal{L}_{prior}$. The canonicalization network receives signal from both $\mathcal{L}_{prior}$ and $\mathcal{L}_{task}$.

- only training the canonicalization network with prior regularization loss $\mathcal{L}_{prior}$, i.e., $\mathcal{L}_{total} = \mathcal{L}_{prior}$. This removes the requirement to train the canonicalization network

separately for each prediction network since a trained $c(.)$ can be placed before any prediction network.

Inference Inference involves the canonicalization network predicting a distribution over transformations followed by sampling a transformation to canonicalize the input. The canonicalized input is passed through the prediction network, and the output is transformed depending on the type of task. The training and inference of the canonicalization network is shown in Figure 3.3. The overall canonicalization pipeline with equivariant canonicalizer is presented in Figure 3.4.

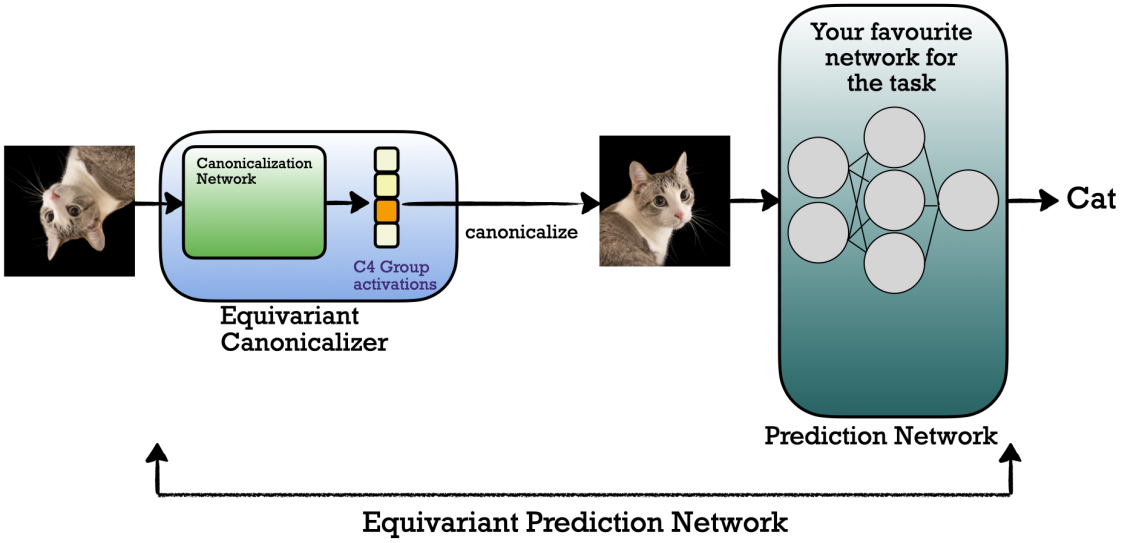


Figure 3.4 Canonicalization with an equivariant canonicalization network. We use an equivariant network to predict a distribution over apriori-defined transformations to canonicalize the input.

3.2.4 Expressivity of equivariant canonicalization networks

In our formulation with prior regularization, during training, the canonicalization network has to predict a distribution similar to the prior distribution. Particularly, as shown

above in Section 3.2.2, it should place a significantly higher probability mass on group identity e.

This capacity to map inputs to group transformations as per a defined prior relates to the expressivity of the canonicalization network. Since explicitly designed equivariant networks are constrained, we require deep equivariant networks to obtain expressive models for learning this mapping. However, this comes at the cost of training and inference speeds. We observe and report this in the case of instance segmentation task with the Microsoft COCO dataset (Lin et al., 2014) in Table 4.5 and Table 4.6. This motivates us to replace equivariant networks with more expressive and faster non-equivariant networks and leverage the optimization approach described in Section 3.1.2. We design losses to learn a desired energy function and propose EQUIOPTADAPT where the canonicalization network can also be non-equivariant.

3.3 EquiOptAdapt

3.3.1 Contrastive loss

We extend the optimization approach to enable the use of any neural network for canonicalization. The optimization formula for a discrete group of transformations, denoted by $\mathcal{G}$, is (Section 3.1.2):

$$g \in \text{argmin}_{g \in \mathcal{G}} E(\rho(g)^{-1}x) \tag{3.12}$$

Assuming there are no symmetric elements in the orbit represented by $\{\rho(g)^{-1}x | g \in \mathcal{G}\}$, it's essential to ensure the function $E(.)$ has a unique minimum to establish a canonical orientation. Additionally, should symmetric elements exist within the orbit, and if the minimum is attained among these symmetric positions, selecting one of them will yield the correct orientation (Kaba et al., 2023; Kaba and Ravanbakhsh, 2023).

To design this function $E(.)$ and learn unique representations for each element in the orbit, we learn it using a neural network and minimize the similarity among the output of the elements in the orbit. We output vectors corresponding to every element in orbit using a standard neural network $s_\theta(.)$, which allows us to use techniques from self-supervised learning (SSL) literature to prevent representation collapse Oord et al. (2018); Wang and Isola (2020); Chen et al. (2020b); Balestriero et al. (2023) including non-contrastive approaches that rely on maximizing the eigenspectrum of the covariance matrix (Zbontar et al., 2021; Bardes et al., 2021).

Since outputting scalars directly from the neural network can make optimization difficult, we limit ourselves to contrastive learning techniques. We define the function $E(.)$ is obtained by taking a dot product of outputs of $s_\theta(.)$ with a reference vector v_R, which we can either learn or keep fixed. We get the distribution induced by the canonicalization function $\mathbb{P}_{c(x)}$ by taking a softmax over $\{v_R \cdot s_\theta(\rho(g)^{-1}x)/\tau | g \in \mathcal{G}\}$, where τ is a temperature

parameter to control the sharpness of the distribution. The final optimization formulation becomes:

$$g \in \text{argmin}_{g \in \mathcal{G}} - \frac{\exp(v_R \cdot s_\theta(\rho(g)^{-1}x)/\tau)}{\sum_{g' \in \mathcal{G}} \exp(v_R \cdot s_\theta(\rho(g')^{-1}x)/\tau)} \tag{3.13}$$

Now, we train $s_\theta(.)$ to output different vectors for every unique element in the orbit using the following loss:

$$\mathcal{L}_{\text{Opt}} = \mathbb{E}_{x \in \mathcal{D}}[\sum_{g_i, g_j \in \mathcal{G}, g_i \neq g_j} s_\theta(\rho(g_i)^{-1}x) \cdot s_\theta(\rho(g_j)^{-1}x)] \tag{3.14}$$

where $\mathcal{D}$ is the training dataset. This loss prevents the collapse of learnt vectors in the output space of $s_\theta(.)$ for different transformations of x by minimizing their similarity. In the context of training from scratch (Kaba et al., 2023), this loss can be jointly optimized with the task loss. Similarly, for fine-tuning or zero-shot adaptation (Mondal et al., 2023), an additional prior regularization loss is used, which is given by:

$$\mathcal{L}_{prior} = \mathbb{E}_{x \in \mathcal{D}_f}[-\log\Big(\frac{\exp(v_R \cdot s_\theta(\rho(g)^{-1}x)/\tau)}{\sum_{g \in \mathcal{G}} \exp(v_R \cdot s_\theta(\rho(g)^{-1}x)/\tau)}\Big)] \tag{3.15}$$

where $\mathcal{D}_f$ is the finetuning dataset, and the identity transformation is assumed to be the prior for natural image dataset (Mondal et al., 2023). Figure 3.5 shows a schematic of our approach.

Typically, we choose $s_\theta()$ that are smaller and faster than the large prediction network Φ. Therefore, requiring $|\mathcal{G}|$ forward passes in parallel through $s_\theta()$ instead of the prediction function Φ, makes our method significantly more efficient than symmetrization based methods (Basu et al., 2023a,b). However, since we don't use equivariance as an

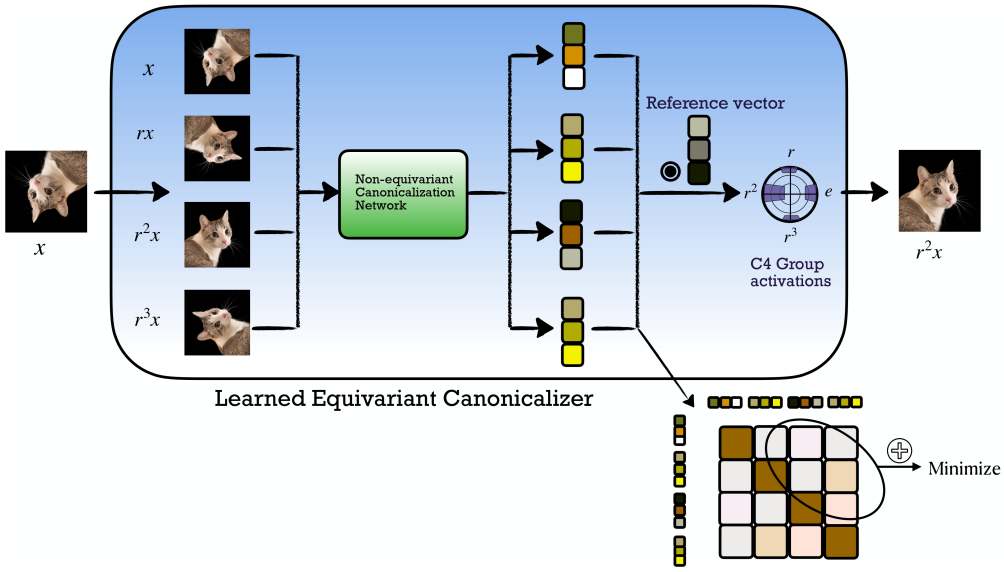


Figure 3.5 Learning equivariant canonicalizer with a non-equivariant canonicalization network. All the group transformations are applied to the input image and passed through the canonicalization network in parallel. A dot product of the output of the canonicalization network with a reference vector gives us a distribution over the transformations to canonicalize the input. We also minimize the similarity between the vectors to get a unique canonical orientation.

inductive bias explicitly in the canonicalization function, this framework aims to achieve approximate instead of exact equivariance.

3.3.2 Pretrained models as canonicalization networks

As our formulation in Section 3.3.1 transfers the equivariance constraint of Equation 3.1 to finding the minima of Equation 3.14, we can conveniently initialize the $s_\theta(.)$ with a pretrained network to ease the optimization process further. In our experiments, we initialize the non-equivariant canonicalization network with popular pretrained ResNet (He et al., 2016) and WideResNet (Zagoruyko and Komodakis, 2016) [1] weights. Although these pre-

[1] https://pytorch.org/vision/main/models/wide_resnet.html

trained models have a higher number of parameters than their equivariant counterparts, they are significantly faster and have higher expressivity. We perform transfer learning by replacing the final layer with a new trainable, fully connected layer with the output size of reference vector $|v_R|$.

3.4 Evaluation Setup

Our evaluation setup attempts to investigate the performance (e.g., accuracy, mean Average Precision or mAP) on the original test set and a transformed test set to measure the robustness and equivariance of the models. If the models are equivariant, then the difference between the performance on the original and transformed test sets should be minimal, ideally zero. In the case of image inputs, this difference can be non-zero due to the appearance of rotation artifacts, particularly in the image corners when the images are rotated. More details on task-specific evaluation are provided below, Section 3.4.1 for image classification, which is an invariant task and Section 3.4.2 for instance segmentation task, which is an equivariant task.

Prior to evaluating the metrics, we analyze the capabilities of the canonicalization framework with and without our novel prior regularization and demonstrate the effectiveness of regularization loss in enabling the canonicalization network to map the inputs to the defined prior distribution. Further, we observe a direct correlation between the performance of the prediction network on the original test set and the capability of the canonicalization network to predict distribution similar to the prior distribution. This can be explained intuitively; original test sets contain image orientations that follow the prior distribution. Thus, for the canonicalization framework to retain the performance of the large pretrained model, it has to map the images to the prior distribution.

3.4.1 Invariant task: image classification

For image classification, we rely on the accuracy. Thus, the metric for the original test set is termed as *Accuracy*. We use cyclic group $\mathcal{G} = C_8$ (8 discrete rotations) for obtaining the transformed test set for EquiAdapt while cyclic group $\mathcal{G} = C_4$ (4 discrete rotations) for EquiOptAdapt to avoid rotation artifacts while training non-equivariant canonicalization networks. We categorize the metric as $\mathcal{G}$-*Average Accuracy*, where $\mathcal{G} = C_4, C_8$. We observe that the explicitly designed equivariant networks are comparatively more robust to rotation artifacts than non-equivariant networks trained with an optimization approach. Note that transformations with group elements of C_4, i.e., multiples of $90°$, do not introduce rotation artifacts. We evaluate *Accuracy* and Cn-*Average Accuracy* for popular networks pretrained on ImageNet-1K dataset (Deng et al., 2009) such as ResNet50 (He et al., 2016) and ViT-Base (Dosovitskiy et al., 2020) on datasets used widely to test transfer learning properties, such as CIFAR10, CIFAR100 (Krizhevsky et al., 2009), and STL10 dataset (Coates et al., 2011).

3.4.2 Equivariant task: instance segmentation

We report mean zero-shot Average Precision or mAP for the original *val* set of Microsoft COCO 2017 (Lin et al., 2014) for Mask-RCNN (He et al., 2017) and Segment-Anything Model (SAM, Kirillov et al. (2023)), which are pretrained on Microsoft COCO (Lin et al., 2014) and SA-1B Kirillov et al. (2023) datasets respectively. We perform instance segmentation, i.e., predicting the correct mask for instances present in the image for Mask-RCNN. These classes are already defined in the Microsoft COCO. However, since the Segment-Anything Model can segment all possible objects in an image, we resort to *prompted* instance segmentation - we provide the ground truth bounding box in the images around the objects defined in Microsoft COCO. For the transformed *val* set, we perform augmentations with C_4 to maintain the structure of bounding boxes and masks while avoiding rotation artifacts. This metric is termed as $C4-$ *Average mAP*.

Chapter 4

Experimental Results and Discussion

4.1 Implementation of Canonicalization Networks

Below, we describe the architectural details of equivariant canonicalization networks used for discrete C_n and continuous $SO(2)$ rotation groups. The design of a non-equivariant canonicalization network is straightforward and described in Section 3.3.2. We extensively use `escnn` library (Weiler and Cesa, 2019; Cesa et al., 2022) to build equivariant networks.

4.1.1 Discrete rotation group

The canonicalization network needs to output logits corresponding to every group element in the discrete cyclic group C_n. This can be achieved by using a G-CNN (Cohen and Welling, 2016a) or an $E(2)$-Steerable Network (Weiler and Cesa, 2019) that produces outputs using *regular* representation. Each layer of the C_n-equivariant convolutional network consists of convolution with regular representation except the first layer, which maps the trivial representation of the C_n group to its regular representation. We use equivariant implementation of batch normalization, ReLU activation function, and drop out as proposed in Weiler and Cesa (2019); Cesa et al. (2022). Major hyperparameters tuned include

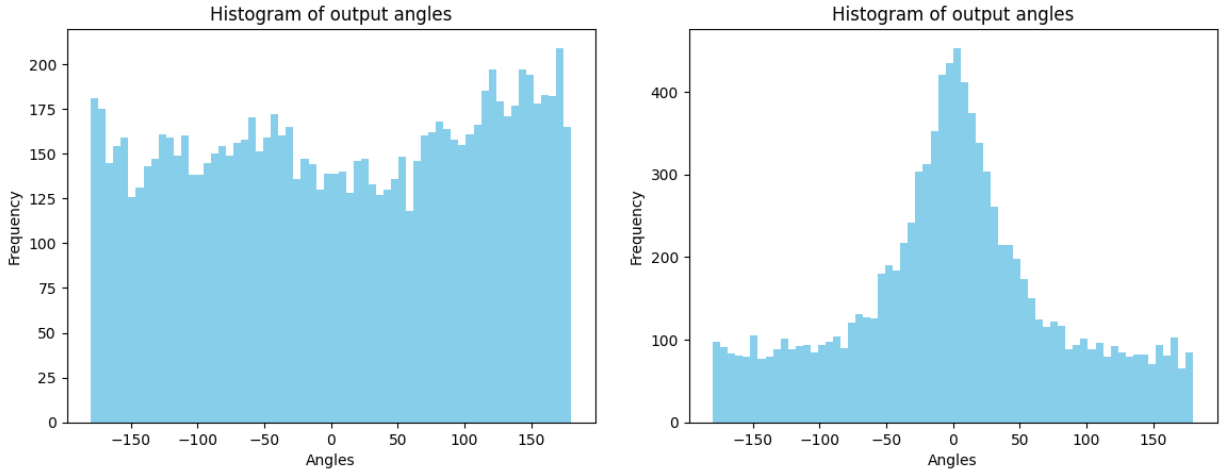
the number of layers, kernel sizes, dropout, and learning rates. We use $n = 4, 8$ when we evaluate $C4$- and $C8-$*Average Accuracy* respectively.

C_4-equivariant WideResNet, used for comparing the effect of canonicalization network expressivity in instance segmentation task in Table 4.5, includes repetitive stacking of equivariant versions of *basic* residual blocks on several consecutive *bottleneck* residual blocks (for details on these residual blocks, we refer readers to Figure 1 in Zagoruyko and Komodakis (2016)). The rest of the architecture details and hyperparameters are identical to the design of the C_8-equivariant convolutional network above.

We follow the overall implementation proposed in Kaba et al. (2023) to design the canonicalization function. We take a spatial average and get logits corresponding to every element in the group along the fibre dimension. Now, we get a discrete distribution over the group elements by taking a softmax and minimizing the $\mathcal{L}_{total}$ objective. During training, we utilize the argmax operation instead of sampling from this distribution using Gumbel Softmax (Jang et al., 2016) and employ the straight-through gradient trick (Bengio et al., 2013). All our image-based experiments use this discrete rotation group canonicalization function.

4.1.2 Continuous rotation group

In this case, the canonicalization network needs to output rotation matrices $\mathbf{R}_{c(x)} \in SO(2)$ that equivariantly transforms with the input (image). This can be achieved using a $E(2)$-Steerable Network (Weiler and Cesa, 2019) that outputs two vector fields. To design the canonicalization function, we can take a spatial average over both vector fields and Gram-Schmidt orthonormalize the vectors to get a 2D rotation matrix. While this sounds promising in theory, in practice, we found it empirically challenging to optimize using the loss to enforce canonicalization prior in Equation 3.11 for images.



Distribution of angles at the start of training

Distribution of angles after training

Figure 4.1 Distribution of angles output from steerable canonicalization function in $SO(2)$ with prior regularization (Equation 3.11) for CIFAR10 (Krizhevsky et al., 2009) before and after training. x-axis denotes angles from $-180°$ to $+180°$. Frequency denotes the number of images mapped to a particular angle.

Optimization challenges. In Figure 4.1, we present the distributions of predicted angles ranging from -180 to $+180$. Our analysis shows that the mean and standard deviation of the predicted angles on the test set are -0.54 and 80.23, respectively. This observation signifies that while the prior guides the canonicalization function to output angles close to 0 (identity element of $SO(2)$), there exist instances where angles significantly deviate from this central value. We leave this investigation as future work and expect to reduce the standard deviation with additional regularization during training and more expressive networks.

4.2 Augmentation and Alignment Effects

To understand the augmentation and alignment effect with the canonicalization network (Section 3.2.1), we train from scratch and fine-tune[1] a ResNet50 (He et al., 2016) in the canonicalization framework in different augmentation setups - no rotation augmentation, small rotation angles, large rotation angles, and learned canonicalization. Note that this effect can be observed with both equivariant and non-equivariant canonicalization networks. However, we follow the learned canonicalization framework in Kaba et al. (2023) and investigate the issue with an equivariant network. As explained before, learned canonicalization also provides an augmentation effect when initially the canonicalization network is initialized randomly. We report Accuracy and $C8-$Average Accuracy on CIFAR10 and CIFAR100 (Krizhevsky et al., 2009).

Table 4.1 Augmentation and alignment effect on the prediction network. Top-1 classification accuracy and $\mathcal{G}$-Averaged classification accuracy for CIFAR10 and CIFAR100 (Krizhevsky et al., 2009). $C8-$Avg Acc refers to the top-1 accuracy on the augmented test set obtained using the group $\mathcal{G} = C_8$, with each element of $\mathcal{G}$ applied on the original test set.

Dataset →		CIFAR10		CIFAR100	
Prediction Network ↓	Rotation Augmentation	Acc	$C8$-Avg Acc	Acc	$C8$-Avg Acc
ResNet50 (He et al., 2016)	None	**91.64 ± 0.22**	43.82 ± 0.75	**77.57 ± 0.37**	38.20 ± 0.24
	-10 to $+10$ degrees	90.96 ± 0.41	44.87 ± 0.60	74.83 ± 0.15	37.14 ± 0.42
	-180 to $+180$ degrees	84.60 ± 1.83	81.04 ± 1.86	61.07 ± 0.27	59.42 ± 0.70
	Learned Canonicalization (LC) (Kaba et al., 2023)	83.11 ± 0.35	**82.89 ± 0.41**	59.84 ± 0.67	**59.45 ± 0.49**
ResNet50 (pretrained on ImageNet)	None	**97.44 ± 0.03**	57.47 ± 0.14	85.11 ± 0.06	44.34 ± 0.09
	-10 to $+10$ degrees	96.97 ± 0.01	57.77 ± 0.25	**85.84 ± 0.10**	44.86 ± 0.12
	-180 to $+180$ degrees	94.91 ± 0.07	90.11 ± 0.19	80.21 ± 0.09	74.12 ± 0.05
	Learned Canonicalization (LC) (Kaba et al., 2023)	93.29 ± 0.01	**92.96 ± 0.09**	78.50 ± 0.15	**77.52 ± 0.07**

We observe that training with no or small rotation angles leads to the best downstream performance on the test set but reduces the robustness, as we observe in $C8-$Average Accuracy for both datasets. However, training with large rotation angles improves the performance on the augmented test set but significantly affects the results on the original

[1] pretrained on ImageNet-1K(Deng et al., 2009)

test set and does not guarantee equivariance. Finally, training with a learnable canonicalization network guarantees equivariance but reduces the original test set performance due to the augmentation effect. This observation is identical for training from scratch as well as fine-tuning. To obtain the best of both worlds, we introduce the prior regularization in the canonicalization framework to improve alignment with small augmentations while guaranteeing equivariance.

4.3 Image Classification

In this section, we first describe the considered baselines in Section 4.3.1 followed by the effect of our proposed prior regularization in learning the correct orientations in the fine-tuning dataset in Section 4.3.2. We detail the improvements in results with EquiAdapt over other baselines in Section 4.3.3 and further the benefits of EquiOptAdapt in Section 4.3.4. As mentioned in Section 3.4.1, we use C_8 and C_4 groups for evaluating EquiAdapt and EquiOptAdapt, respectively.

4.3.1 Baselines

We compare different fine-tuning setups for the invariant image classification task in CIFAR10, CIFAR100 (Krizhevsky et al., 2009), and STL10 (Coates et al., 2011) datasets:

1. **Vanilla**: The *Vanilla* model refers to fine-tuning the pretrained checkpoints using data augmentations such as horizontal flips and small angle rotations.

2. **Rotation Augmentation**: The *Rotation Augmentation* is identical to *Vanilla* setup and additionally performs augmentation with angles ranging from -180° to $+180^\circ$ instead of small rotation angles.

3. **Learned Canonicalization (LC)**: This is identical to direct approach of Kaba et al. (2023) described in Section 3.1.1.

4. **C8-Augmentation**: This is a strong baseline where we perform augmentation only with group $\mathcal{G} = C_8$ since we evaluate with $C8-$Average Accuracy. While this is a competitive baseline, it does not guarantee equivariance and is expensive for larger augmented test sets with C_n where n is larger.

5. **EquiAdapt**: Our proposed prior-regularization with canonicalization framework (Section 3.2). Note, we set the beta value, the strength of prior loss $\mathcal{L}_{prior}$ to 100.

6. **EquiOptAdapt**: Our proposed prior-regularization with non-equivariant canonicalization network (Section 3.3). Note, we use the size of the reference vector $|v_R| = 128$.

4.3.2 Learning Prior Distribution

We present a comprehensive analysis of the output distribution over eight discrete angles predicted by the canonicalization function, both before and after training, on the test set. These findings are depicted in Figure 4.2a for *LC*, and Figure 4.2b for *EquiAdapt*. Here, the numbers 0 to 7 correspond to angles that are multiples of $45°$, ranging from $0°$ to $315°$, respectively.

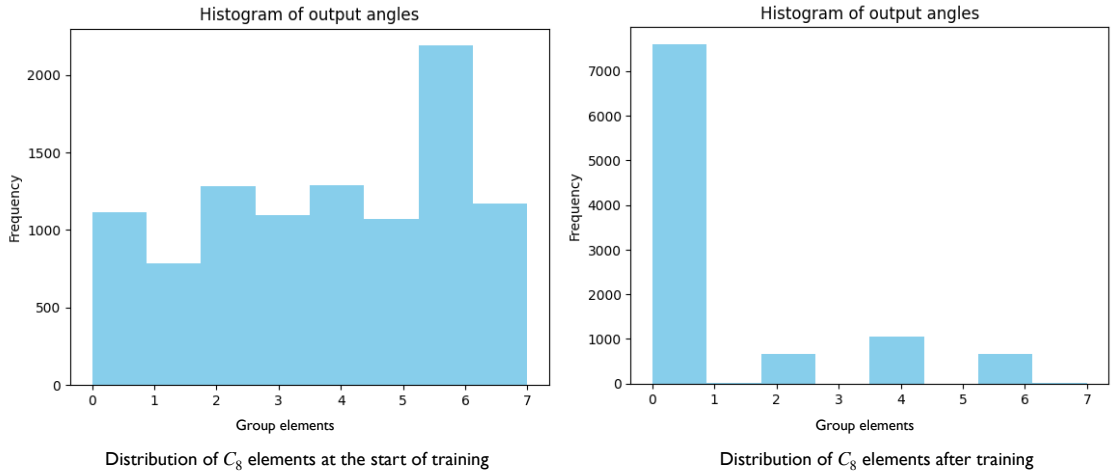
We demonstrate that incorporating prior regularization into the canonicalization function aids in mapping the images to the identity prior (represented by 0). This improvement positively impacts the accuracy of the original test set, as evidenced by the results in Table 4.3. Conversely, relying solely on the classification task loss yields no significant alteration in the angle distribution, as the post-training test set distributions remain random.

Additionally, we provide valuable insights into the fraction of images mapped to the identity element in Table 4.2. It is important to note that the expressivity of the canonicalization function, specifically employing lightweight equivariant networks, contributes to the inability to map all images to the identity elements. This observation calls for further exploration in understanding the role of expressivity and generalization within canoni-

Distribution of C_8 elements at the start of training

Distribution of C_8 elements after training

(a) Learned Canonicalization (LC)



Distribution of C_8 elements at the start of training

Distribution of C_8 elements after training

(b) EquiAdapt (ours)

Figure 4.2 Distribution of angles output from canonicalization function in $C8$ for a considered canonicalization framework for CIFAR10 before and after training. We use indices on the x-axis instead of angle values to represent the corresponding multiple of 45°. Frequency denotes the number of images mapped to a particular multiple of 45°. The histograms show that EquiAdapt is able to map most of the elements to the desired canonical orientation while Learned Canonicalization fails to do so.

calization networks with known prior orientations.

Table 4.2 Fraction of images mapped to identity group element.

		Training Completed	
Dataset ↓	Model	✗	✓
CIFAR10 (Krizhevsky et al., 2009)	Learned Canonicalization (LC)	0.11	0.23
	EquiAdapt (ours)	0.11	0.76

4.3.3 EquiAdapt results

Table 4.3 Performance comparison of large pretrained models fine-tuned on different vision datasets. Both classification accuracy and $\mathcal{G}$-averaged classification accuracies are reported. Acc refers to the accuracy on the original test set, and $C8$-Avg Acc refers to the accuracy on the augmented test set obtained using the group $\mathcal{G} = C_8$.

Pretrained Large Prediction Network →		ResNet50		ViT	
Datasets ↓	Model	Acc	$C8$-Avg Acc	Acc	$C8$-Avg Acc
CIFAR10 (Krizhevsky et al., 2009)	Vanilla	**97.33 ± 0.01**	57.77 ± 0.25	**98.13 ± 0.04**	63.59 ± 0.48
	Rotation Augmentation	94.91 ± 0.07	90.11 ± 0.19	96.26 ± 0.15	93.67 ± 0.39
	Learned Canonicalization (LC)	93.29 ± 0.01	92.96 ± 0.09	95.00 ± 0.01	94.80 ± 0.02
	$C8$-Aug.	95.76 ± 0.07	94.36 ± 0.09	96.36 ± 0.02	94.17 ± 0.08
	EquiAdapt (ours)	96.19 ± 0.01	**95.31 ± 0.17**	96.14 ± 0.14	**95.08 ± 0.10**
CIFAR100 (Krizhevsky et al., 2009)	Vanilla	**85.84 ± 0.10**	44.86 ± 0.12	**87.91 ± 0.28**	55.87 ± 0.14
	Rotation Augmentation	80.21 ± 0.09	74.12 ± 0.05	82.59 ± 0.44	78.39 ± 0.89
	Learned Canonicalization (LC)	78.50 ± 0.15	77.52 ± 0.07	80.86 ± 0.17	80.48 ± 0.20
	$C8$-Aug.	83.00 ± 0.09	79.72 ± 0.10	83.45 ± 0.09	80.08 ± 0.38
	EquiAdapt (ours)	83.44 ± 0.02	**82.09 ± 0.09**	84.27 ± 0.10	**83.61 ± 0.01**
STL10 (Coates et al., 2011)	Vanilla	**98.30 ± 0.01**	75.68 ± 1.43	**98.31 ± 0.09**	76.66 ± 0.93
	Rotation Augmentation	98.08 ± 0.06	94.97 ± 0.08	97.85 ± 0.17	94.07 ± 0.11
	Learned Canonicalization (LC)	95.30 ± 0.19	93.92 ± 0.10	95.11 ± 0.01	94.67 ± 0.02
	$C8$-Aug.	**98.31 ± 0.01**	96.31 ± 0.13	97.83 ± 0.08	94.45 ± 0.35
	EquiAdapt (ours)	97.01 ± 0.01	**96.37 ± 0.12**	96.15 ± 0.05	**95.73 ± 0.16**

We compare EquiAdapt with other baselines in Table 4.3. As anticipated, we found that large pretrained networks for images are not robust to rotation transformations, as indicated by the significant drop in performance from the accuracy to its $C8$-averaged

counterpart for both ResNet50 and ViT. Nevertheless, we observe that ViT is more robust to higher-order rotations compared to ResNet50, which has also been observed by Gruver et al. (2023a). We notice that augmenting with a full range of rotation angles during training improves the $C8$-Average Accuracy as demonstrated by our *Rotation Augmentation* baseline. However, it hurts the accuracy of the prediction network in the original test set and does not guarantee equivariance. Augmenting with necessary rotations in *C8-Augmentation* does not ensure equivariance to C_8 but retains performance on the original test set and reduces the gap between original and C8-averaged accuracies.

LC guarantees equivariance, which can be seen from the minor difference between the accuracies of the original and augmented test sets. Nevertheless, in every dataset, we can observe a significant drop in accuracy for the original test set. We extensively discussed this issue in Section 3.2.1. However, with *EquiAdapt* method, we can reduce the gap between the accuracy on the original test set while still being equivariant to rotations. This demonstrates that this prior regularization is a promising direction to improve the performance of large-pretrained models while guaranteeing robustness to out-of-distribution samples resulting from transformations like rotation.

Ideally, the original test set's accuracy should be nearly identical for both the *Vanilla* and *EquiAdapt* setup. However, we observed a slight difference between their corresponding accuracies. This disparity arises because the canonicalization model cannot perfectly map all data points (images) to the identity element e as demonstrated in Section 4.3.2.

4.3.4 Comparison between EquiOptAdapt and EquiAdapt

We compare our two proposed methods in Table 4.4. Our findings demonstrate that *EquiOptAdapt*, similar to *EquiAdapt*, also exhibits comparable performance to the *Vanilla* setup in terms of test-set accuracy, with *EquiOptAdapt* showcasing superior performance. This suggests that non-equivariant canonicalizers can be designed to be expressive, thereby

Table 4.4 Performance comparison between EquiOptAdapt and EquiAdapt fine-tuning setups for large pretrained models on different vision datasets. Both Accuracy (Acc) and $C4$-Average Accuracy ($C4$-Avg Acc) are reported. Acc refers to the accuracy on the original test set, and $C4$-Avg Acc refers to the accuracy on the augmented test set obtained using the group C_4.

Pretrained Large Prediction Network →		ResNet50		ViT	
Datasets ↓	Model	Acc	$C4$-Avg Acc	Acc	$C4$-Avg Acc
CIFAR10 (Krizhevsky et al., 2009)	Vanilla	**97.33 ± 0.01**	69.72 ± 0.25	**98.13 ± 0.04**	68.98 ± 0.48
	EquiAdapt	96.19 ± 0.01	96.18 ± 0.02	96.14 ± 0.14	96.12 ± 0.11
	EquiOptAdapt	97.16 ± 0.01	**97.16 ± 0.01**	96.96 ± 0.02	**96.96 ± 0.02**
STL10 (Coates et al., 2011)	Vanilla	**98.30 ± 0.01**	88.61 ± 0.34	**98.31 ± 0.09**	78.63 ± 0.25
	EquiAdapt	97.01 ± 0.01	96.98 ± 0.02	96.15 ± 0.05	96.15 ± 0.05
	EquiOptAdapt	98.04 ± 0.05	**98.04 ± 0.04**	97.32 ± 0.01	**97.32 ± 0.01**

enhancing their ability to learn to predict the prior distribution over the elements of the considered group. Additionally, we observe that more expressive canonicalizers lead to higher performance. Further, no gap exists between accuracy and $C4$-average accuracy, demonstrating the successful equivariant adaptation of the considered models.

4.4 Instance Segmentation

In this section, we first demonstrate the effect of increasing the expressivity of equivariant canonicalization networks in Section 4.4.1. A more expressive canonicalization network achieved better mAP values as a consequence of the enhanced ability to learn the prior and thus map the input images in the original test set to identity group element. Further, we demonstrate the additional benefits of EquiOptAdapt over EquiAdapt in Section 4.4.2.

4.4.1 Expressivity of canonicalization network

We compare the zero-shot results for the equivariant instance segmentation task in COCO 2017 *val* set (Lin et al., 2014). First, we perform an ablation on the expressivity of the

Table 4.5 Zero-shot performance comparison of large pretrained segmentation models with trained equivariant canonicalization functions (EquiAdapt) of different expressivity levels on COCO 2017 dataset (Lin et al., 2014). We report mAP and $C4$-averaged mAP values. † indicates G-CNN and ‡ indicates a more expressive G-WRN for canonicalization.

Pretrained Large Segmentation Network →		MaskRCNN		SAM	
Datasets ↓	Model	mAP	$C4$-Avg mAP	mAP	$C4$-Avg mAP
COCO (Lin et al., 2014)	Zero-shot	**48.19**	29.34	**62.32**	58.77
	EquiAdapt†	35.77	35.77	59.28	59.28
	EquiAdapt‡	46.80	**46.79**	62.10	**62.10**

canonicalization network in *EquiAdapt* setup in Table 4.5. We use a C_4-equivariant G-CNN and a C_4-equivariant WideResNet to record the zero-shot results. Each of these networks was trained only with $\mathcal{L}_{prior}$, which requires the networks to map each image to identity (since our prior distribution is an identity distribution). Thus, a more expressive network will perform better than a less expressive version.

We observe that irrespective of the expressivity of the canonicalization network, our prior regularization method is successful in equivariant adaptation of the prediction network. However, a more expressive network retains the performance of the prediction network on the original *val* set. A few qualitative examples with C_4-equivariant WideResNet as canonicalization network are provided in Figure 4.3. Note that the canonicalization network was trained *independently* and utilized for both the prediction network. This further demonstrates the superiority of our proposed approach. Inspired by these insights, we train a non-equivariant pretrained model as a canonicalization network and achieve excellent results.

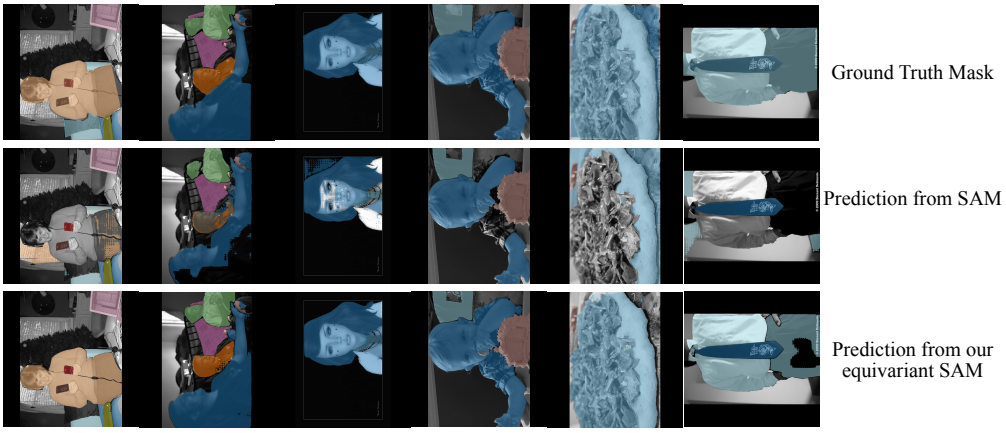


Figure 4.3 Predicted masks from the Segment Anything Model (SAM, (Kirillov et al., 2023)), showcasing both the original model and our proposed EquiAdapt for 90° counter-clockwise rotated input images taken from the COCO 2017 dataset (Lin et al., 2014).

4.4.2 Comparison between EquiOptAdapt and EquiAdapt

Table 4.6 presents the results for various setups. Our analysis reveals that both proposed EquiAdapt and EquiOptAdapt effectively achieve architecture-agnostic equivariant adaptation of large pretrained models while maintaining their mean Average Precision (mAP) performance. Notably, EquiOptAdapt outperforms EquiAdapt in this regard. The inference times for EquiOptAdapt and EquiAdapt indicate that the canonicalization process is $2\times$ faster for EquiOptAdapt.

Additionally, we also compare the relative wall clock time (in minutes) to learn the prior distribution $\mathbb{P}_{c(x)}$ during training between EquiOptAdapt and EquiAdapt in Figure 4.4. Since our chosen $\mathbb{P}_{c(x)}$ is a δ-distribution centred on the identity element e of the group (identity prior), we assess the accuracy of mapping the inputs to the identity element e and refer it as the identity metric. We demonstrate that EquiOptAdapt can learn the prior distribution faster than EquiAdapt. This results from using any existing non-equivariant pretrained WideResNet model that trains and runs faster than its equivariant counterpart

Table 4.6 Zero-shot performance comparison and inference times of large pretrained segmentation models with both non-equivariant (EquiOptAdapt) and equivariant (EquiAdapt) canonicalization functions on the validation set of COCO 2017 dataset (Lin et al., 2014).

Network (→)	**MaskRCNN**		**SAM**		**MaskRCNN**	**SAM**
Setup (↓)	mAP	C4-Avg mAP	mAP	C4-Avg mAP	Inference times (↓)	
Zero-shot	**48.19**	29.34	**62.32**	58.77	23m 53s	2h 28m 43s
EquiAdapt	46.80	46.79	62.10	62.10	27m 09s (+13.68%)	2h 34m 36s (+3.96%)
EquiOptAdapt	48.01	**48.01**	62.30	**62.30**	25m 35s (+7.12%)	2h 30m 42s (+1.33%)

in EquiAdapt. Therefore, our findings suggest that EquiOptAdapt generally offers better performance and faster training and inference times than EquiAdapt.

4.5 Additional Results on Point Cloud Domain

Below, we provide additional results on the point cloud domain - specifically, classification and part segmentation.

Datasets. For our experiments involving point clouds, we utilized the ModelNet40 (Wu et al., 2015) and ShapeNet (Chang et al., 2015) datasets. The ModelNet40 dataset comprises 40 classes of 3D models, totalling 12,311 models. Among these, 9,843 models were allocated for training, while the remaining models were reserved for testing in the classification task. In the case of part segmentation, we employed the ShapeNet-part subset, which encompasses 16 object categories and over 30,000 models. We only train the canonicalization function using the prior loss $\mathcal{L}_{prior}$ in Eq. 3.11.

Evaluation protocol. To ensure consistency and facilitate comparisons, we followed the established conventions set by Esteves et al. (2018a) and adopted by Deng et al. (2021)

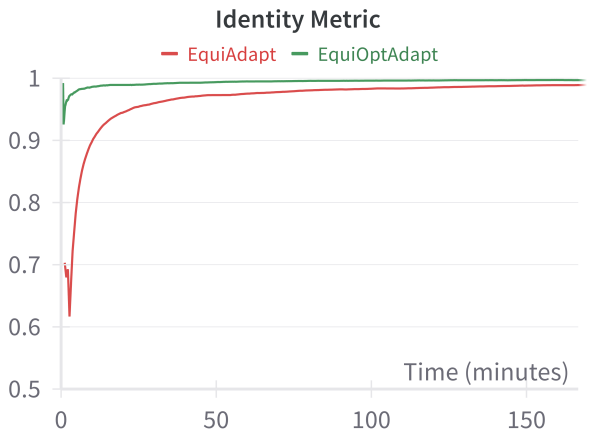


Figure 4.4 Identity metric vs. Relative wall-time (in minutes). We define the identity metric as the percentage of input images mapped to the identity group element e, which is our prior distribution $\mathbb{P}_{c(x)}$. Therefore, it is the accuracy of learning the identity prior. This figure demonstrates that EquiOptAdapt can learn the prior faster than EquiAdapt.

for the train/test rotation setup in the classification and segmentation tasks. The notation x/y indicates that transformation x is applied during training, while transformation y is applied during testing. Typically, three settings are employed: z/z, $z/SO(3)$, and $SO(3)/SO(3)$. Here, z denotes data augmentation with rotations around the z-axis during training, while $SO(3)$ represents arbitrary rotations. However, since we regularize the output of the canonicalization with the identity transformation, we trained our canonicalization function and fine-tuned our pretrained model without any rotation augmentation. During inference, we tested on z and $SO(3)$ augmented test datasets.

Results. We present our results on Table 4.7. Notably, our method showcased superior robustness, outperforming existing methods for point cloud tasks. Specifically, including the prior loss has led to a significant improvement in PointNet's performance compared to DGCNN. This observation aligns with our analysis in Section 3.2.1, where we high-

Table 4.7 Classification accuracy of different point cloud models on the ModelNet40 dataset (Wu et al., 2015) in different train/test scenarios and ShapeNet (Chang et al., 2015) Part segmentation mean IoUs over 16 categories in different train/test scenarios. x/y here stands for training with x augmentation and testing with y augmentation. z here stands for aligned data augmented by random rotations around the vertical/z axis, and $SO(3)$ indicates data augmented by random 3D rotations.

Task →	Classification			Part Segmentation	
Dataset →	ModelNet40			ShapeNet	
Method ↓	z/z	$z/SO(3)$	$SO(3)/SO(3)$	$z/SO(3)$	$SO(3)/SO(3)$
PointNet (Qi et al., 2017)	85.9	19.6	74.7	38.0	62.3
DGCNN (Wang et al., 2019)	**90.3**	33.8	88.6	49.3	78.6
VN-PointNet	77.5	77.5	77.2	72.4	72.8
VN-DGCNN	89.5	89.5	90.2	81.4	81.4
LC-PointNet (Kaba et al., 2023)	79.9 ± 1.3	79.6 ± 1.3	79.7 ± 1.3	73.5 ± 0.8	73.6 ± 1.1
LC-DGCNN (Kaba et al., 2023)	88.7 ± 1.8	88.8 ± 1.9	90.0 ± 1.1	78.4 ± 1.0	78.5 ± 0.9
Ours (with pretrained PointNet and DGCNN for each task)					
	no-aug/z	no-aug/$SO(3)$		no-aug/$SO(3)$	
EquiAdapt-PointNet	84.1 ± 1.1	84.3 ± 1.2		82.6 ± 1.3	
EquiAdapt-DGCNN	**90.2 ± 1.4**	**90.2 ± 1.3**		**84.3 ± 0.8**	

light that training the prediction network with large rotations can hinder its performance and serve as a bottleneck for equivariance within the learnt canonicalization framework. The empirical evidence, particularly in the $SO(3)/SO(3)$ results of vanilla PointNet and DGCNN, where we notice a more pronounced drop in PointNet's performance, supports this and strengthens our findings.

Chapter 5

Conclusion

5.1 Summary

In this thesis, we delved into the architecture-agnostic equivariant adaptation of large pretrained models, tackling the significant challenge of ensuring robust performance under various transformations. Our investigation revealed that directly applying the canonicalization framework to this problem results in reduced downstream task performance for the models under consideration. To address this issue, we highlighted the necessity of introducing a novel prior regularization to refine the canonicalization approach for large pretrained networks.

In response, we proposed two frameworks - EquiAdapt and EquiOptAdapt, featuring equivariant and non-equivariant canonicalization networks, respectively. These frameworks were extensively evaluated against multiple baselines, particularly tasks in computer vision and point cloud processing. Our results demonstrated their effectiveness in achieving equivariant versions of existing large pretrained and foundation models without compromising their inference speed and performances. To encourage the adoption of our findings, we have made available an open-source Python package, `equiadapt`[1].

[1] `https://github.com/arnab39/equiadapt`

This package provides a user-friendly, plug-and-play solution for transforming existing networks into equivariant networks, thereby facilitating the wider use and integration of our proposed frameworks.

5.2 Key Findings

This thesis tackles the critical challenge of ensuring robustness in large pretrained models by leveraging equivariance with respect to known transformation groups through the canonicalization framework. The most important takeaways are listed below:

- We identified that the canonicalization framework (Kaba et al., 2023) struggles with alignment issues when trained with large pretrained models due to the insufficient signal from using the task loss, leading to a mismatch in the large pretrained network between the orientations encountered during pretraining and training for the downstream task.

- These alignment issues can be addressed with our proposed novel prior regularization technique to align canonical orientation with the orientations present in pretraining datasets. We termed this approach EquiAdapt. Our results indicated that EquiAdapt effectively maintained the performance of pretrained models while enhancing their robustness to transformations.

- Despite its potential, we found that equivariant networks face challenges in effectively predicting the prior distribution of orientations due to their limited expressivity. To overcome this, we leveraged ideas from contrastive learning literature to train highly expressive pretrained non-equivariant networks as canonicalization functions. This approach resulted in the creation of EquiOptAdapt, which not only achieved faster inference times and more efficient learning of the prior distribution but also outperformed EquiAdapt in performance.

5.3 Future Work

There are several interesting directions for our work which can result in improved core ideas of our prior-regularized canonicalization framework and wider applicability:

- A straightforward extension involves applying EquiAdapt and EquiOptAdapt to continuous groups, such as the group of 2D rotations $SO(2)$. For EquiAdapt, this would include addressing the optimization problems with $SO(2)-$Steerable networks (Section 4.1.2), potentially by increasing the expressivity of canonicalization networks. Similarly, for EquiOptAdapt, using continuous group will require test time optimization using the output energy values, which can make inference significantly more expensive.

- Other than continuous rotations, another similar extension involves applying EquiOptAdapt to higher-order discrete rotations. The finer rotation angles present an intriguing challenge for both continuous and higher-order discrete rotations due to the artifacts introduced at the corners of images. To address this, we aim to design novel techniques to make the pretrained non-equivariant canonicalization network robust to the effect of artifacts. Moreover, exploring other non-contrastive correlation-based methods to train the canonicalizer is another direction for future research.

- Automating prior discovery based on the performance of the pretrained model over the different transformations of the input in the finetuning data can, in principle, replace manually specifying the prior distribution. This leads to a more general equivariant adaption technique agnostic to the choice of model and data.

- Finally, future research could extend the canonicalization framework to other transformations and tasks beyond image and point-cloud domains. This includes scientific applications such as materials and molecule generation, broadening the impact

and utility of our findings.

5.4 Outlook

EquiAdapt and EquiOptAdapt offer straightforward, architecture-agnostic methods for designing equivariant architectures in the era of large pretrained and foundation models. We integrate novel prior regularization and non-equivariant networks in a canonicalization framework to provide practical and efficient solutions for enhancing the robustness of AI models. We release `equiadapt` python package for easier integration of our contributions to relevant applications. Finally, we hope our research contributions will have significant implications for designing and deploying robust AI systems, promoting the development of more generalizable and efficient models.

Bibliography

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. 2024. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pages 1–3.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Kenneth Atz, Francesca Grisoni, and Gisbert Schneider. 2021. Geometric deep learning on molecular representations. *Nature Machine Intelligence*, 3(12):1023–1032.

Matan Atzmon, Koki Nagano, Sanja Fidler, Sameh Khamis, and Yaron Lipman. 2022. Frame averaging for equivariant shape space learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 631–641.

Waiss Azizian et al. 2020. Expressive power of invariant and equivariant graph neural networks. In *International Conference on Learning Representations*.

Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, et al. 2023. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*.

Adrien Bardes, Jean Ponce, and Yann LeCun. 2021. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*.

Sourya Basu, Pulkit Katdare, Prasanna Sattigeri, Vijil Chenthamarakshan, Katherine Rose Driggs-Campbell, Payel Das, and Lav R Varshney. 2023a. Efficient equivariant transfer learning from pretrained models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Sourya Basu, Prasanna Sattigeri, Karthikeyan Natesan Ramamurthy, Vijil Chenthamarakshan, Kush R Varshney, Lav R Varshney, and Payel Das. 2023b. Equi-tuning: Group

equivariant fine-tuning of pretrained models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6788–6796.

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. 2022. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):2453.

Erik J Bekkers. 2020. B-spline cnns on lie groups. In *International Conference on Learning Representations*.

Erik J Bekkers, Maxime W Lafarge, Mitko Veta, Koen AJ Eppenhof, Josien PW Pluim, and Remco Duits. 2018. Roto-translation covariant convolutional networks for medical image analysis. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part I*, pages 440–448. Springer.

Erik Johannes Bekkers, Marco Loog, Bart M ter Haar Romeny, and Remco Duits. 2017. Template matching via densities on the roto-translation group. *IEEE transactions on pattern analysis and machine intelligence*, 40(2):452–466.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Benjamin Bloem-Reddy, Yee Whye, et al. 2020. Probabilistic symmetries and invariant neural networks. *Journal of Machine Learning Research*, 21(90):1–61.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J Bekkers, and Max Welling. 2021. Geometric and physical quantities improve e (3) equivariant message passing. In *International Conference on Learning Representations*.

Gabriele Cesa, Leon Lang, and Maurice Weiler. 2022. A program to build e(n)-equivariant steerable CNNs. In *International Conference on Learning Representations*.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

Shuxiao Chen, Edgar Dobriban, and Jane Lee. 2020a. A group-theoretic framework for data augmentation. *Advances in Neural Information Processing Systems*, 33:21321–21333.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020b. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Adam Coates, Andrew Ng, and Honglak Lee. 2011. 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.

Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. 2019a. Gauge equivariant convolutional networks and the icosahedral cnn. In *International conference on Machine learning*, pages 1321–1330. PMLR.

Taco Cohen and Max Welling. 2016a. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR.

Taco S Cohen, Mario Geiger, and Maurice Weiler. 2019b. A general theory of equivariant cnns on homogeneous spaces. *Advances in neural information processing systems*, 32.

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

Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi S Jaakkola. 2022. Diffdock: Diffusion steps, twists, and turns for molecular docking. In *The Eleventh International Conference on Learning Representations*.

Rumen Dangovski, Li Jing, Charlotte Loh, Seungwook Han, Akash Srivastava, Brian Cheung, Pulkit Agrawal, and Marin Soljacic. 2021. Equivariant self-supervised learning: Encouraging equivariance in representations. In *International Conference on Learning Representations*.

Pim De Haan, Maurice Weiler, Taco Cohen, and Max Welling. 2020. Gauge equivariant mesh cnns: Anisotropic convolutions on geometric graphs. In *International Conference on Learning Representations*.

Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. 2021. Vector neurons: A general framework for so (3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12200–12209.

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

Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Songyang Zhang, Haodong Duan, Wenwei Zhang, Yining Li, et al. 2024. Internlm-xcomposer2-4khd: A pioneering large vision-language model handling resolutions from 336 pixels to 4k hd. *arXiv preprint arXiv:2404.06512*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Thomas D Downs. 1972. Orientation statistics. *Biometrika*, pages 665–676.

Alexandre Agm Duval, Victor Schmidt, Alex Hernández-García, Santiago Miret, Fragkiskos D. Malliaros, Yoshua Bengio, and David Rolnick. 2023. FAENet: Frame averaging equivariant GNN for materials modeling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 9013–9033. PMLR.

Bryn Elesedy and Sheheryar Zaidi. 2021. Provably strict generalisation benefit for equivariant models. In *International conference on machine learning*, pages 2959–2969. PMLR.

Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. 2018a. Learning so (3) equivariant representations with spherical cnns. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–68.

Carlos Esteves, Christine Allen-Blanchette, Xiaowei Zhou, and Kostas Daniilidis. 2018b. Polar transformer networks. In *International Conference on Learning Representations*.

Ben Finkelshtein, Chaim Baskin, Haggai Maron, and Nadav Dym. 2022. A simple and universal rotation equivariant point-cloud network. In *Proceedings of Topological, Algebraic, and Geometric Learning Workshops 2022*, volume 196 of *Proceedings of Machine Learning Research*, pages 107–115. PMLR.

Johannes Gasteiger, Janek Groß, and Stephan Günnemann. 2019. Directional message passing for molecular graphs. In *International Conference on Learning Representations*.

Jonathan Gordon, David Lopez-Paz, Marco Baroni, and Diane Bouchacourt. 2020. Permutation equivariant models for compositional generalization in language. In *International Conference on Learning Representations*.

Nate Gruver, Marc Anton Finzi, Micah Goldblum, and Andrew Gordon Wilson. 2023a. The lie derivative for measuring learned equivariance. In *The Eleventh International Conference on Learning Representations*.

Nate Gruver, Anuroop Sriram, Andrea Madotto, Andrew Gordon Wilson, C Lawrence Zitnick, and Zachary Ward Ulissi. 2023b. Fine-tuned language models generate stable inorganic materials as text. In *The Twelfth International Conference on Learning Representations*.

Sharut Gupta, Joshua Robinson, Derek Lim, Soledad Villar, and Stefanie Jegelka. 2023. Structuring representation geometry with rotationally equivariant contrastive learning. In *The Twelfth International Conference on Learning Representations*.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Lingshen He, Yuxuan Chen, Zhengyang Shen, Yiming Dong, Yisen Wang, and Zhouchen Lin. 2021. Efficient equivariant network. In *Advances in Neural Information Processing Systems*.

Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. 2011. Transforming auto-encoders. In *Artificial Neural Networks and Machine Learning–ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21*, pages 44–51. Springer.

Emiel Hoogeboom, Jorn WT Peters, Taco S Cohen, and Max Welling. 2018. Hexaconv. In *International Conference on Learning Representations*.

Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, Liang Zhang, Bo Zhang, Chen Li, Ji Zhang, Qin Jin, Fei Huang, et al. 2024. mplug-docowl 1.5: Unified structure learning for ocr-free document understanding. *arXiv preprint arXiv:2403.12895*.

Li Hu, Xin Gao, Peng Zhang, Ke Sun, Bang Zhang, and Liefeng Bo. 2023. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. *arXiv preprint arXiv:2311.17117*.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. 2015. Spatial transformer networks. *Advances in neural information processing systems*, 28.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*.

Rui Jiao, Wenbing Huang, Peijia Lin, Jiaqi Han, Pin Chen, Yutong Lu, and Yang Liu. 2023. Crystal structure prediction by joint equivariant diffusion. *Advances in Neural Information Processing Systems*, 36.

Chaitanya K Joshi, Cristian Bodnar, Simon V Mathis, Taco Cohen, and Pietro Lio. 2023. On the expressive power of geometric graph neural networks. In *International Conference on Machine Learning*, pages 15330–15355. PMLR.

Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. 2023. Equivariance with learned canonicalization functions. In *International Conference on Machine Learning*, pages 15546–15566. PMLR.

Sékou-Oumar Kaba and Siamak Ravanbakhsh. 2023. Symmetry breaking and equivariant neural networks. In *NeurIPS 2023 Workshop on Symmetry and Geometry in Neural Representations*.

Mohamed A Khamsi and William A Kirk. 2011. *An introduction to metric spaces and fixed point theory*. John Wiley & Sons.

Jinwoo Kim, Dat Nguyen, Ayhan Suleymanzade, Hyeokjun An, and Seunghoon Hong. 2023. Learning probabilistic symmetrization for architecture agnostic equivariance. *Advances in Neural Information Processing Systems*, 36.

Jinwoo Kim, Saeyoon Oh, and Seunghoon Hong. 2021. Transformers generalize deepsets and can be extended to graphs & hypergraphs. *Advances in Neural Information Processing Systems*, 34:28016–28028.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026.

Risi Kondor and Shubhendu Trivedi. 2018. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *International conference on machine learning*, pages 2747–2755. PMLR.

Alex Krizhevsky et al. 2009. Learning multiple layers of features from tiny images.

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

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. 2022. Graphcast: Learning skillful medium-range global weather forecasting. *arXiv preprint arXiv:2212.12794*.

Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. 2007. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, pages 473–480.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Yann LeCun, Yoshua Bengio, et al. 1995. Convolutional networks for images, speech, and time series.

Taeyoung Lee. 2018. Bayesian attitude estimation with the matrix fisher distribution on so (3). *IEEE Transactions on Automatic Control*, 63(10):3377–3392.

Jan Eric Lenssen, Matthias Fey, and Pascal Libuschewski. 2018. Group equivariant capsule networks. *Advances in neural information processing systems*, 31.

Ren-Wu Li, Ling-Xiao Zhang, Chunpeng Li, Yu-Kun Lai, and Lin Gao. 2023. E3sym: Leveraging e(3) invariance for unsupervised 3d planar reflective symmetry detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14543–14553.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

Yuliang Liu, Biao Yang, Qiang Liu, Zhang Li, Zhiyin Ma, Shuo Zhang, and Xiang Bai. 2024b. Textmonkey: An ocr-free large multimodal model for understanding document. *arXiv preprint arXiv:2403.04473*.

David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110.

Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. 2024. Segment anything in medical images. *Nature Communications*, 15(1):654.

Chengzhi Mao, Lingyu Zhang, Abhishek Vaibhav Joshi, Junfeng Yang, Hao Wang, and Carl Vondrick. 2023. Robust perception through equivariance. In *International Conference on Machine Learning*, pages 23852–23870. PMLR.

Diego Marcos, Michele Volpi, Nikos Komodakis, and Devis Tuia. 2017. Rotation equivariant vector field networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. 2019. Provably powerful graph networks. *Advances in neural information processing systems*, 32.

Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. 2018. Invariant and equivariant graph networks. In *International Conference on Learning Representations*.

Amil Merchant, Simon Batzner, Samuel S Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. 2023. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85.

David Mohlin, Josephine Sullivan, and Gérald Bianchi. 2020. Probabilistic orientation estimation with matrix fisher distributions. *Advances in Neural Information Processing Systems*, 33:4884–4893.

Arnab Kumar Mondal, Vineet Jain, Kaleem Siddiqi, and Siamak Ravanbakhsh. 2022. Eqr: Equivariant representations for data-efficient reinforcement learning. In *International Conference on Machine Learning*, pages 15908–15926. PMLR.

Arnab Kumar Mondal, Siba Smarak Panigrahi, Oumar Kaba, Sai Rajeswar Mudumba, and Siamak Ravanbakhsh. 2023. Equivariant adaptation of large pretrained models. In *Advances in Neural Information Processing Systems*, volume 36, pages 50293–50309. Curran Associates, Inc.

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. 2019. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4602–4609.

Ryan Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. 2019. Relational pooling for graph representations. In *International Conference on Machine Learning*, pages 4663–4673. PMLR.

Ryan L Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. 2018. Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs. In *International Conference on Learning Representations*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Omri Puny, Matan Atzmon, Edward J Smith, Ishan Misra, Aditya Grover, Heli Ben-Hamu, and Yaron Lipman. 2021. Frame averaging for invariant and equivariant network design. In *International Conference on Learning Representations*.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. 2017. Equivariance through parameter-sharing. In *International conference on machine learning*, pages 2892–2901. PMLR.

Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, et al. 2021. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

David W Romero and Jean-Baptiste Cordonnier. 2020. Group equivariant stand-alone self-attention for vision. In *International Conference on Learning Representations*.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. 2021. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR.

Mehran Shakerinava, Arnab Kumar Mondal, and Siamak Ravanbakhsh. 2022. Structuring representations using group invariants. *Advances in Neural Information Processing Systems*, 35:34162–34174.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. 2022. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations*.

Kai Sheng Tai, Peter Bailis, and Gregory Valiant. 2019. Equivariant transformer networks. In *International Conference on Machine Learning*, pages 6086–6095. PMLR.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. 2024. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Llion Jones, Jakob Uszkoreit, Aidan N Gomez, and Ł ukasz Kaiser. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5994–6004.

Dian Wang, Robin Walters, and Robert Platt. 2021. So(2)-equivariant reinforcement learning. In *International Conference on Learning Representations*.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pages 9929–9939. PMLR.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. 2019. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12.

Maurice Weiler and Gabriele Cesa. 2019. General e (2)-equivariant steerable cnns. *Advances in neural information processing systems*, 32.

Maurice Weiler, Patrick Forré, Erik Verlinde, and Max Welling. 2021. Coordinate independent convolutional networks–isometry and gauge equivariant convolutions on riemannian manifolds. *arXiv preprint arXiv:2106.06020*.

Marysia Winkels and Taco S Cohen. 2022. 3d g-cnns for pulmonary nodule detection. In *Medical Imaging with Deep Learning*.

Daniel Worrall and Gabriel Brostow. 2018. Cubenet: Equivariance to 3d rotation and translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 567–584.

Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. 2017. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5028–5037.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.

Renjun Xu, Kaifan Yang, Ke Liu, and Fengxiang He. 2023. $e(2)$-equivariant vision transformer. In *Uncertainty in Artificial Intelligence*, pages 2356–2366. PMLR.

Dmitry Yarotsky. 2022. Universal approximations of invariant maps by neural networks. *Constructive Approximation*, 55(1):407–474.

Sergey Zagoruyko and Nikos Komodakis. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. 2017. Deep sets. *Advances in neural information processing systems*, 30.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. 2021. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pages 12310–12320. PMLR.

Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Sasha Shysheya, Jonathan Crabbé, Lixin Sun, Jake Smith, et al. 2023. Mattergen: a generative model for inorganic materials design. *arXiv preprint arXiv:2312.03687*.

Bohang Zhang, Shengjie Luo, Liwei Wang, and Di He. 2022. Rethinking the expressive power of gnns via graph biconnectivity. In *The Eleventh International Conference on Learning Representations*.

Jiayuan Zhu, Yunli Qi, and Junde Wu. 2024. Medical sam 2: Segment medical images as video via segment anything model 2. *arXiv preprint arXiv:2408.00874*.