ROBUSTNESS, SCALABILITY AND INTERPRETABILITY OF EQUIVARIANT NEURAL NETWORKS ACROSS DIFFERENT LOW-DIMENSIONAL GEOMETRIES

JOSHUA MITTON

SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
Doctor of Philosophy

SCHOOL OF COMPUTING SCIENCE

COLLEGE OF SCIENCE AND ENGINEERING
UNIVERSITY OF GLASGOW

MARCH 2023

Abstract

In this thesis we develop neural networks that exploit the symmetries of four different low-dimensional geometries, namely 1D grids, 2D grids, 3D continuous spaces and graphs, through the consideration of translational, rotational, cylindrical and permutation symmetries. We apply these models to applications across a range of scientific disciplines demonstrating the predictive ability, robustness, scalability, and interpretability.

We develop a neural network that exploits the translational symmetries on 1D grids to predict age and species of mosquitoes from high-dimensional mid-infrared spectra. We show that the model can learn to predict mosquito age and species with a higher accuracy than models that do not utilise any inductive bias. We also demonstrate that the model is sensitive to regions within the input spectra that are in agreement with regions identified by a domain expert. We present a transfer learning approach to overcome the challenge of working with small, real-world, wild collected data sets and demonstrate the benefit of the approach on a real-world application.

We demonstrate the benefit of rotation equivariant neural networks on the task of segmenting deforestation regions from satellite images through exploiting the rotational symmetry present on 2D grids. We develop a novel physics-informed architecture, exploiting the cylindrical symmetries of the group $\mathrm{SO}^+(2,1)$, which can invert the transmission effects of multi-mode optical fibres (MMFs). We develop a new connection between a physics understanding of MMFs and group equivariant neural networks. We show that this novel architecture requires fewer training samples to learn, better generalises to out-of-distribution data sets, scales to higher-resolution images, is more interpretable, and reduces the parameter count of the model. We demonstrate the capability of the model on real-world data and provide an adaption to the model to handle real-world deviations from theory. We also show that the model can scale to higher resolution images than was previously possible.

We develop a novel architecture which provides a symmetry-preserving mapping between two different low-dimensional geometries and demonstrate its practical benefit for the application of 3D hand mesh generation from 2D images. This models exploits both the 2D rotational symmetries present in a 2D image and in a 3D hand mesh, and provides a mapping between the two data domains. We demonstrate that the model performs competitively on a range of benchmark data sets and justify the choice of inductive bias in the model.

We develop an architecture which is equivariant to a novel choice of automorphism group through the use of a sub-graph selection policy. We demonstrate the benefit of the architecture, theoretically through proving the improved expressivity and improved scalability, and experimentally on a range of widely studied benchmark graph classification tasks. We present a method of comparison between models that had not been previously considered in this area of research, demonstrating recent SOTA methods are statistically indistinguishable.

Acknowledgements

First and foremost I would like to thank my supervisor Roderick for the guidance and advice offered throughout my PhD. I am grateful for the assistance in developing many successful collaborations as well as giving me the freedom to pursue research avenues that I found of particular interest. In addition to the technical assistance, I have learned a lot from the many conversations we have had and I am glad to have undertaken my PhD within such a welcoming environment.

I would also like to thank Klaas Wynne and Francesco Baldini for their supervision during my PhD, from which I learnt a great deal.

I am grateful to have been supported by a Lord Kelvin Adam Smith Scholarship from the University of Glasgow.

I am also thankful for my parents who have always supported me in whichever career directions I have chosen to take.

Finally, I would like to thank my partner for supporting and encouraging me throughout my PhD, as well as for her help during the final writing phase, for which I am very grateful.

Contents

List of Tables

List of Figures

1 Introduction

The research field of deep learning has rapidly grown in the past decade, overtaking human level performance in some aspects of computer vision (He et al., 2016), and playing games (Schrittwieser et al., 2020; Silver et al., 2017). Many recent advances have been a result of including prior knowledge into models (LeCun et al., 1989b; Mikolov et al., 2010). Although, this leads to a veritable zoo of separate architectures (Bronstein et al., 2021) and without a unified theoretical foundation it can be difficult to develop models for previously unutilised prior knowledge. Despite surpassing human level performance on some tasks, the research field of deep learning has not seen the same breakthroughs on tasks where datasets are small due to the time and cost associated with data collection. In this situation a different approach is required. As a result, in many instances deep learning models still struggle to generalise to objects or tasks that are out-of-distribution of the training dataset; cannot scale due to the high-dimensionality of the data; lack interpretability of how predictions are made and are not robust to symmetry transformations of the data. Developing deep learning models that overcome these issues could enable breakthroughs in applications that have previously been inaccessible.

A unification of the understanding of different types of geometry was developed and named the *Erlangen Programme* (Klein, 1872). This was achieved through the study of *invariants* and looking at which properties of each geometry remained unchanged used some class of transformations. This is known as the symmetries of the different geometries. The study of symmetries is especially important as it was used in Noether's Theorem (Noether, 1971), where conservation laws of physics can be derived from first principles using symmetry principles. Geometric deep learning is a framework for studying various deep learning architectures through the lens of symmetries. It attempts to provide a unified view of deep learning through considering the different inductive biases built into deep learning models through symmetries. Developing deep learning models with some inductive bias built in provides a way of incorporating prior knowledge into the model. By considering symmetries within the data and building models that are equivariant to such symmetries the space of functions possible to be learned by the model is reduced. Therefore, such inductive biases find a lower-dimensional subspace of the learning problem. This guarantees that the model is robust to symmetry transformations within the data.

In this thesis we develop novel equivariant neural networks for different scientific applications. This involves the consideration of different low-dimensional geometries, each relevant for the specific applications, and the symmetries associated to the specific task. When developing each equivariant network, which is robust to a symmetry transformation, we use the language of group theory, and hence considered different groups.[1] This thesis focuses on 1D grids and the group $\mathbb{Z}$ (Chapter 3); 2D grids and the groups $\mathrm{SO}(2)$ and $\mathrm{SO}^+(2,1)$ (Chapter 4); 2D grids and 3D manifolds, and the groups $C_8 \rtimes \mathbb{Z}_2$, $\mathbb{R}^2$, and $\mathbb{R}^3$ (Chapter 5); and graphs and the automorphism group (Chapter 6). Further, we develop novel models that exploit symmetries not previously considered, providing connections between the group theoretic understanding and the understanding from the domain of interest. We also show that using a suitable inductive bias, given by considering the symmetries of the task, improves the model's ability to solve the given task, whether it be classification, segmentation or generation. We show that equivariant models are able to better generalise to out-of-distribution data; scale better to higher-dimensional data and are more robust than their non-equivariant counterparts. We further demonstrate that models with suitable inductive bias are more interpretable than unconstrained models, and better align with the intuition of domain experts.

1.1 Outline and Contributions

The research contributions of this thesis are contained in Chapters 3, 4, 5, 6. Each chapter pertains to a specific data structure and its corresponding symmetry structure, where an application of the developed model is presented that is suitable for the respective symmetry structure. In essence each chapter develops a specific equivariant neural network and explores the benefits of such a model for various applications. In each chapter we address the symmetry structure and equivariant model developed; the relevant applications of the proposed model and either the robustness, scalability, or interpretability of the model. The thesis is structured as follows:

Chapter 2 introduces the necessary mathematical background for the entire thesis, cov-

[1] Further details on each group are provided in Chapter 2.

ering core concepts used in the following chapters.

Chapter 3 is based on the paper:

Siria, D.J., Sanou, R., Mitton, J., Mwanga, E.P., Niang, A., Sare, I., Johnson, P.C., Foster, G.M., Belem, A.M., Wynne, K. and Murray-Smith, R., 2022. Rapid age-grading and species identification of natural mosquitoes for malaria surveillance. *Nature Communications*, 13(1), pp.1-9.

My contribution to this paper was in leading the machine learning aspects of the paper, designing the architecture, and running all the experiments.

Chapter 4 is based on the papers:

Mitton, J. and Murray-Smith, R., 2021. Rotation Equivariant Deforestation Segmentation and Driver Classification. *NeurIPS 2021 Workshop on Tackling Climate Change with Machine Learning.*

Mitton, J., Mekhail, S.P., Padgett, M., Faccio, D., Aversa, M. and Murray-Smith, R., 2022. Bessel Equivariant Networks for Inversion of Transmission Effects in Multi-Mode Optical Fibres, *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*.

My contribution to this paper was in leading the paper, developing the theory, developing the connection between physics and group theory, designing the architecture, and running all the experiments.

Chapter 5 is based on the paper:

Mitton, J., Kaul, C. and Murray-Smith, R., 2021. Rotation Equivariant 3D Hand Mesh Generation from a Single RGB Image. *arXiv preprint* arXiv:2111.13023.

My contribution to this paper was in leading the paper, designing the architecture, and running all the experiments.

Chapter 6 is based on the paper:

Mitton, J. and Murray-Smith, R., 2021. Local Permutation Equivariance For Graph Neural Networks. *arXiv preprint* arXiv:2111.11840.

2

Background

Geometric deep learning concerns building neural networks that exploit some symmetries in high-dimensional data that arise due to the underlying low-dimensionality and structure of the physical world (Bronstein et al., 2017). These symmetries arise across a range of scientific disciplines and exist for data which have different low-dimensional geometries. Here we introduce some of these low-dimensional geometries which we seek to exploit. Furthermore, the study of symmetries most often concerns the use of group theory and as such this section introduces the necessary definitions and concepts. We also introduce the concept of equivariance which plays a crucial role in geometric deep learning and has a strong relation to group theory. Finally, we present the different groups and corresponding low-dimensional geometries considered throughout. As this work focuses on applications across a range of scientific disciplines we introduce the necessary preliminaries and background literature for each of these in the relevant chapter rather than introducing those here. Each chapter makes use of neural networks and core deep learning techniques for training the various models, we also introduce here.

2.1 Low-Dimensional Geometries

When we say low-dimensional geometries here we mean the domain on which the data lives. These domains are often the result of the process of physical measurement (Bronstein et al., 2021); for example images often live on a 2D grid as a result of the sensor array within a camera. Each chapter considers a different low-dimensional geometry with Chapter 3 considering the domain of 1D grids, Chapter 4 2D grids, Chapter 5 2D grids and 3D continuous spaces, and Chapter 6 graphs comprising nodes and edges. The various data types and their respective domain of interest is summarised in Table 2.1.

One physical measurement process utilised in this work is infrared spectroscopy which produces spectra. Spectra are sampled on a 1D grid, but form a high-dimensional space due to the large number of wavenumbers sampled. This high-dimensionality corresponds to a large n value in the domain column and spectra row in Table 2.1. The spectra are signals on this domain that are 1D real vectors corresponding to the absorption of light at the particular wavenumber sampled. These signals ($x : \Omega \to \mathbb{R}^1$) form a vector space, which we use as the input function or input feature space to the deep learning model. It is the regular structure of this high-dimensional 1D grid which we exploit in Chapter 3.

We also make use of images which are sampled on a 2D grid, but can form a high-dimensional space as the resolution of the image increases. This high-dimensionality corresponds to a large n value in the domain column and images row in Table 2.1. The images are sampled on this domain which create 3D real vectors corresponding to the RGB values typically associated to images. Similarly to the spectra, these signals ($x : \Omega \to \mathbb{R}^3$) form a vector space, which we use as the input function or input feature space to the deep learning model. In the instance grey scale images are used the signal changes to $x : \Omega \to \mathbb{R}^1$, although the only difference is that the vector space reduces in dimension to $\mathbb{R}^1$, reducing the input feature space of a neural network. It is the regular structure of this 2D grid which we exploit in Chapter 4.

Furthermore, we make use of point clouds which are sampled on a 3D continuous real space. Despite the input domain not being as high-dimensional as the spectra or an image, a point cloud does not generally comprise of just a single point and generally has n points, where n can become large. The points sampled on this domain create 3D real vectors corresponding to the x, y, z position vectors, which locate the point in the domain. Again, these signals ($x : \Omega \to \mathbb{R}^3$) form a vector space, which we use as the input function or input feature space to the deep learning model. It is worth noting this is not the only signal that could be used for a point cloud, for example if each point had a color associated with it this could also be used. In Chapter 5 we exploit the structure of both images and point clouds and produce a mapping between the two domains which preserves structure.

Finally, graphs consist of a set of nodes $\mathbb{V}$ and an adjacency tensor $\mathbf{A}$ which provides information on the connectivity of the graph through the edges connecting nodes. Similarly to a point cloud a graph generally consists of many nodes. Both the nodes and edges of the graph are signals ($x : \Omega \to \mathbb{R}^n$) which form a vector space and are used as the input function or input feature space to the deep learning model. This vector space is presented here very generally as graphs typically cover a wide range of applications which would each yield a different vector space $\mathbb{R}^n$. In Chapter 6 we exploit the structure of a graph, or more specifically the lack or ordering of the nodes.

Data Type	Domain Name	Domain Ω	Vector Space
Spectra	1D Grid	$\mathbb{Z}_n$	$\mathbb{R}^1$
Images	2D Grid	$\mathbb{Z}_n \times \mathbb{Z}_n$	$\mathbb{R}^3$
Point Clouds	3D Continuous Space	$\mathbb{R}^3$	$\mathbb{R}^3$
Molecules / Social Networks	Graphs	$(\mathbb{V}, \mathbf{A})$	$(\mathbb{R}^n, \mathbb{R}^n)$

Table 2.1: An overview of the various data types, their respective data domains, and the vector spaces considered for the specific type of data.

2.2 Symmetries

The symmetries of an object are usually referred to as the invariants of the object (Cohen & Welling, 2016). These symmetries are transformations that can be applied to the object which leave it unchanged (Cohen et al., 2021), which is to say the object is invariant to these transformations. Such transformations include—but are not exclusive to—translations, rotations, reflections, and permutations. For example, in computer vision when considering an image it is possible to translate each pixel and the resulting image remains an image of the same object. Therefore, translation is a symmetry which can be exploited in computer vision. In each chapter we exploit a different symmetry in the low-dimensional geometry in which the data lives.

A symmetry (or isometry) is a mapping $f : \Omega \to \Omega$ which preserves distances. Every symmetry is a bijection, meaning that it has a one-to-one correspondence. Again in the example of translational symmetry in computer vision, translating all of the pixels maps each pixel to a new location and no two pixels are mapped to the same location. In addition, every symmetry being a bijection means that each symmetry has an inverse (Bronstein et al., 2021). This can also be seen in the translation example, where, after translating an image there always exists another translation which moves each pixel back to its original location. Furthermore, symmetries can be combined to create a new symmetry (Bronstein et al., 2021). Therefore, given two symmetries $g : \Omega \to \Omega$ and $h : \Omega \to \Omega$, their compositions $g \circ h$ and $h \circ g$ are symmetries (Bronstein et al., 2021). Finally, a symmetry has an identity object $e : \Omega \to \Omega$ where $e \circ h = h$ (Bronstein et al., 2021); in the translation example this can be seen as the zero translation which doesn't move any pixels. These conditions, which are necessary to be a symmetry, form the laws required for an algebraic object to be known as a group. Geometric deep learning is usually defined through the lens of group theory and as such we provide further details and background in the following sections.

2.3 Functions

A function f can be thought of as a black box, where any element from $\mathcal{X}$ can be fed into the box, and an element from $\mathcal{Y}$ will be produced on the other end (Cameron, 2007).

The set of allowable inputs $\mathcal{X}$ to the function is called the domain of f (Cameron, 2007). Similarly, there is a set of allowable outputs $\mathcal{Y}$ to the function which is called the codomain of f (Cameron, 2007). It is worth noting that it is not strictly required that every possible value of the codomain $\mathcal{Y}$ be producible by the function f. Some examples of the input domains $\mathcal{X}$ are detailed in Table 2.1 under the Domain Ω column. An ordered pair (x, y) is a convenient way of detailing that an input x produces output y (Cameron, 2007).

Definition 1. A function $f : \mathcal{X} \to \mathcal{Y}$ is a subset of $\mathcal{X} \times \mathcal{Y}$ such that, for every element $x \in \mathcal{X}$, there is a unique element $y \in \mathcal{Y}$ for which $(x, y) \in f$ (Cameron, 2007).

As noted by Cameron (2007), a function is often called a map or mapping. This is due to the fact that a function f maps an input $x \in \mathcal{X}$ to an output $y \in \mathcal{Y}$. Throughout this work we will interchangeably use the terms map, mapping, and function.

2.4 Binary Operations

In Section 2.2 we used the notation $\circ$ to denote composition of two symmetries without saying anything about what this $\circ$ does. Here we define a binary operation and other useful relations for considering two mathematical objects.

Definition 2. Binary Operation $\circ$
A binary operation on a set A is a function f from $A \times A$ to A (Cameron, 2007).

Here we add the $\circ$ notation in with the binary operation definition as we make use of this notation throughout. This amounts to using the influx notation $a \circ b$ for $f(a, b)$ (Cameron, 2007). Another common approach is to use juxtaposition and simply write ab for $f(a, b)$ (Cameron, 2007). Examples of binary operations which typically have their own notation are $+$, $-$, $\cdot$, $\times$, and $*$ (Cameron, 2007).

Definition 3. Binary Relation
A binary relation R on a set A is a subset of the Cartesian product $A \times A$ (Cameron, 2007).

A binary relation can be thought of as a function which returns either true of false (Cameron, 2007). A binary relation takes in two elements from a set a and b and returns true if the pair satisfy the relation or false if they do not Cameron (2007).

Definition 4. Equivalence Relation
An equivalence relation, R is a binary relation on a set A which satisfies the following laws (Cameron, 2007):
(Reflexive law): $(a, a) \in R$ for all $a \in A$.
(Symmetric law): If $(a, b) \in R$ then $(b, a) \in R$.
(Transitive law): If $(a, b) \in R$ and $(b, c) \in R$ then $(a, c) \in R$.

2.5 Group Theory

Definition 5. Group
A group is a set G with a binary operation $\circ$, usually denoted $(G, \circ)$ satisfying the following laws (Cameron, 2007):
(G0) (Closure law): For all $g, h \in G,\ g \circ h \in G$.
(G1) (Associative law): $g \circ (h \circ k) = (g \circ h) \circ k$ for all $g, h, k \in G$.
(G2) (Identity law): There exists $e \in G$ such that $g \circ e = e \circ g = g$ for all $g \in G$.
(G3) (Inverse law): For all $g \in G$, there exists $h \in G$ with $h \circ g = g \circ h = e$.

Definition 5 provides a definition of a group, where a group is commonly written as $(G, \circ)$, although where the binary operation is not ambiguous we will write a group as G. The commutative law is not included within the definition of a group, although if a group also satisfies $g \circ h = h \circ g$ for all $g, h \in G$ then the group is called commutative or abelian.

Throughout this work we will follow the common method for reducing notations with groups and often write $g \circ h$ as gh. Further, we will use the power notation $g^n = g \cdot g \ldots g$ to abbreviate the combination of an element g with itself n times. Finally, we will often use g^{-1} for the inverse of the element g.

Example 1. Real numbers and addition.
The set $\mathbb{R}$ together with the binary operation $+$ yields a group $(\mathbb{R}, +)$ (Cameron, 2007). This is the group of real numbers with the binary operation of addition.

To show that $(\mathbb{R}, +)$ in Example 1 is a group we are required to consider each of the axioms.
(G0) For all $g, h \in \mathbb{R}$ we need to check that $g + h \in \mathbb{R}$. From the definition of $\mathbb{R}$ is clear that addition of two number yields a real number.
(G1) The order of summation of real numbers does not impact the result and hence $g + (h + k) = (g + h) + k$ for all $g, h, k \in \mathbb{R}$, so it is associative.

(G2) It is known that adding zero to a real number does not change it, hence $0 \in \mathbb{R}$ satisfies $g + 0 = 0 + g = g$ for all $g \in \mathbb{R}$, and it has an identity.
(G3) Finally, for each real number there exists the negative equivalent, such that for all $g \in \mathbb{R}$ there exists $-g \in \mathbb{R}$ satisfying $-g + g = g + -g = 0$.

Therefore we can conclude that $(\mathbb{R}, +)$ is a group.

Definition 6. Subgroup
Given a group G, a subgroup of G is a subset of G which, using the same operation as in G, is itself a group (Cameron, 2007). A subgroup H of G is denoted by $H \leq G$.

Now that we have the definition of groups and subgroups, we are interested in mappings between these structures. We need to be able to define mappings between different structures to be able to consider different symmetries of the structures.

Definition 7. Group Homomorphism
Given two groups G and H, a homomorphism $\theta : G \to H$ is a function θ from G to H that satisfies the condition (Cameron, 2007)

$$(g_1 g_2)\theta = (g_1\theta)(g_2\theta) \;\; \forall g_1, g_2 \in G. \tag{2.1}$$

Definition 8. Group Isomorphism
A homomorphism that is one-to-one and onto is called an isomorphism (Cameron, 2007). That is a group isomorphism is bijective (surjective and injective).

Two groups G and H are called isomorphic is there is an isomorphism between them. Two groups being isomorphic means that from the point of view of abstract algebra they are the same, even if their elements are completely different (Cameron, 2007).

Definition 9. Group Automorphism
Given a group G a group automorphism is an isomorphism from G to G.
Let $\mathcal{X}$ be some mathematical object for which we can formulate the notion of homomorphism (or isomorphism). Then an automorphism of $\mathcal{X}$ is an isomorphism $\theta : \mathcal{X} \to \mathcal{X}$; in other words, it is a permutation of $\mathcal{X}$ which happens also to be a homomorphism satisfying (Cameron, 2007)

$$(x \circ y)\theta = x\theta \circ y\theta. \tag{2.2}$$

Let $\mathrm{Aut}(\mathcal{X})$ be the set of all automorphisms of $\mathcal{X}$. Then $\mathrm{Aut}(\mathcal{X})$ is a group, the automorphism group of $\mathcal{X}$. This can also be considered for a group rather than a general object $\mathcal{X}$ and therefore we can talk about the automorphism group of a group (Cameron, 2007).

An important equivalence relation defined on a group is that of conjugacy. Conjugacy provides a measure of similarity between elements in the group and allows for the splitting of group elements into equivalence classes. Each equivalence class is a unique partition of the group elements. The group G hence can be thought of as the disjoint union of equivalence classes, which are called conjugacy classes.

Definition 10. Conjugacy
For a group G, two elements of the group g, h of G are conjugate (written $g \sim h$) if $h = g^{-1}xg$ for some $g \in G$ (Cameron, 2007).

Throughout this work we are interested in how a group acts upon some data. Equipped with the concept of a low-dimensional geometry which is the underlying domain on which the data lives, we can consider how the group acts upon this domain. Following this, we are interested in how the group acts upon the signals (features) which live on this domain.

Definition 11. Group Action
A group action of a group G on a set Ω is a function $\mu : \Omega \times G \to \Omega$ with the following two properties (Cameron, 2007):
(GA1) $\mu(\mu(u, g), h) = \mu(u, gh)$ for all $u \in \Omega,\ g, h \in G$.
(GA2) $\mu(u, 1) = u$ for all $u \in \Omega$, where 1 is the identity of G.

Therefore if we have a group G acting on a base space Ω, we automatically obtain an action of G on the feature space $\mathcal{X}(\Omega)$ (Bronstein et al., 2021):

$$(gx)(u) = x(g^{-1}u) \tag{2.3}$$

for $g \in G$, $x \in \mathcal{X}(\Omega)$, and $u \in \Omega$.

Throughout this work we will make use of linear group actions, known as group representations (Bronstein et al., 2021). This type of group action is of particular interest for deep learning as typically the functions learned to update the feature spaces $\mathcal{X}(\Omega)$ are linear. The non-linearity is introduced through specific non-linear functions inserted after each linear update function. Therefore, introducing group representations, which are linear group actions, is a well studied tool which can be utilised in deep learning to constrain the update functions such that they respect the properties of a group action. Given the connection made between symmetries and group theory (Cohen et al., 2021; Cohen & Welling, 2016; Bronstein et al., 2021) considering group representations in the context of deep learning provides a method of constraining neural networks to respect the symmetries of the considered low-dimensional geometry.

2.6 Representation Theory

The feature spaces considered in deep learning are typically vector spaces. The concept of a vector is quite commonly introduced in high-school or any engineering or science degree, and is generally distinguished from scalar values by the use of bold type. On the other hand, a vector space can be considered as an algebraic object, just like a group. For a formal definition of a vector space we would divert the reader to Cameron (2007).

As stated at the end of Section 2.5, we are particularly interested in linear maps in deep learning. Often a homomorphism of a vector spaces is called a linear transformation or linear map, and as such we adopt this notion.

Definition 12. Linear Transformation
For a vector space V a linear transformation from V to V is a function $\theta : V \to V$ which satisfies the following conditions (Cameron, 2007):

$$(v_1 + v_2)\theta = v_1\theta + v_2\theta \tag{2.4}$$

$$(cv_1)\theta = c_1(v_1\theta) \tag{2.5}$$

for scalars c and $v_1, v_2 \in V$.

Definition 13. Finite Group Representation
A representation of a finite group G on a finite-dimensional complex vector space V is a homomorphism $\rho : G \to \mathrm{GL}(V)$ of G to the group of automorphisms of V (Fulton & Harris, 2013).

Similarly, we can define a group representation as a map $\rho : G \to \mathbb{C}^{n\times n}$ that assigns to each group element $g \in G$ an invertible matrix $\rho(g)$, and further satisfies $\rho(gh) = \rho(g)\rho(h)$ (Bronstein et al., 2021).

Throughout this work we consider feature spaces as vector spaces. Therefore, group representations provide a convenient way represent the group actions on such vector spaces and constrain the linear transformations with some symmetry group. In general the dimensions n of the invertible matrix $\rho(g)$ will be equal to the dimensionality of the feature space on which the group acts $\mathcal{X}(\Omega)$.

An example of a group representation, which is used in later chapters, is that of the rotation group $\mathrm{SO}(2)$.

Example 2. Rotation Group $\mathrm{SO}(2)$
The group representation given as a homomorphism to the general linear group and as

a map to rotation matrices Cesa (2020):

$$\rho : \mathrm{SO}(2) \to \mathrm{GL}(\mathbb{R}^2),\ \rho : \mathrm{SO}(2) \to \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{2.6}$$

The rotation group $\mathrm{SO}(2)$ is the group of all planar rotations and rotates the vector space through the angle θ.

Definition 14. Equivalent Representations
Two representations ρ and ρ' on a vector space V are called equivalent or isomorphic iff they are related by a change of basis $Q \in \mathrm{GL}(V)$ (Cesa, 2020),

$$\forall g \in G,\ \ \rho'(g) = Q\rho(g)Q^{-1}. \tag{2.7}$$

Equivalent relations therefore behave similarly. This can also been seen for composition of representations through (Cesa, 2020):

$$\rho'(g_1)\rho'(g_2) = Q\rho(g_1)Q^{-1}Q\rho(g_2)Q^{-1} = Q\rho(g_1)\rho(g_2)Q^{-1}. \tag{2.8}$$

Similarly to a group action providing details on the action of a group on a base space, a group representation details how the group acts on signals $x \in \mathcal{X}(\Omega)$ (Cohen et al., 2021)

$$\rho(g)x(u) = x(g^{-1}u), \tag{2.9}$$

which also satisfies

$$(\rho(g)(\rho(h)))x(u) = (\rho(gh)x)(u). \tag{2.10}$$

It is also useful in the context of deep learning to consider mappings between different group representations. An example, which will be seen later in Chapter 6, is that of a mapping from a graph to a set or from a graph to pooled graph. These will both be mappings between different group representations. A map ψ between two representations $\rho(G)(V)$ and $\rho(G)(W)$ on different vector spaces V and W, but of the same group G is a vector space map $\psi : \rho(G)(V) \to \rho(G)(W)$ such that

$$\begin{tikzcd}
\rho(G)(V) \arrow[r, "\psi"] \arrow[d, "g"'] & \rho(G)(W) \arrow[d, "g"] \\
\rho(G)(V) \arrow[r, "\psi"'] & \rho(G)(W)
\end{tikzcd}$$

commutes for every $g \in G$. This is denoted a G-linear map to distinguish it from an arbitrary linear map.

Definition 15. Direct Sums of Representations
Multiple representations can be combined by taking the direct sum of the representations. Given two representations ρ_1 and ρ_2 of a group G, their direct sum $\rho_1 \oplus \rho_2$ is defined as (Weiler & Cesa, 2019; Finzi et al., 2021)

$$(\rho_1 \oplus \rho_2)(g) = \begin{bmatrix} \rho_1(g) & 0 \\ 0 & \rho_2(g) \end{bmatrix}, \tag{2.11}$$

which generalises to multiple representations as

$$\bigoplus_i \rho_i(g) = \rho_1(g) \oplus \rho_2(g) \oplus \ldots. \tag{2.12}$$

Definition 16. Irreducible Representations (Irreps)
A representation is called irreducible if it can only be decomposed as the direct sum of trivial representations.

Definition 17. Regular Representation
The regular representation of a finite group G acts on itself by translation. The action on a G-dimensional vector space permutes its axes.

2.7 Equivariance

The previous background sections provided a background into group theory and developed this to provide an understanding of group representations and how these govern group actions on the low-dimensional geometry of the data domain Ω. This provides the mathematical concepts required to study how the symmetries of the domain Ω underlying the signals $\mathcal{X}(\Omega)$ impose structure on functions f defined on the signals. This allows us to build on the definitions of maps between representations and build maps such that the functions are constrained by imposing structure from the symmetries chosen. This is a powerful inductive bias, as it reduces the space of possible functions f (Weiler & Cesa, 2019). This inductive bias of symmetry can be understood through the terms equivariance and invariance and we define this now.

Definition 18. Equivariance
A function $f : \mathcal{X}(\Omega) \to \mathcal{X}(\Omega)$ is said to be equivariant to the group G (Bronstein et al., 2021) if

$$f(\rho(g)x) = \rho(g)f(x) \ \ \forall g \in G, \ x \in \mathcal{X}(\Omega) \tag{2.13}$$

Equivariance states that if a group action $\rho(g)$ is applied on the input feature space x before passing the features through a function f, it produces the same output as first passing the input feature space x through the function f and then applying the group action $\rho(g)$.

Definition 19. Invariance
A function $f : \mathcal{X}(\Omega) \rightarrow \mathcal{X}(\Omega)$ is said to be invariant to the group G (Bronstein et al., 2021) if

$$f(\rho(g)x) = f(x) \;\; \forall g \in G, \; x \in \mathcal{X}(\Omega) \tag{2.14}$$

Invariance states that if a group action $\rho(g)$ is applied on the input feature space x before passing the features through a function f, it produces the same output as only passing the input feature space x through the function f. Therefore, the output of the function f is unaffected by the group action on the input.

2.8 Training Deep Learning Models

Machine learning can broadly be categorised into three different branches supervised, unsupervised, and semi-supervised. Supervised machine learning characterises the scenario when there exists data samples and corresponding labels or targets to be predicted. On the other hand unsupervised machine learning covers the scenario where data samples exist but there is no corresponding labels. Semi-supervised machine learning sits somewhere between the two where an algorithm should learn from both labelled and unlabelled data. This thesis focuses on supervised machine learning and as such the following details will be more specific to the supervised machine learning scenario.

2.8.1 Neural Networks

In supervised machine learning it is often assumed that there is a set of N samples where each consists of some data point x and label y, yielding a dataset of $\mathcal{D} = \{(x_i, y_i)\}_{i=0}^{N}$ (Bronstein et al., 2021). It is assumed that these samples are drawn $i.i.d$ from an underlying distribution defined over $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are the data and label domains respectively (Bronstein et al., 2021). It is further assumed that the labels $y \in \mathcal{Y}$ are generated by some unknown function f, such that $y_i = f(x_i)$ (Bronstein et al., 2021). The task of supervised machine learning is therefore to estimate the function f by training a model on the dataset $\mathcal{D}$. In deep learning this function estimation is accomplished by learning some parameterised function class $\mathcal{F} = \{f_{\theta \in \Theta}\}$, where the parameters $\theta \in \Theta$

are the network weights (Bronstein et al., 2021).

2.8.2 Transfer Learning and Fine-Tuning

Transfer learning is a training technique in machine learning where a pre-trained model is re-purposed for a new task. This necessitates that a model has already been trained for a task previously. There is an assumption that the two tasks are sufficiently similar that there are features within the pre-trained model that will be useful for the new task. These features can then be re-purposed or transferred to the new task. Fine-tuning is also a training technique that is used in conjunction with transfer learning and refers to the situation where some of the weights are unfrozen and trained from the initialisation of a model which was trained on a different task. Fine-tuning can also incorporate the situation where some additional weights are added to a pre-trained model and then trained.

When both transfer learning and fine-tuning are used as techniques to train a model some weights, typically those in early layers of the model, are frozen and some weights, typically those in later layers of the model, are un-frozen and trained. The freezing of some weights in the model fixes the feature extraction of this part of the model. The model is re-trained to update the non-frozen weights to re-purpose the model for a new task. This concept of transferring the weights to a new model to solve a new task and freezing part of the model is demonstrated in Figure 2.1.

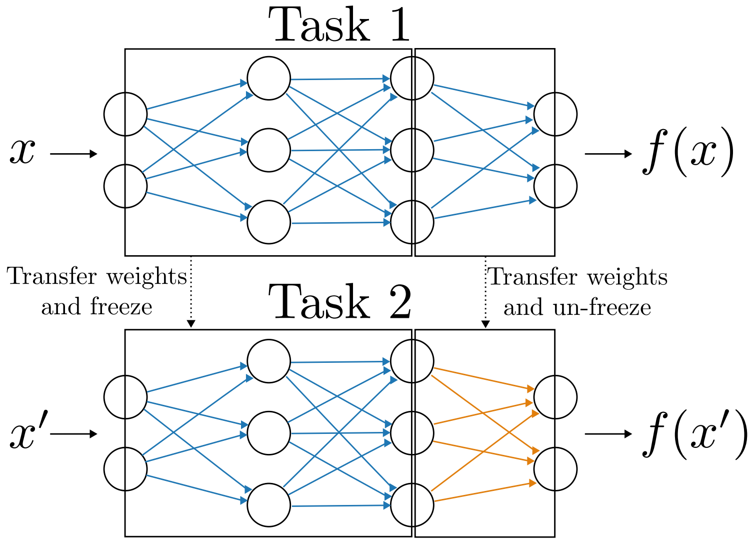


Figure 2.1: Example of transfer learning and fine-tuning. An initial model is trained on task 1. Then the weights are transferred to a new model for task 2. This model is then fine-tuned by freezing some of the weights while un-freezing and training other weights.

Transfer learning is used when there is a relatively small dataset, but there exists a much larger dataset which has similar properties. Therefore, the large dataset can be used to train a general model, which will learn useful features for solving the specific task associated to that dataset, and is less likely to suffer from over-fitting issues due to the size of the dataset. Then, these learned features can be taken advantage of for the new task associated to the smaller dataset, where it would be more likely that the model would over-fit to the smaller dataset and generalise poorly if this pre-training was not utilised. The main goal of using transfer learning and fine-tuning is to improve the generalisation of the model, speed up the training time, and re-use feature capability from a prior task.

3

Translational Symmetries on 1D Grids

In this chapter we develop a model comprising of translation equivariant 1D convolutions for the task of predicting mosquito age and species. We demonstrate that the model can predict mosquito age and species with a high level of accuracy for lab, semi-field, and wild data, overcoming the challenges of working with small wild data sets through the use of transfer learning from a model trained on lab data. Further, we show that the model is sensitive to regions of chemical relevance in the input spectra. Finally, we demonstrate that the model can be a useful tool in assessing the impact of malaria intervention schemes.

3.1 Introduction

This chapter builds on the development of translation equivariant models to develop a model for the prediction of mosquito age and species from mid-infrared spectra. Mid-infrared spectra consist of a 1D feature space corresponding to the absorption of light which lives on a high-dimensional 1D grid due to the large number of wavenumbers considered. We develop a model that consists of multiple layers that respect the symmetry of translation, which here means these layers enforce equivariance to the group $\mathbb{Z}$ on a domain of 1D grids; this is achieved through the use of 1D convolutions. Following the translation equivariant layers we add pooling layers and fully connected layers for the final prediction of both the age and species of the input sample. The use of translation equivariant 1D convolutions allows the model to extract features at different spatial locations using the same model weights, providing an efficient weight sharing scheme for high-dimensional data. Spectra consist of many peaks and troughs, created by absorption of light due to the presence of different molecules within the mosquito, and each of these can be a superposition of many smaller peaks and troughs produced by the different cuticular components they belong to (chitin, proteins, or waxes). As these occur at different wavenumbers, being able to detect peaks at differing wavenumbers is of importance. Therefore, the translation equivariance of 1D convolutions allows the feature extraction from peaks and troughs to be shared spatially, which is of importance in mid-infrared spectra. Furthermore, having the translational weight sharing provided by a translation equivariant network allows structures to be detected within the signal more efficiently than a non-equivariant model, which is important in high-dimensional data. We empirically demonstrate the performance of the model on data of mosquitoes reared in the lab and semi-field, and those collected in the wild. For the semi-field and wild data we overcome the challenge of learning on a very small data set through the use of transfer learning, re-using the translation equivariant feature extraction part of the model pre-trained on lab data. In addition, we perform a sensitivity analysis on the model to assess which input wavenumbers the model prediction is sensitive to and show that this corresponds to regions within the spectra that have chemical relevance. Finally, for the wild data, where there is a lack of labelled data, we demonstrate that the predicted age distribution of the model is similar to the true distribution of wild mosquitoes, which would be of practical relevance for population monitoring used to assess the impact of malaria intervention schemes. Our contributions are:

- A model comprising translation equivariant 1D convolutions, which overcomes the high-dimensionality of spectra data without feature selection for mosquito age and species prediction.
- A transfer learning method for overcoming the challenge of working with small mosquito data sets, which allows the model to generalise to wild and semi-wild data.
- A demonstration that the proposed model predicts age and species more accurately than prior methods.
- A sensitivity analysis demonstrating model sensitivity to chemically relevant regions in the input spectra.
- A demonstration of how the model can be used to predict age distributions of wild mosquitoes.

3.2 Background

3.2.1 Mosquitoes and Mid-Infrared Spectroscopy

The transmission of malaria presents a paradox; it depends on malaria mosquitoes that have a shorter mean lifespan than the malaria parasite requires for its development to become transmissible (Beier, 1998). Consequently, the persistence of malaria is dependent upon the the small proportion of mosquitoes that survive long enough to transmit the malaria sporozoites to a mammalian host. As a result, small changes in mosquito longevity have a big impact on malaria transmission (Macdonald, 1956). Malaria control interventions have therefore focused on targeting adult mosquitoes, and have reduced the incidence of malaria in Africa (Macdonald et al., 1952; Bhatt et al., 2015). Despite this progress, the effectiveness of control interventions maybe threatened by insecticide resistance (Churcher et al., 2016).

A method for rapid, accurate, and reliable assessment of mosquito age structure is of crucial importance for monitoring the impact of mosquito control interventions. However, current mosquito age grading methods typically rely on old techniques based on ovary dissections such as that proposed by Beklemishev et al. (1959) that are slow, labour-intensive, coarse, and imprecise, and which vary between mosquito species (Hugo et al., 2014; Johnson et al., 2020). Many alternatives have been investigated with variable success (Schlein, 1979; Caputo et al., 2005; Cook et al., 2007; Bass et al., 2007; Mayagaya et al., 2009; Sikulu et al., 2015). Malaria is transmitted by multiple, often morphologically indistinguishable, mosquito species that differ in longevity and behaviour (Ferguson et al., 2010; Cohuet et al., 2010). Therefore, the most useful method for the prediction of mosquito age would also predict the mosquito species and would not rely on time-consuming techniques and expensive reagents.

The cuticle of a mosquito, the outermost part of the body, has a chemical composition which differs between different species and changes with age (Johnson et al., 2020). Infrared spectroscopy can detect changes in mosquito cuticle by quantifying how it absorbs light (Mayagaya et al., 2009; Lambert et al., 2018). Mid-Infrared Spectroscopy (MIRS) measures the absorption of light in the wavenumber region of 4000-400 cm^{-1}. MIRS measures discrete fundamental vibrations of bio-molecules, allowing information to be extracted from biological samples (Waynant et al., 2001; Sorak et al., 2012). Therefore MIRS is a useful technique for extracting information about the chemical composition of mosquitoes. Despite this benefit, MIRS provides a 1D signal that has a high-dimensional feature space, which presents a challenge for non-ML techniques and generally requires some pre-processing (Jiménez et al., 2019).

3.2.2 Translation Equivariance

In this chapter we are concerned with translation equivariance on 1D grids due to the improved weight sharing offered by the inductive bias of a 1D convolution. We are therefore interested in the translation group $G = \mathbb{T}$ acting on the domain $\mathcal{X}(\Omega) = (\mathbb{Z}_n, +)$ of an n-dimensional grid. Therefore the equivariance constraint of a translation equivariant layer, f, is:

$$f(\rho(g)x) = \rho(g)f(x) \ \ \forall g \in \mathbb{T}, \ x \in (\mathbb{Z}_n, +), \tag{3.1}$$

which states that if the input features x are first translated by the group action $\rho(g)$ before passing through the function (neural network layer) f, this should equal first passing the input features x through the function (neural network layer) f and then translating by the group action $\rho(g)$. This constraint is satisfied by a 1D convolution implemented efficiently in most deep learning frameworks.

A convolutional layer can be viewed as a set of convolutional kernels which are convolved with the image. This amounts to multiplying the kernels weights with the features of a region of the input domain that is of the same size as the kernel, and then repeating this process for each distinct region on the input domain. The act of repeating for each distinct region on the input domain highlights that the convolution operation will similarly process an input independent of a translation action applied to the base space. A discrete convolution operation is expressed through:

$$y = \sum_{i=0}^{n-1} \sum_{k=0}^{n-1} w_{(i-k) \mathrm{mod} n} x_k, \tag{3.2}$$

for output y, input x, and weight matrix w.

Writing this in matrix multiplication yields a circulant matrix as the weight matrix, which is formed by stacking w n-times shifted modulo n. Hence, convolution can be written as matrix multiplication with a circulant matrix. A circulant matrix has a multi-diagonal structure with elements on each diagonal sharing the same value. Circulant matrices have an interesting property that they commute, i.e. $W_0 W_1 = W_1 W_0$ for circulant matrices W_0 and W_1. Here we are interested in translation equivariant functions, hence it is worth noting that translation can be written as a shift matrix S, which is also a circulant matrix. Therefore the translation equivariance of convolution can be seen through matrix multiplication with shift matrices:

$$WS^T = S^T W, \tag{3.3}$$

for the weight matrix W from a convolutional kernel and shift matrix S.

Translation equivariant models were considered in the convolutional neural network of LeCun et al. (1989a), where this translational symmetry was utilised as part of a model for image classification. Since their introduction, convolutions have been a key component of recent developments of neural networks for image based tasks with many models introducing new components to the overall model such as pooling, regularisation, non-linearities, batch normalisation, or skip connections to advance the state-of-the-art in image classification (LeCun et al., 1998; Krizhevsky et al., 2012; Simonyan & Zisserman, 2015; Szegedy et al., 2014; He et al., 2016). Despite most advances being made in the image domain the techniques developed and advances made are applicable for 1D convolutions.

3.3 Mosquito Age and Species Classification

3.3.1 Datasets

In this chapter, four datasets are used:

1. A dataset of mosquito larvae from the lab which were reared in the lab (LV).
2. A dataset of mosquito larvae from the field which were reared in the lab (GV).
3. A dataset of mosquito larvae from the lab that were reared in a semi-field environment (EV).
4. A dataset of mosquito larvae from the field that also developed in the field (Wild).

Each dataset has a different purpose with the ultimate goal to be able to correctly predict the age and species of the wild mosquitoes. The LV dataset is relatively easy to collect as the age can be easily monitored and the species is known; further, it is possible to have a higher degree of control over the quantity of each group of mosquitoes collected. The GV dataset also shares many of the benefits of the LV dataset due to being reared in the lab, but introduces genetic variability due to not coming from a population of mosquitoes that have existed for many generations in the lab. The EV dataset closer replicates wild mosquitoes due to being reared in a semi-field environment and it is therefore of interest to be able to correctly predict the age and species of these mosquitoes. The EV dataset has the advantage that the age of mosquitoes is easily known by monitoring the date they are released into the semi-field from the lab. Conversely, as we only had access to one semi-field environment for each location, this dataset is relatively small in the machine

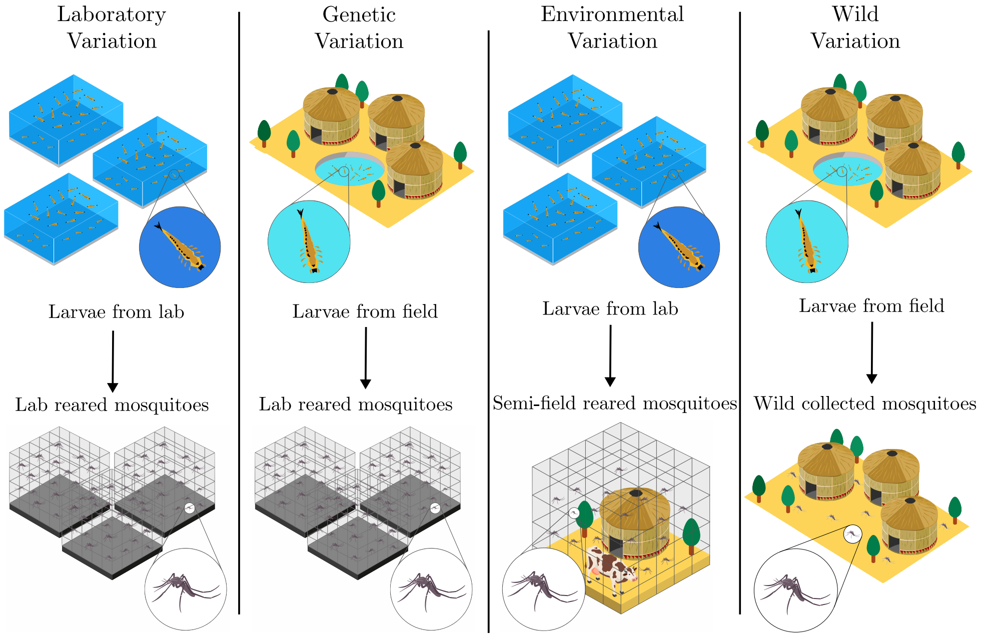


Figure 3.1: Four different datasets are used to determine the ability of the model to predict age and specifies of mosquitoes. Each introduces a new variation making the task more difficult, yet closer to the required prediction ability for this method to assist in population understanding of mosquitoes. The laboratory variation (LV) comprises mosquito larvae from the lab which are reared in the lab, the genetic variation (GV) comprises mosquito larvae from the field which are reared in the lab, the environmental variation (EV) comprises mosquito larvae from the lab which are reared in a semi-field environment, and the wild variation (Wild) comprises mosquito larvae from the field which subsequently develop in the field and are collected from the field. This figure is adapted from Siria et al. (2022).

learning context, which presents a challenge for deep learning algorithms. Finally, the Wild dataset is of particular interest as these are truly wild mosquitoes collected in the field and the main goal is to be able to predict age and species of these mosquitoes. Details on the method for determining the chronological age of wild mosquitoes is provided in (Siria et al., 2022). Due to the time cost of dissecting mosquitoes, not all wild collected mosquitoes are dissected. To independently test model predictions, the non-dissected mosquitoes are also scanned to produce MIRS. These were selected at random from the same populations as the dissected ones. Therefore, it is assumed that the age structure of dissected and non-dissected mosquitoes should be similar. This dataset is challenging to work with as it is very small and highly unbalanced due to the challenges of collecting and dissecting the mosquitoes to determine the age, which presents a challenge from a deep learning perspective. The key differences between each dataset are shown in Figure 3.1.

For each dataset the input data is a mid-infrared spectrum and the target to be predicted by the model is the age and species of the mosquito. The input spectrum contains information about the chemical composition of the mosquito and the task for the model is to learn how the shape of the signal correlates with age and species. Therefore, the model is required to learn to correctly predict the mosquito age and species from these 1D signals. For the age prediction task we group mosquitoes into three different classes, those which are:

- 1-4 days old (not infected).
- 5-10 days old (potentially infected but not infectious).
- ≥ 11 days old (old enough to be infectious).

For the species prediction task the three different species form the three different classes, these are:

- An. Arabiensis.
- An. Coluzzii.
- An. Gambiae.

The input mid-infrared spectra used throughout this chapter consist of 1550 different wavenumbers. An example of spectra for different age and species of mosquitoes is provided in Figure 3.2.

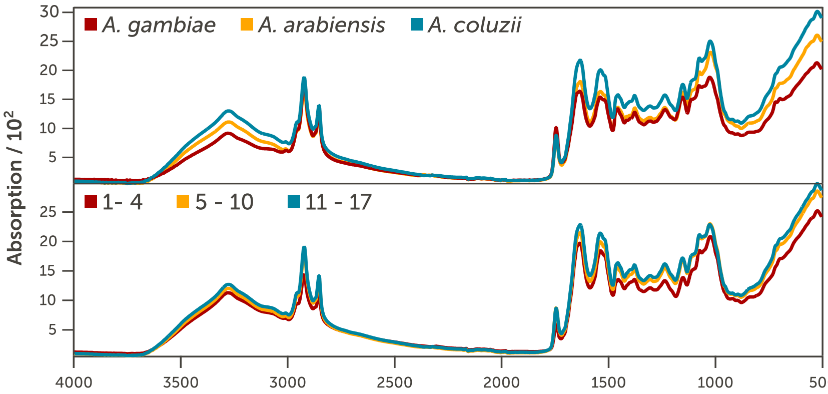


Figure 3.2: Representative variation of mid-infrared absorption spectra of An. Arabiensis, An. Coluzzii and An. Gambiae and of the three age groups. This figure is from Siria et al. (2022).

In addition to the description of each dataset, we perform some preliminary analysis of the data using UMAP (McInnes et al., 2018). This unsupervised clustering shown in Figure 3.3 reveals signatures within the MIR spectra of both geographic origin and rearing conditions. This discrepancy between clusters indicates two properties of these datasets:

- Firstly, there exists useful variation between species, suggesting that MIRS are predictive of species.

- Secondly, there is unsurprising (Krajacich et al., 2017; Khoshmanesh et al., 2017) variation between mosquito origins and rearing environments, indicating that models based on cuticle composition must include representative samples from each origin to statistically adjust for origin bias.

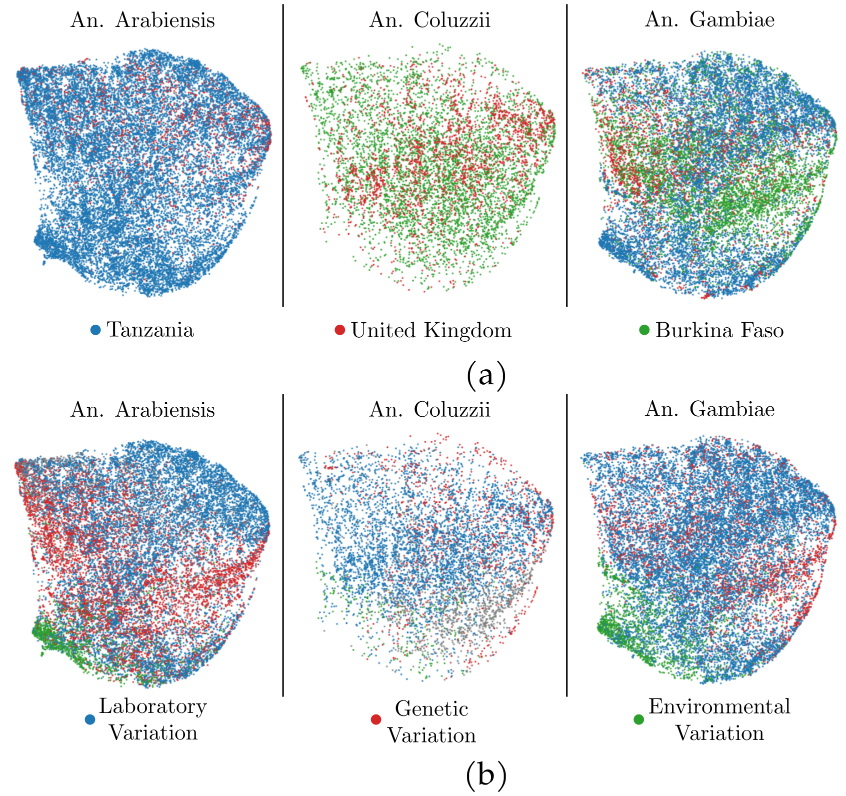


Figure 3.3: UMAP plots showing an unsupervised clustering of MIRS data projecting down to 2-dimensions. (a) coloured according to site of origin and (b) source of variation.

3.3.2 Model Specification

3.3.2.1 Model Architecture

In Section 3.2.2 we stated that 1D convolutions provide an efficient implementation of a deep neural network layer that is translation equivariant. Further, given the nature of mid-infrared spectra, consisting of multiple peaks and troughs on a 1D grid, utilising 1D convolutions is a well motivated method to learn from the input features. This is due to the need to extract information about the peaks and troughs at different spatial locations in an efficient manner. The inductive bias of 1D convolutions, which shares kernel weights spatially across the entire input, provides exactly this. The translation equivariance of 1D convolutions means that if a peak occurs in a different spatial location in the input spectra it will be processed similarly by the same weights. This is due to the occurrence of multiple similar peaks in different spatial locations appearing locally like a shift in the input spectra and convolutions have a fixed size receptive field meaning they process local patches of the input.

As a result of the fact that 1D convolutions are efficient and physically motivated for the input data to our model, we build a model that comprises multiple 1D convolutional layers followed by fully connected layers. We interleave pooling layers between the convolutional layers to reduce the high-dimensionality of the data. As a result, the fully connected layers operate on a lower-dimensional data domain. The fully connected layers allow information sharing across the entire data domain allowing learned features to be combined from either end of the spectra in a way that is not possible with only convolutions (unless sufficiently many convolutional layers are used, which we do not do). Further, the fully connected layers produce the final output prediction of both age and species. An overview of the architecture of the model is given in Figure 3.4.

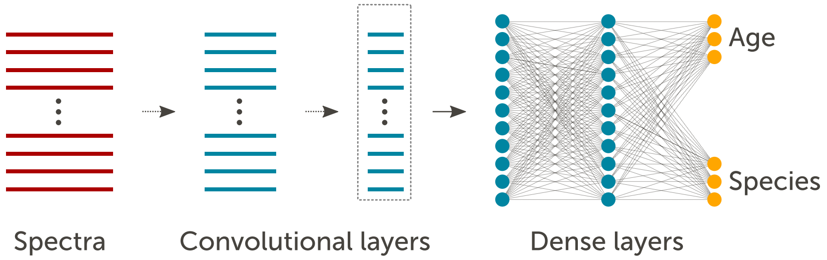


Figure 3.4: Schematic representation of the deep convolutional neural network that takes MIR spectra as inputs and outputs mosquito age and species. The input layer (wavenumber values) is fed through five 1-dimensional convolutional layers, comprising of 16 filters each (convolutional layers region), followed by a dense layer of 500 features and age and species output layers (dense layers) that were used to make predictions. This figure is from Siria et al. (2022).

In the early convolutional layers of the model, batch normalisation (Ioffe & Szegedy, 2015) was used to improve the stability of the neural network and max pooling was used to reduce the spatial size of the representation. Further, $\ell 2$ regularisation (Krogh & Hertz, 1991; Schmidhuber, 2015) was used in each layer to reduce overfitting, with a convolutional stride of size one in convolutional layers one, three and five, and stride of two in convolutional layers two and three. We used size two max pooling in the final convolutional layer, and dropout was applied before the dense layer to further reduce overfitting. The convolutional neural network architecture was found by optimising the hyper- parameters. For this, the number of layers was hand optimised, while the kernel, stride, and pooling sizes for each convolutional layer were optimised using gp_minimize from scikit-optimise (Pedregosa et al., 2011). The overall architecture is provided in more detail in Figure 3.5.

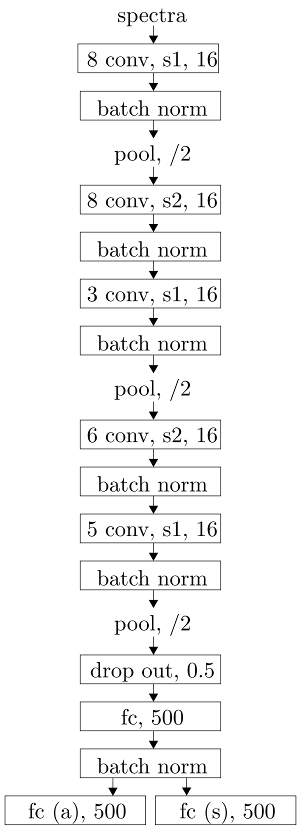


Figure 3.5: Graphical diagram of the deep convolutional neural network developed for the prediction of mosquito age and species from spectra. The convolutional layers are written in the form n conv, sm, o, where n is the width of the filter, m is the stride size, and o is the dimension of the feature space.

3.3.2.2 Model Training

The model is trained in two different ways depending on the specific task. When considering the lab-reared data there is access to a large dataset and as such the model is initialised with random weights and trained from this initialisation. Conversely, when considering semi-field or wild data the dataset is small. Therefore, transfer learning is used. For this, the model is initialised with the weights from a pre-trained model, the convolutional layers weights are frozen, and the model is trained from this initialisation. Here, the pre-trained model is a model trained on the larger lab dataset. This allows for the features learned by the model on the lab data to be re-used and only the final layers in the model are fine-tuned to avoid over-fitting on the semi-field and wild datasets. The wild dataset is not only small, but also imbalanced and as such, a re-weighting is applied to the optimisation gradients during training to mimic a balanced dataset.

3.3.3 Experiments

Unless otherwise stated, the model was trained using datasets balanced across mosquito age-groups and species. Further, in each experiment the dataset was split such that there is a 10% testing dataset and the remaining 90% of the dataset was used for optimising the model through 10-fold cross validation. Further, the spectra were standardised such that they are centered around the global mean and scaled to unit variance.

3.3.3.1 Lab Reared Data

The first experiments test whether the model can learn to correctly classify the age-groups and species of mosquitoes from their MIR spectra. Initially the laboratory variation (LV) dataset is used to assess how well the model can predict lab reared data when trained on lab reared data. Figure 3.6 (a) and (b) demonstrate that the model can predict both age-groups and species of LV data with a high level of accuracy achieving 84% and 93% accuracy in classifying age-group and species respectively. In comparison we trained a fully connected network on the same dataset and Figure 3.7 (a) and (b) show that this model achieved 74% and 87% accuracy in classifying age-group and species respectively. Secondly, both the LV and genetic variation (GV) datasets are used, where LV data is only utilised in the training dataset while the GV dataset is split into training and testing datasets. This assesses how well the model can predict lab reared data with genetic variability. Figure 3.6 (c) and (d) demonstrate that the model can again predict both age-groups and species of lab reared data with a high level of accuracy achieving 89% and 95% accuracy in classifying age-group and species respectively from the GV dataset. In comparison we trained a fully connected network on the same training and testing datasets and Figure 3.7 (c) and (d) show that this model achieved 82% and 84% accuracy in classifying age-group and species respectively. These results demonstrate that the convolutional model can be trained to accurately predict both age-groups and species of lab reared mosquitoes. While the full-connected model also achieves a high level of accuracy, in both instances it under-performs in prediction accuracy when compared to the convolutional model. This demonstrates that the convolutional layers are beneficial for the prediction of mosquito age and species. Furthermore, both of these models outperform which hand select specific wavenumbers to use as input features and logistic regression to predict age and species achieving 53% and 83% accuracy respectively (Jiménez et al., 2019).

3.3.3.2 Semi-Field Data

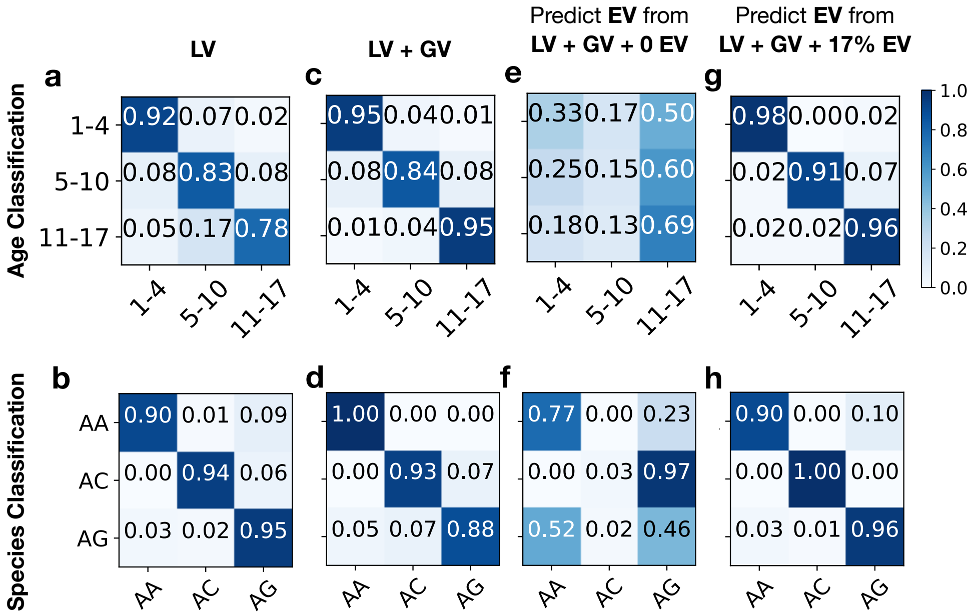


Figure 3.6: Confusion matrices of the convolutional model classification accuracy. (a-b) A model trained on LV data and tested on LV data. (c-d) A model trained on LV+GV data and tested on GV data. (e-f) A model trained on LV+GV data and tested on EV data. (g-h) A model initially trained on LV+GV data, re-trained using transfer learning with EV data (here, 1452 examples), and tested on EV data.

The ability to predict age and species of semi-field or wild mosquitoes with a sufficiently high degree of accuracy, such as to enable population monitoring, which is required to assess the effectiveness of malaria control measures, has been previously unachievable (Siria et al., 2022). The ability of the model to generalise to new datasets is assessed by testing the model trained on lab reared LV and GV data on mosquitoes reared in a semi-field through the use of the environmental variation (EV) dataset. Figure 3.6 (e) and (f) demonstrates that the model trained on LV and GV data does not generalise well to new data from a different source, such as the EV dataset. Given the strong performance of the model in predicting both LV and GV datasets test sets with high accuracy, it is most likely that the drop in performance when testing on the EV dataset is due to distinct differences in the data. Several hypotheses can be drawn, as to why the data is different. Some of these include regular access to a food source in the lab versus a natural food source in the semi-field; a climate-controlled environment in the lab versus a naturally varying climate in the semi-field; and a smaller space without other objects in the lab versus a larger space and natural objects to interact with, such as a hut and animals, in the semi-field. Furthermore, we also assessed the ability for the fully-connected model to generalise to the EV dataset from a model trained only on LV and GV data. Figure 3.7 (e) and (f) demonstrates that the fully connected model does not generalise well to the EV dataset either. This further validates that it is not simply that the convolutional model generalises poorly, but that the datasets are different.

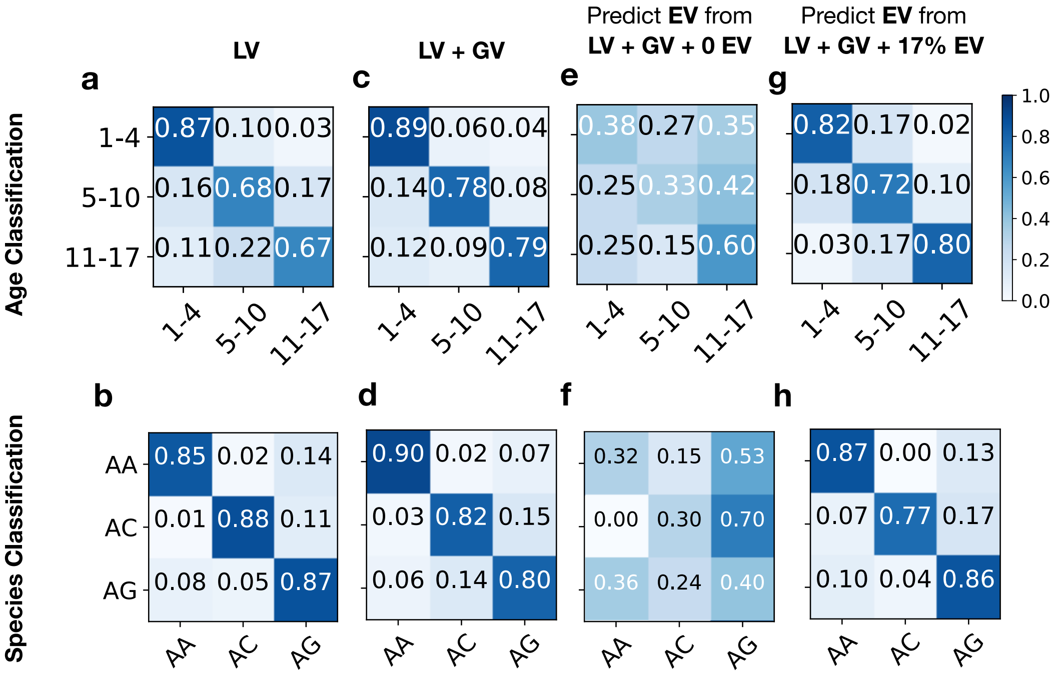


Figure 3.7: Confusion matrices of the fully connected model classification accuracy. (a-b) A model trained on LV data and tested on LV data. (c-d) A model trained on LV+GV data and tested on GV data. (e-f) A model trained on LV+GV data and tested on EV data. (g-h) A model initially trained on LV+GV data, re-trained using transfer learning with EV data (here, 1452 examples), and tested on EV data.

To enable the model to classify semi-field mosquitoes we utilise transfer learning using the model pre-trained on lab data. This allows the model to make use of the features learned from the lab data and reduces the quantity of training data required from the semi-field. Figure 3.6 (g) and (h) demonstrates that with 1452 training data points from the EV dataset used to re-train the model it is able to classify both age-groups and species with 94% accuracy. Therefore, with relatively few mosquito samples collected from a new site, the model is able to be re-trained to make accurate predictions. We again repeated this experiment on the fully connected model, following the same training set-up except for here we froze the first fully connected layer and allowed the remaining layers to be trained. Figure 3.7 (g) and (h) demonstrates that with 1452 training data points from the EV dataset used to re-train the fully connected model it is able to classify age-group and species with 81% and 88% accuracy respectively. This highlights that the convolutional layers are beneficial when utilising transfer learning to generalise to a new small dataset.

In addition to the result using 1452 training data points, Figure 3.8 shows the impact on the test accuracy when using different size EV training datasets. This demonstrates that a high test accuracy can be achieved with a small number of training data points. Increasing the quantity of EV data in the training set caused the prediction accuracy to increase rapidly, already exceeding 80% accuracy with 324 training examples and exceeding 90% accuracy when over 815 EV data points were used for training. This is of practical relevance given the field collection effort of collecting mosquito samples. Further, for comparison, Figure 3.9 shows the same experiment performed with the fully connected model, which demonstrates that the convolutional model generalises better to the new dataset with fewer training samples.

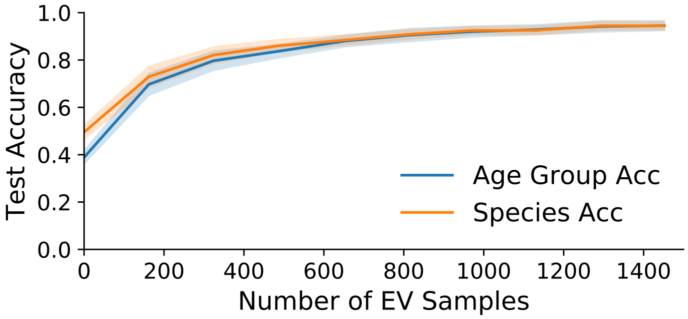


Figure 3.8: Classification accuracy improved from ∼50% to 94% for both age group and species with a training set comprising 0 (i.e. effects of increasing sampling of lab-reared mosquitoes only) through 1452 semi-field (EV) mosquitoes used to re-train the transfer learned model. The solid and shaded lines indicate the mean and standard deviation of the mean of 20 trained models, respectively.

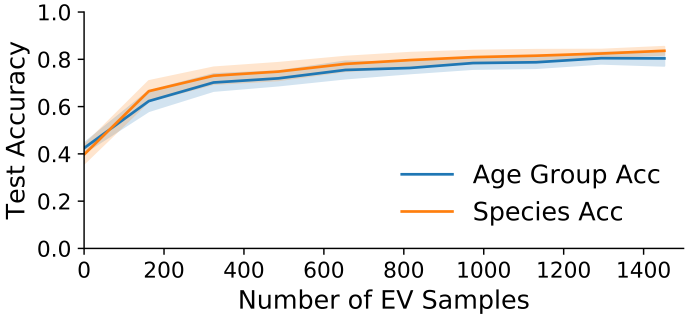


Figure 3.9: Classification accuracy improved from ∼40% to 81% and 88% for age group and species respectively with a training set comprising 0 (i.e. effects of increasing sampling of lab-reared mosquitoes only) through 1452 semi-field (EV) mosquitoes used to re-train the transfer learned model. The solid and shaded lines indicate the mean and standard deviation of the mean of 20 trained models, respectively.

3.3.3.3 Model Sensitivity

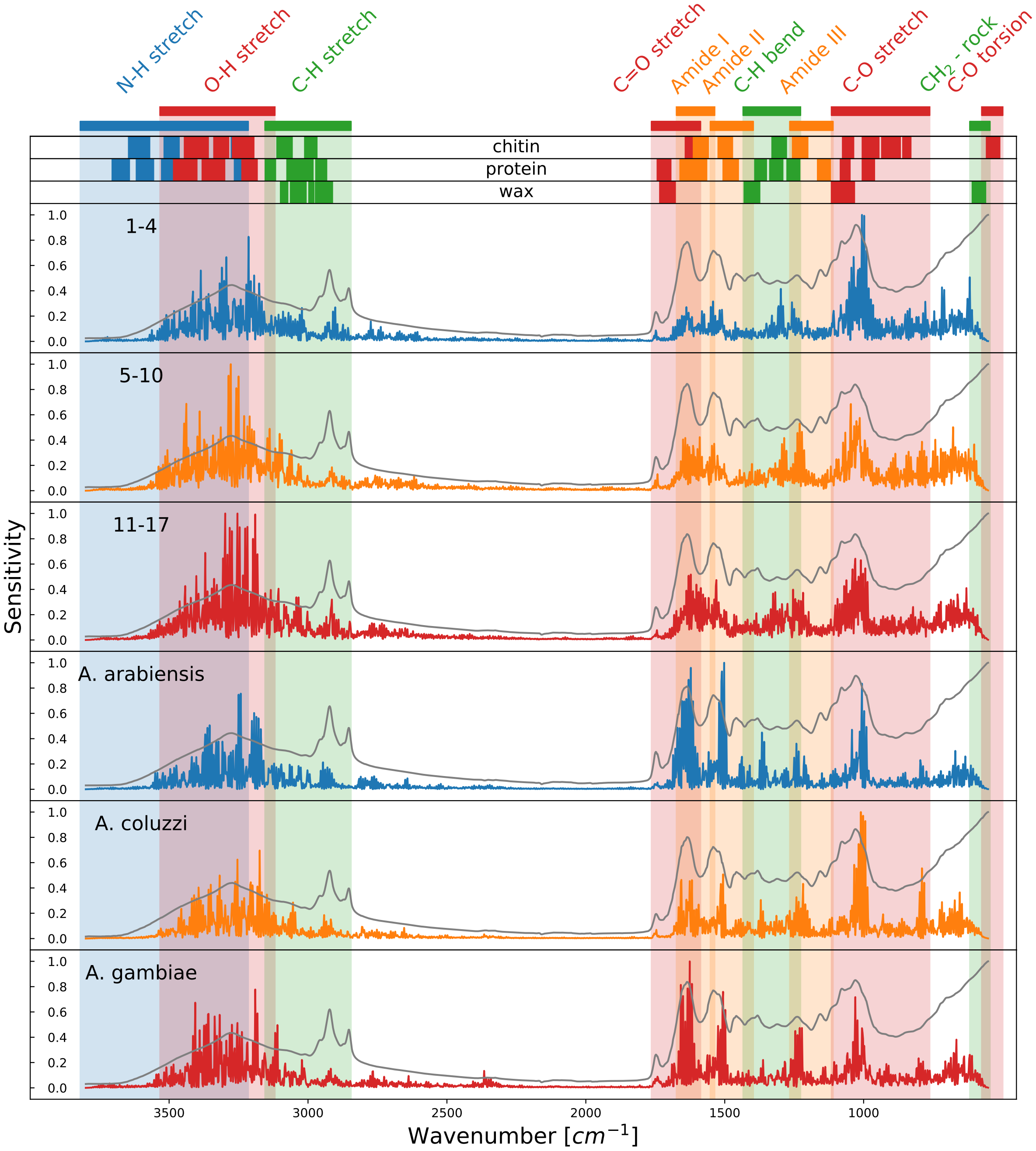


Figure 3.10: Average model sensitivities to different wavenumber values and comparison with the features of the average absorption spectrum (grey line) of each output class. The coloured stripes show the regions associated with the particular vibration of a functional chemical group. The upper part (maxima) displays the intervals of wavenumber values in which the maximum of the absorption peaks of each vibration appear for each of the three most abundant components in the cuticle of a mosquito (Jiménez et al., 2019). Here, the vibration of the same bonds appears in different wavenumber values depending on which cuticular component they belong to (chitin, protein or wax), which modifies the shape of the peaks.

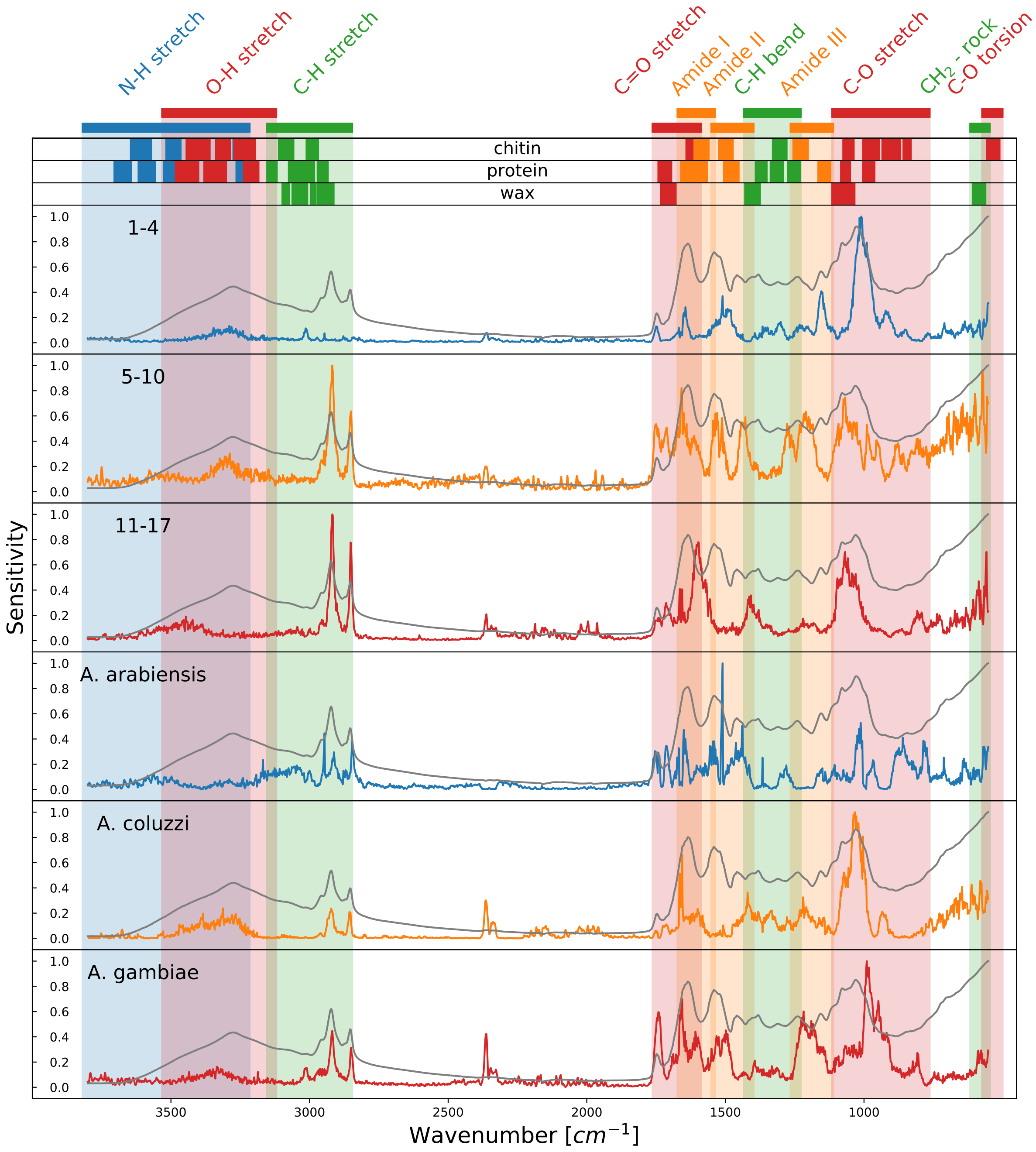


Figure 3.11: Average model sensitivities to different wavenumber values and comparison with the features of the average absorption spectrum (grey line) of each output class. The coloured stripes show the regions associated with the particular vibration of a functional chemical group. The upper part (maxima) displays the intervals of wavenumber values in which the maximum of the absorption peaks of each vibration appear for each of the three most abundant components in the cuticle of a mosquito (Jiménez et al., 2019). Here, the vibration of the same bonds appears in different wavenumber values depending on which cuticular component they belong to (chitin, protein or wax), which modifies the shape of the peaks.

Figure 3.10 shows the sensitivity of the convolutional model, which was trained for predicting both LV and GV datasets, to understand the regions in the MIR spectra that are most informative of mosquito age and species. The approach taken here follows that outlined by Selvaraju et al. (2017). Firstly, the gradient of the model inputs with respect to the model class output of interest is found. The gradients that act negatively towards the class prediction are then zeroed, such that only wavenumbers in the spectra that are leading to a positive classification of that class are retained. This is done as we are interested in finding the regions in the spectra that are causing the model to predict that samples are of this class, rather than those causing the model to predict that samples are not of a particular class. This is repeated for each input spectrum and the gradients are averaged. This is repeated for each age group and species class, which enables the sensitivity regions in the spectra to be assessed for each age group and species.

The sensitivity profiles indicate that the model extracted key biochemical features present in the spectra, corresponding more specifically to wavenumber values associated with the vibration of chitin and protein vibrations as is shown by the vertical colored bands in Figure 3.10. Furthermore, the aliphatic hydrocarbon bands (green stripes) contributed little to the model, suggesting that lipids like wax in the cuticle are less informative in distinguishing age and species of mosquitoes. The sensitivity plot has high frequency fluctuations in sensitivity, which we believe is a similar phenomena to that utilised in the area of adversarial attacks on neural networks. Commonly when talking about adversarial attacks on neural networks this is in reference to convolutional neural networks trained on images. Adversarial attacks often involve injecting small amounts of high frequency information into an image, which is unnoticeable to humans, that trick the network into mis-classifying the image (Szegedy et al., 2014; Goodfellow et al., 2014). This implies that convolutional neural networks are sensitive to high frequency information, which is demonstrated by our network as it is sensitive to high frequency information within the spectra. Enforcing smoothness in the convolutional kernels can overcome this, making the model less sensitive to high frequency information, although Wang et al. (2020) showed that this comes at cost to the classification accuracy of the model. As a result, the high frequency nature of the sensitivity plot shows that the convolutional model is extracting high frequency information from the spectra. Given that the user of such a model is also the person interested in the results, it seems counter intuitive that the model would be used in a situation where adversarial attacks can lead to incorrect classifications. Therefore, the model learning high frequency information is not concerning and it would not be advantageous to lower classification performance to stop this behaviour.

The same sensitivity analysis was repeated on the fully connected model. Figure 3.11 shows that the fully connected model also has some sensitivity to regions within the spectra that correspond to key biochemical regions. Furthermore, the model is sensitive to the aliphatic hydrocarbon bands (green stripes), suggesting that the model is using information from changes in lipids like wax in the cuticle to distinguish age and species of mosquitoes. Although, the fully connected model is also sensitive to regions which do not align with the vibration of any bonds that exist in the mosquito. One example of this is the peak in sensitivity around the $2350\mathrm{cm}^{-1}$ region, which corresponds to CO_2, and should not be a good indicator of mosquito age and species. This region in the spectra is most likely an indicator of differences in the processing of mosquito samples and the operator of the mid-infrared spectrometer. The sensitivity of the model to these regions, which are void of meaningful information, has two possible explanations:

- The model is using this region as a form of calibration to adjust for offsets in the spectra.
- The model is over fitting to the data and has learned some feature in these regions that can more easily classify the data.

Given that we expected the useful information within the spectra to be the peaks and troughs associated with key biochemical regions, this is most likely an indicator that the model is over fitting. This assumption is further backed up by the fully connected model requiring more EV training examples to generalise in Figure 3.9 than the convolutional model in Figure 3.8.

When comparing the sensitivity plots from both models we notice that both models are sensitive to the C-O stretch region, which could be a strong indicator that this region features useful information for determining mosquito age and species. In addition, both models are sensitive to the C=O strectch, Amide I, and Amide II regions, which suggests that these regions are also useful indicators of age and species.

3.3.3.4 Wild Data

The model was then evaluated on the ability to predict the age of wild mosquito populations. The wild data differs from the LV, GV, and EV datasets in that we have a small dataset of mosquitoes which have been dissected to determine the age and another dataset of mosquitoes which the age is unknown. This is due to the cost of dissecting mosquitoes making it only possible to have access to one training dataset of known age. This situation is representative of a real situation in which a small dataset of mosquitoes could be dissected to be used for transfer learning of the model trained on lab data.

Similar to the approach for EV data, a model is trained using transfer learning from the model pre-trained on lab data as the base model. The target age classes for the wild data are the number gonotrophic cycles. Although the convolutional layers were trained to predict chronological age classes, it is expected that these share many features of wild mosquitoes classified into gonotrophic cycles. Indeed, the three classes of 1–4, 5–10 and $\geq$ 11 days old correspond to females that underwent 0, 1 or $\geq$ 2 gonotrophic cycles (Detinova et al., 1962). Separate models were trained with 335 wild mosquitoes collected and dissected in Burkina Faso and 758 from Tanzania.

It is not possible to assess the classification accuracy of the age groups as was done for the lab and semi-field data due to the true age groups not being known for wild mosquitoes. Instead, predictions are made by the model for the age groups of each sample from the wild test dataset. Next the predicted distributions over the age groups are averaged over the samples. This is repeated for 10 models, each trained on the wild data, to assess the variability predicted age distribution. Figure 3.12 shows that the models predicted very similar age structures for the non-dissected test dataset and the dissected (morphologically assessed) training dataset. This suggests that the model can be readily adapted to diverse field settings and ageing methodologies.

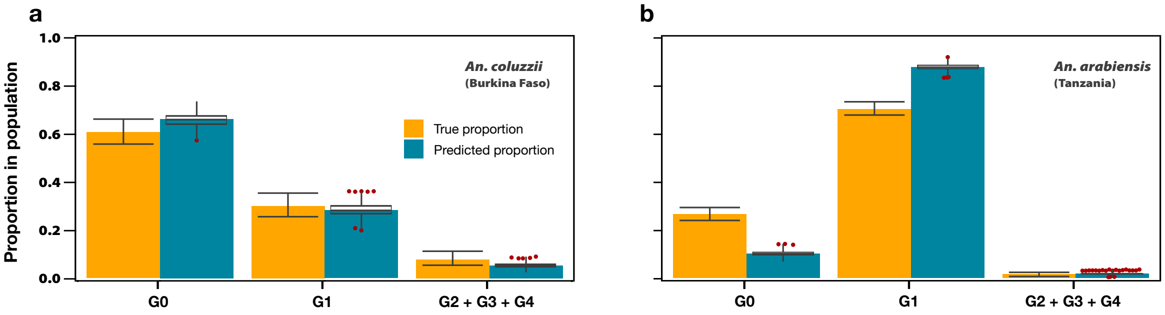


Figure 3.12: Testing on wild mosquito populations. The proportion of wild female mosquitoes with 0, 1 or $\geq$ 2 gonotrophic cycles (G0, G1, G2+G3+G4) was determined by ovarian dissection and morphological characterisation (yellow). The predictions made by the model on non-dissected mosquitoes (blue). The mean proportions and 95% credible intervals of the age proportion from dissected mosquitoes (yellow) were estimated with a Dirichlet distribution and provided in (Siria et al., 2022). The age proportion predicted by the model (blue) is presented as box and whisker plots showing the median, interquartile range (IQR, box), lowest/highest data within 1.5 IQR (whiskers), and outliers (red points) of the probability distribution of predictions from ten different models.

3.3.3.5 Discussion

The results show that considering translation equivariance for high-dimensional spectra data is a promising direction given the improved performance over an MLP model. Further, the sensitivity analysis of the two types of models demonstrates that the convolutional based model is more sensitivity to high frequency information in the spectra. This suggests that capturing higher resolution spectra could be a promising direction. The results, especially on the wild data, shows that this approach could be used to determine the age structure of mosquito populations. This could therefore be used as a tool in determining the success rate of mosquito interventions.

3.4 Conclusions

A novel model is presented for the classification of age groups and species of mosquitoes. This can not only be utilised for predicting age and species of mosquitoes reared in the lab, but also for wild mosquitoes, overcoming the challenge of working with very small unbalanced data through the use of transfer learning. This demonstrates that translation equivariant 1D convolutions are a useful tool for learning on high-dimensional mid-infrared spectra. We demonstrate that the developed model outperforms previous methods in the prediction of age and species of lab reared mosquitoes. Further, the trained model is sensitive to regions within the input spectra that correspond to regions that have chemical relevance, suggesting that the model is learning useful features. We also demonstrate that transfer learning can be used as a technique to train a model that can predict age groups and species of semi-field mosquitoes with high levels of accuracy while requiring a small quantity of data. In addition, transfer learning can be used to train a model that predicts an age structure similar to what is seen in the wild with a small quantity of data. Finally, the proposed model can be used to assist malaria mosquito surveillance and will hopefully assist in assessing the effectiveness of malaria control interventions.

4

Rotational Symmetries on 2D Grids

In this chapter we develop two different models, both with rotational symmetries on image domains. The first is a steerable convolutional neural network with symmetries given by the discrete rotational group C_8 for the application of segmentation of deforestation regions. The second is novel model with cylindrical symmetries given by the group $\mathrm{SO}^+(2,1)$. This model is developed for the task of inverting transmission effects of multi-mode optical fibres.

4.1 Introduction

In this chapter we construct a steerable convolutional neural networks for discrete rotation groups in Section 4.3. This consists of developing a U-Net style model for both classification and segmentation of deforestation regions from satellite images. This presents a clear example of a rotation equivariant architecture for an application where rotational transformations naturally occur. The improved classification and segmentation accuracy demonstrates the benefits of such an approach. Finally, the stability of the predicted segmentation maps is demonstrated, which is highly desirable for the given application.

In addition, we develop a model that correctly models the cylindrical symmetries of imaging through an optical fibre in Section 4.4. This work overcomes the challenges of translating the terminology between physics and equivariant neural networks to compose a novel architecture that correctly models the cylindrical symmetries of the task by developing a network with $\mathrm{SO}^+(2,1)$. This overcomes the non-local relationship between images and their corresponding speckled patterns through incorporating a useful inductive bias into the model, namely correctly modelling the cylindrical symmetries of the imaging task. This non-local relationship presents a challenge for prior works such as fully convolutional based methods, as they assume the inductive bias of locality where it does not exist. We combine this model with a convolutional post-processing model, which we demonstrate makes our method more interpretable than prior works. Despite this, we also demonstrate the method is applicable to real-world data through the use of a dataset collected in the laboratory, and show how our method can be adapted to overcome deviations between theory and the physical world. Furthermore, we explore how our method performs under a range of experiments, demonstrating improved robustness to the training dataset size, noise in the speckled images, and the parameterisation of the Bessel basis set inside the model. Finally, we demonstrate that our model can scale to higher resolution images that had previously been possible, which could unlock the use of imaging through multi-mode fibres for new applications.

4.2 Background

The first works incorporating rotational equivariance for images selected a discrete rotation group such as the group $p4$ or $p4m$ consisting of rotations by 90° and with or without mirror reflections (Cohen & Welling, 2016; Veeling et al., 2018). Such models replace the convolution operator used in a typical Convolutional Neural Network (CNN) with a group-convolution. Therefore, instead of performing convolution over an image domain, group-convolution is performed over a group. Convolution and cross-correlation are often used interchangeably in deep learning texts. As such we detail the different between convolution and group-convolution as cross-correlations as this is the typical implementation used in deep neural networks. Given that cross-correlation is typically performed on the image domain $\Omega = \mathbb{Z}^2$ with an input signal $f = \mathcal{X}(\Omega) : \mathbb{Z}^2 \to \mathbb{R}$ and filter $\psi : \mathbb{Z}^2 \to \mathbb{R}$, it can be defined as:

$$[f \star \psi](x) = \sum_{y \in \mathbb{Z}^2} f(y)\psi(y - x). \tag{4.1}$$

Cross-correlation is equivariant to translation group actions. This can be demonstrated by showing that a translation action applied to the input signal f is equivalent to a translation action applied to the output signal:

$$\begin{aligned}\left[[L_t f] \star \psi\right](x) &= \sum_{y \in \mathbb{Z}^2} f(y-t)\psi(y-x) \\ &= \sum_{y \in \mathbb{Z}^2} f(y)\psi(y+t-x) \\ &= \sum_{y \in \mathbb{Z}^2} f(y)\psi(y-(x-t)) \\ &= \left[L_t\left[f\right] \star \psi\right](x).\end{aligned} \tag{4.2}$$

Then group cross-correlation can be considered as considering a larger set of transformations, where these transformations have a group structure. As such, group cross-correlation can be defined as

$$[f \star \psi](g) = \sum_{y \in \mathbb{Z}^2} f(y)\psi(g^{-1}y). \tag{4.3}$$

This group cross-correlation endows each of the $|G|$ feature channel to a different element $g \in G$. A single layer then has $n \times |G|$ features where n would be chosen as is typically chosen in a regular CNN. For such small rotational groups the equivariance required for the network can be simply implemented by stacking rotated and reflected versions of the kernel typically used in a regular CNN.

An important step in the development of rotation equivariant convolutional neural networks was that of steerable CNNs (Cohen & Welling, 2017; Cohen et al., 2018, 2019; Weiler & Cesa, 2019). Steerable CNNs describe $\mathrm{E}(2)$-equivariant convolutions on the images domain $\mathbb{R}^2$, where $\mathrm{E}(2)$ is the group of rotations and reflections of the plane $\mathbb{R}^2$. The feature spaces of steerable CNNs are vector or scalar spaces, where a group representations ρ determines the transformation behaviour of the feature space under transformation of the input. To guarantee the transformation behaviour of the feature fields the convolutional kernels are constrained, depending on the group representation used. A general approach for constructing such a network was provided by Weiler & Cesa (2019) through the use of irreducible representations of the group.

Steerable CNNs define the feature space of the input signal as $f : \mathbb{R}^2 \rightarrow \mathbb{R}^c$, which associates a c-dimensional feature vector $f(x) \in \mathbb{R}^c$ to each point x in the base space $\mathbb{R}^2$. For a scalar feature field $\mathbb{R}^c = \mathbb{R}^1$, which associate the trivial representation, $\rho(g) = 1 \forall g \in G$, to the feature space detailing how the features transform under a group action. In the case of the trivial representation this amount to multiplication by the identity; hence, the features move to a new position, but do not change orientation. While for a vector feature field the group representation associated to the feature space details how it transforms under a group action. That is, an input vector feature space is transformed as

$$v(x) \longmapsto \rho(g) \cdot v(g^{-1}(x-t)), \tag{4.4}$$

where the input is not only moved to a new position, but also changes orientation via the group action $g \in G$. Each layer in the network of a steerable CNN is required to be equivariant so that the transformation law of the feature spaces is preserved. Therefore the cross-correlation is performed with a G-steerable kernel, which for input and output feature space transformation laws given by ρ_{in} and ρ_{out} is required to satisfy the kernel constraint (Weiler & Cesa, 2019)

$$\phi(gx) = \rho_{\text{out}}(g)\phi(x)\rho_{\text{in}}(g) \;\; \forall g \in G, x \in \mathbb{R}^2. \tag{4.5}$$

The concept of steerable CNNs encompasses the group convolutions described above, where a discrete rotation group is used. Although, in addition, steerable CNNs describe rotation equivariant CNNS which are equivariant to group actions of the continuous rotation group $\mathrm{SO}(2)$. In the case of discrete rotation groups, a different feature space is associated to each rotation angle. Storing features for an infinite number of rotation angles is not computationally tractable. In Equation 4.5 the kernel constraint is given to ensure that a network layer utilising steerable convolutions is equivariant, which features the group representations ρ_{out} and ρ_{in}. This constraint is solved by Weiler & Cesa (2019) by decomposing the constraint in terms of the irreps of ρ_{out} and ρ_{in}, where the irrep decomposition is given by

$$\rho = Q^{-1} \left[\bigoplus_{i \in I} \psi_i \right] Q, \tag{4.6}$$

where Q is a change of basis matrix, $\{\psi_i\}$ are the irreps of G, and I is an index set encoding the types and multiplicities of irreps in ρ.

As the groups considered for ensuring rotation equivariant on images are all subgroups of $\mathrm{O}(2)$, their action on $\mathbb{R}^2$ are all norm preserving. As a result the decomposition into irreps does not constrain the radial component, only the angular component. In addition, each irrep of such a group is always associated to a unique angular frequency. Therefore, the kernel constraint can be expressed in terms of an angular Fourier series, where instead of choosing a discrete number of rotations a maximum frequency can be chosen. Solving for the kernel space of permissible filters that can be used within a steerable CNN and ensuring that the model maintains rotation equivariance yields only the spectrally localised circular harmonics (Worrall et al., 2017; Weiler & Cesa, 2019; Franzen & Wand, 2021). Therefore a rotation equivariant CNN can be created by solving for the circular harmonics up to a certain rotational frequency and combining this with a radial profile function and sampling a basis of resolution given by the CNN filter size. This yields a set of bases which can be linearly combined with a learnable weighting applied to each to form the kernel for the CNN.

We present an example of a steerable CNN on a domain where rotational transformations naturally occur in Section 4.3. Here we follow the concept of steerable CNNs using a discrete rotation group.

Of the works that incorporate rotation equivariance into CNNs, we are particularly interested in those using a continuous rotation group $\mathrm{SO}(2)$ in Section 4.4. Although, in Section 4.4 we are not interested in developing a group equivariant convolutional model due to the non-similar spatial arrangements between the speckled and original image domains. Despite this, the concept of equivariance still applies, as the concept of learning a model from a fixed basis set, which guarantees symmetry properties are conserved, is relevant due to the nature of transmission through multi-mode optical fibres.

4.3 Rotation Equivariant Deforestation Segmentation

In this section we develop a rotation equivariant U-Net model for the segmentation of deforestation regions from satellite images. Given that fewer examples of rotation equivariant models exists, when compared to convolutional neural networks, we hope this example will make it easier for such a model to be utilises for new applications. This model predicts stable segmentation maps under rotations, which is a natural transformation of the input images, unlike non-rotation equivariant models. Further, it improves the accuracy of segmentation map prediction and deforestation drive classification. Therefore, our contributions are:

- An example of a rotation equivariant network for an application where the input images have no fixed orientation.
- A demonstration of the practical benefits of correctly considering the symmetries of a given problem.

4.3.1 Background

Deforestation has been greatly accelerated by human activities with many drivers leading to a loss of forest area. Deforestation has a negative impact on natural ecosystems, biodiversity, and climate change and it is becoming a force of global importance (Foley et al., 2005). Palm plantation deforestation is projected to contribute 18-22% CO_2-equivalent emissions in Indonesia (Carlson et al., 2013), which is the leading producer of palm oil (Carlson et al., 2012). Furthermore, deforestation in the tropics contributes roughly 10% of annual global greenhouse gas emissions (Arneth et al., 2019). In addition to the emissions caused by deforestation, over one quarter of global forest loss is due to deforestation with the land being permanently changed to be used for the production of commodities, including beef, soy, palm oil, and wood fiber (Curtis et al., 2018). "Climate tipping points are when a small change in forcing, triggers a strongly nonlinear response in the internal dynamics of part of the climate system" (Lenton, 2011). Deforestation is one of the contributors that can cause climate tipping points (Lenton, 2011). Therefore, understanding the drivers for deforestation is of significant importance for preventing climate tipping points.

The availability and advances in high-resolution satellite imaging have enabled applications in mapping to develop at scale (Roy et al., 2014; Verpoorter et al., 2012, 2014; Janowicz et al., 2020; Karpatne et al., 2018). A range of prior works have used decision trees, random forest classifiers, and convolutional neural networks for the task of classifying and mapping deforestation drivers from satellite images (Phiri et al., 2019; Descals et al., 2019; Poortinga et al., 2019; Hethcoat et al., 2019; Sylvain et al., 2019; Irvin et al., 2020). However none of these previous methods leverage advances in rotation equivariant convolutional networks (Cohen & Welling, 2016, 2017; Weiler & Cesa, 2019), and as such the methods are not stable under rotation of the input images. This is a significant weakness as rotational transformation can naturally occur during the capture of such data. As a result the segmentation regions predicted by previous methods are not independent from the choice of orientation of the image. Therefore, the region of deforestation predicted will be different depending on the orientation of the satellite image, which makes the use of these methods unreliable when used in practise.

4.3.2 Model Specification

To overcome the limitations of prior works, namely their lack of independence from the choice of orientation of the input image we develop a rotation equivariant model. Due to the recent emergence of equivariant neural networks (Cohen & Welling, 2016), the availability of already implemented models as well as pre-trained models is scarcer than non-group equivariant model. Therefore, we develop a U-Net (Ronneberger et al., 2015) style architecture for the task of segmentation utilising the e2cnn package Weiler & Cesa (2019) and attach an MLP to the lowest dimensional feature space for classification. We utilise two models to assess the benefits of incorporating rotational equivariance for the task of deforestation prediction:

- A translation equivariant convolutional model.
- A translation-rotation equivariant convolutional model.

The first model uses translation equivariant convolutional layers and is not equivariant under rotations, while the second model uses translation-rotation equivariant convolutional layers and is therefore rotation equivariant. For the rotation equivariant version, we choose the group C_8 of discrete rotations by $45°$ as the symmetry group. Therefore, this model has a guaranteed transformation behavior for $45°$ rotations of the input image. We chose to consider the equivariance group C_8 as the lower the choice of N in the group C_N, the lower the memory requirement is for the model for a fixed feature dimension. This is due to the implementation of equivariance to the cyclic group consisting of N rotated copies of the feature fields.

The input to the model is RGB images, which consists of three independent 1-dimensional vector spaces on the domain $\mathbb{R}^2$. Therefore, the input feature space consists of the direct sum of three trivial representations, which associates three independent 1-dimensional feature vectors $f(x) \in \mathbb{R}$ to each point x of the base space. The trivial representation is characterised by the group representation $\rho(g) = 1 \; \forall g \in C_8$. Therefore the feature space is defined as $f = \bigoplus_i^3 f_i$ feature fields which transform under the direct sum $\rho = \bigoplus_i^3 \rho_i$, where each ρ_i is the trivial representation. Each hidden layer in the model has a feature space that consists of multiple regular representations of the group C_8. Therefore, the feature space of hidden layers is defined as $f = \bigoplus_i^m f_i$, for $f_i : \mathbb{R}^2 \to \mathbb{R}^8$, where m is the choice of number of features commonly used in a neural network. The structure of each feature space is fully characterised by the group representation $\rho_{\text{reg}}^{C_8}$, with the group representation characterising the transformation law of the the hidden layers given as the direct sum of individual representations $\rho = \bigoplus_i^m \rho_{\text{reg}}^{C_8}$. Finally, the output of the model is a prediction of the segmentation map. This is a single 1-dimensional vector with transformation law characterised by a single trivial representation.

We chose the dimension m of the feature space of the rotation equivariant network such that the model has a similar size of feature space to the non-rotation equivariant model. The overall architecture of the model is details in Figure 4.1.

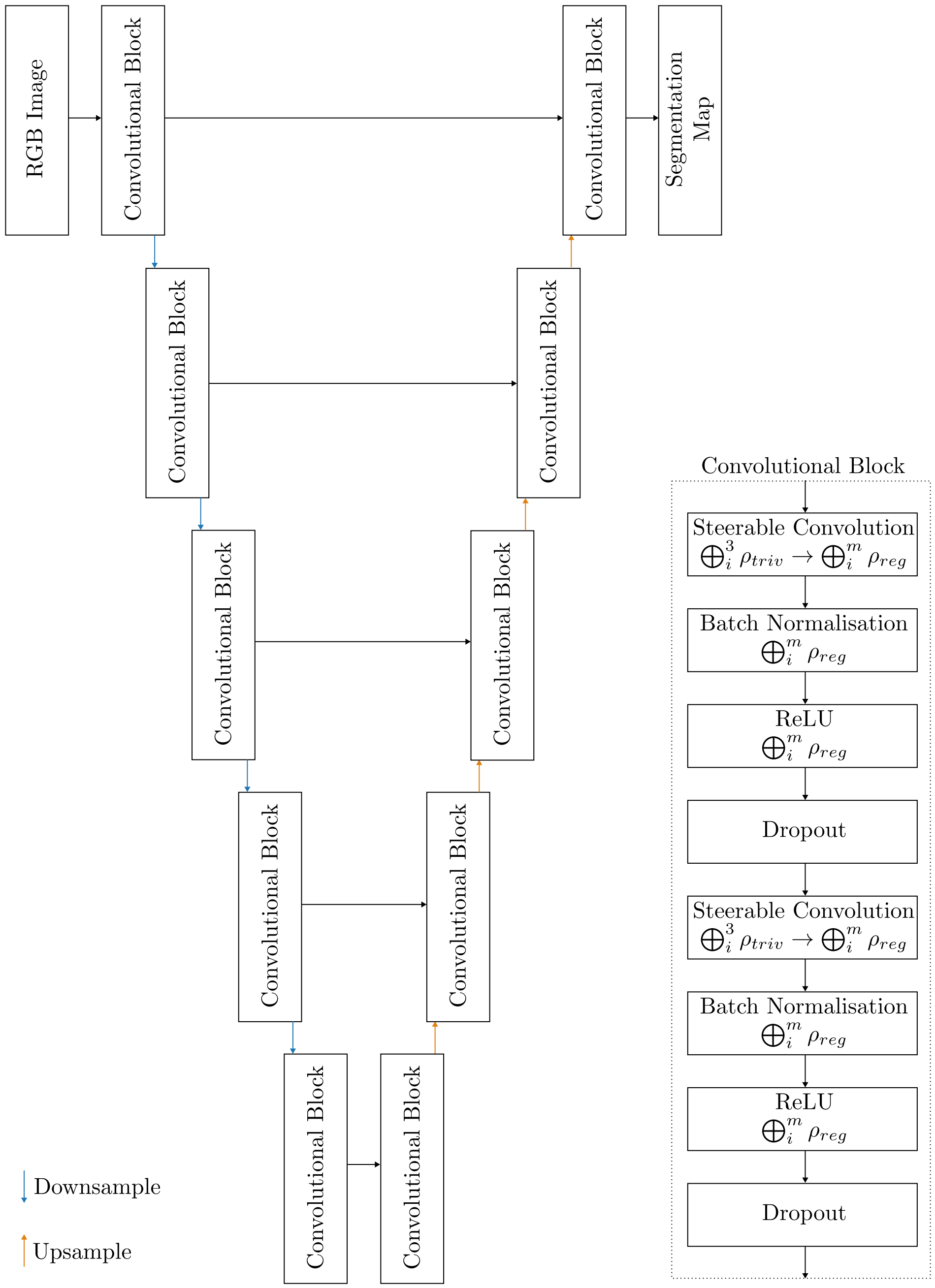


Figure 4.1: A breakdown of the rotation equivariant model, which is a U-Net style model with the convolutional blocks replaced with rotation equivariant ones. The model consists of five convolutional blocks with down sampling in between each, followed by five convolutional blocks with up sampling in between each. In addition, each convolutional block in the down sampling region has a skip connection connecting it with the convolutional block in the up sampling region, which processes inputs of the same resolution. A breakdown of the convolutional block is also given in the lower right hand side of the figure.

4.3.3 Experiments

The dataset used is the same as that used by Irvin et al. (2020), where forest loss event coordinates and driver annotations were curated by (Austin et al., 2019). Random samples of primary natural forest loss events were obtained from maps published by Global Forest Change (GFC) at 30m resolution from 2001 to 2016. These images were annotated by an expert interpreter (Austin et al., 2019) to determine deforestation regions and deforestation drivers. The drivers are grouped into categories determined feasible to identify using 15m resolution Landsat 8 imagery, while ensuring sufficient representation of each category in the dataset (Irvin et al., 2020). The mapping between expert labelled deforestation driver category and driver group used as a classification target is provided in Table 4.1. The dataset consists of 2,756 images, segmentation maps, and class labels. We follow the training/validation/testing set splits as provided by Irvin et al. (2020).

Table 4.1: The mapping between deforestation driver groups as defined in (Irvin et al., 2020) and the expert labelled deforestation driver categories defined in (Austin et al., 2019). The deforestation driver groups are used as classification targets when training models.

Expert Labelled Deforestation Driver Category	Classification Target Driver Group
Oil palm plantation Timber plantaion Other large-scale plantations	Plantation
Grassland/shrubland	Grassland/shrubland
Small-scale agriculture Small-scale mixed plantation Small-scale oil palm plantation	Smallholder agriculture
Mining Fish pond Logging road Secondary forest Other	Other

The prediction accuracy of the segmentation maps between the two models is compared in Table 4.2. This shows that the rotation equivariant model achieves better test segmentation accuracy. One cause of this benefit is that the model can share learned segmentation features across different orientations that occur across the different images in the dataset, while the non-rotation equivariant model has to learn repeated features for each orientation. The stability of the segmentation map prediction is also demonstrated by testing on a rotated version of the test dataset. The rotation equivariant network has an identical test accuracy on both the rotated and non-rotated datasets, demonstrating the stability of the segmentation maps under rotation. Surprisingly, the non-rotation equivariant model has a near identical test accuracy on the rotated and non-rotated datasets. This is most likely a result of the deforestation regions being predicted by the model being either too large or too small and therefore, despite the deforestation map not deforming smoothly under rotation, it does not impact the accuracy significantly.

Table 4.2: Comparison between a model with translation equivariant convolutions and a model with both translation and rotation equivariant convolutions. Results are displayed as percentages for the segmentation accuracy of per pixel prediction averaged between the true deforestation and non-deforestation areas to account for the class imbalance towards non-deforestation areas.

Model	Train	Validation	Test	Rotated Test
UNET - CNN	72.9	68.7	67.8	67.9
UNET - C8 Equivariant	**84.1**	**71.3**	**72.3**	**72.3**

The model trained with rotation equivariance also outperforms the non rotation equivariant model for classification of the drivers of deforestation, shown in Table 4.3. One reason that the rotation equivariant model achieves a higher classification accuracy is that the model is using the features more efficiently by not learning similar features at different orientations. As a result the model predicts more accurate segmentation maps, which it can then use to better classify the deforestation drivers.

Table 4.3: Comparison between a model with translation equivariant convolutions and a model with both translation and rotation equivariant convolutions. Results are displayed as percentages for the classification accuracy of driver of deforestation.

Model	Train	Validation	Test	Rotated Test
UNET - CNN	**90.3**	60.6	57.9	56.3
UNET - C8 Equivariant	82.7	**67.1**	**63.0**	**64.3**

The segmentation map predictions for the non rotation equivariant model and rotation equivariant models are shown to compare the stability of segmentation under rotation in Figure 4.2. This highlights that the segmentation map prediction for the non-rotation equivariant model changes as the image is rotated, which would be highly undesirable if used in practice as the rotation orientation of the satellite should not effect the segmentation map prediction of deforestation. On the other hand, the rotation equivariant model segmentation map prediction is stable under rotation, which is a highly desirable property of the model.

4.3.4 Discussion

The results demonstrate that when the data domain has a known symmetry, such as a rotational symmetry in this case, it is beneficial to make the model equivariant to this symmetry. This ensures that no undesired deformations exist in the model predictions

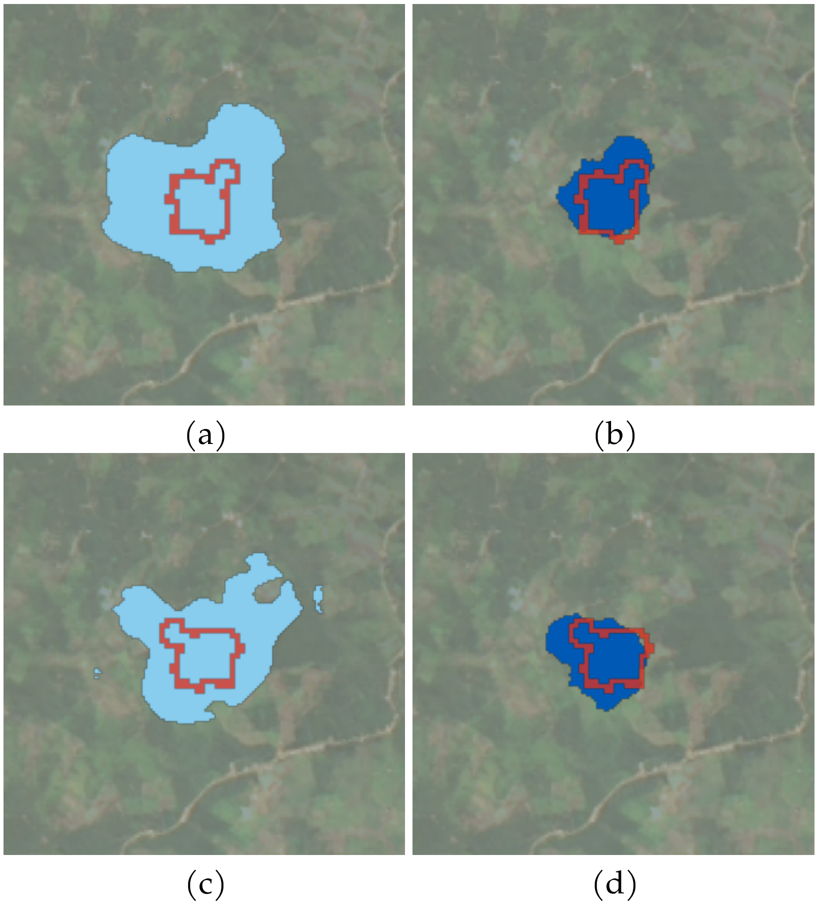


Figure 4.2: A comparison of predicted segmentation maps under rotation for both the non-rotation equivariant model and the rotation equivariant model. The original image is shown in (a) and (b) with the edge of the true segmentation map in red. Image (a) shows the predicted segmentation map for the non-rotation equivariant model in light blue. Image (b) shows the predicted segmentation map for the rotation equivariant model in dark blue. The $90°$ rotated image is shown in (c) and (d) with the edge of the true segmentation map in red. Image (c) shows the predicted segmentation map for the non-rotation equivariant model in light blue. Image (d) shows the predicted segmentation map for the rotation equivariant model in dark blue.

when a natural symmetry transform of the data occurs. Given that rotations of the images captured by satellites can occur, rotation equivariant networks present a promising direction for building segmentation models for satellite imagery.

4.4 Cylindrical Equivariant Inversion of Multi-Mode Fibre Transmission Effects

In this section we present a model which naturally accounts for the difference in spatial arrangement between speckled and original images and scales more efficiently than previous methods to higher resolution images. This is the first method to demonstrate an ability to invert 256×256 pixel speckled images into 256×256 pixel original images. Our approach also takes advantage of the circular correlations in the speckled images, and improves upon previous general imaging results. Concerning the equivariance literature, we develop a model comprising of cylindrical harmonic basis functions, a basis set which has seen little attention in the equivariance literature, and make the connection between the transmission of light through a fibre and the group theoretic understanding used in developing equivariant neural networks. In addition, to the equivariant component of our model we also introduce a post processing model, which is a convolutional model, which correctly operates on images in the same spatial arrangement to those being predicted. The separation of the model into two complementary components creates a more interpretable model than previous approaches. In the experimental section we introduce a new theoretical dataset which makes it possible to explore the predictive performance of the model on out-of-distribution datasets, with the existence of noise in the speckled patterns, with fewer training examples, and with an under parameterised bases space. Our contributions are:

1. A connection between group theoretic equivariant neural networks and the inversion of MMF transmission effects, providing a novel type of model to tackle the problem
2. A more interpretable model due to splitting the model into a physics motivated equivariant model and convolutional model
3. A more data-efficient model to solve the inversion of MMF transmission effects
4. A more scalable model to solve the inversion of MMF transmission effects, which can scale to previously unachievable resolutions
5. A model that provides better generalisation to out-of-training domain images

4.4.1 Background

4.4.1.1 Multi-Mode Fibres

Multi-mode fibres present a clear advantage over single-mode fibre bundles due to having 1-2 orders of magnitude greater density of modes than a fibre bundle (Choi et al., 2012). As a result, multi-mode fibres (MMF) have many potential applications in medical imaging, cryptography, and communications. In the medical domain, the use of multi-mode fibre imaging has potential to create hair-thin endoscopes for imaging sensitive areas of the body. However, to utilised for these applications, the fibre transmission properties must be compensated for to return a clear image (Stasio, 2017). This section concerns the use of a single multi-mode fibre and does not consider fibre bundles due to the advantages presented by multi-mode fibres.

A MMF has multiple different fibre modes, each of which propagates at a different velocity. This leads to an amplitude and phase mixing of the image as it propagates through the fibre (Mitschke, 2016). As a result, an input image creates a complex-valued speckled pattern on the output of the MMF. If each propagation mode and velocity was known the transmission matrix (TM) could be computed, providing a linear system that inverts the transmission effects, although in general this is not known. This information be found by acquiring the output amplitude and phase relative to each mode, but in practise this is time consuming and requires many measurements to be taken to fully characterise the TM. In addition, this is only applicable to the specific fibre in its current configuration and as such bending the fibre changes the TM and requires the entire process to be repeated.

Inverting a speckled image is challenging for multiple reasons. Firstly, the speckled images have a non-local relationship with respect to the original images. As a result, solely local patch-based models, such as convolutional neural networks, do not make sense as a solution without some dense mapping function. Therefore, the non-locality necessitates mapping the speckled images into a spatial arrangement similar to the original images before typical image-based deep learning techniques can be used, such as convolutions and pooling. In addition, the speckled images have circular correlation, which could be taken advantage of, although as noted by Moran et al. (2018), finding these requires solving the inversion, creating a chicken-and-egg problem. Further, the mode mixing interference during propagation of an image can lead to information loss such that inverting the transmission does not yield the true original input image, as is demonstrated in Figure 4.3. Finally, the fibre is equivalent to an unknown complex transmission matrix (TM) so, the inverse of this could be found using a complex-valued linear model, although this presents challenges in terms of memory requirements. A mapping between 350×350 original and speckled images would result in a TM with $350^4 \approx 15$ billion entries, requiring a linear model with as many parameters. We provide further details on the inversion of the TM in Section 4.4.1.2.

The ability to accurately, in a scalable way, learn to invert the transmission effects, would unlock MMF imaging as a useful tool across a range of domains. Although, a model which correctly models the physics of inverting the transmission effects should produce the inverted image in Figure 4.3 and producing the original image requires addition information to be learned. We provide further details into the propagation of light through optical fibres and how we construct theoretical TMs in Section 4.4.1.5.

4.4.1.2 Invertibility of TMs

Given a TM, which models how an image propagates through a fiber, it is possible to invert the matrix and get the corresponding mapping back from the speckled image space to the original image space. We provide details of the construction of theoretical TMs in Section 4.4.1.5. Utilising this we can inspect information loss due to the limited number of modes of the fibre by passing the image through the TM and back through the inverse to create an inverted image. Therefore, if we knew the TM this inverted image would be the recoverable information. We show some examples of the original image, the speckled image, and the inverted image in Figure 4.3.

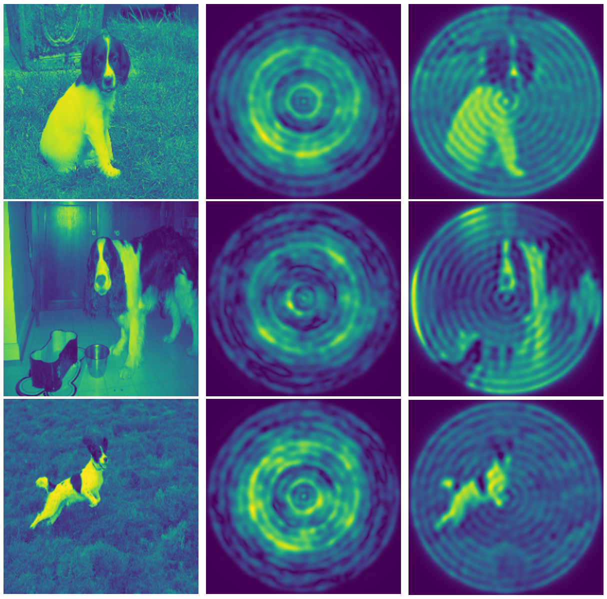
This is useful as it allows us to separate the task of inverting the transmission effects of a fibre into that which could be reconstructed by understanding the physics of the TM and that which requires generating due to being lost information. We believe this could therefore be viewed as two tasks:

1. A task of inverting the transmission effects which would have the aim of generating the inverted images
2. A task of predicting the original images from the output of the first task

4.4.1.3 Related Work

Previous work in inverting the transmission effects of MMFs can be categorised into three different approaches:

1. Characterisation of the TM through experimentation
2. Learning a dense linear model



(a) Original (b) Speckled (c) Inverted

Figure 4.3: (a) The original image, (b) the speckled image created by passing the original images through a theoretical TM, (c) the inverted image created by passing the speckled images through the inverted theoretical TM.

3. Learning a convolutional model

Method 1 requires extensive experimentation to characterise the TM of the fibre (Čižmár & Dholakia, 2011, 2012; Choi et al., 2012; Mahalati et al., 2013; Papadopoulos et al., 2012; Plöschner et al., 2015; Leite et al., 2021), where the number of experimental measurements required for re-calibration was reduced by Li et al. (2021) by exploiting sparstiy in the TM. The general approach taken in method 1 is to characterise the TM by acquiring the output amplitude and phase relative to each mode (Čižmár & Dholakia, 2011, 2012). Choi et al. (2012) present an approach to construct the TM in a scanner-free method based on measurement of amplitude and phase of the output. Although the method requires 500 measurement repetitions at different incidence angles. Mahalati et al. (2013) develop a method in which the number of resolvable image features approached four times the number of spatial modes. Papadopoulos et al. (2012) develop a digital phase conjugation technique to restore images without the requirement of calculating the full TM. Despite this, the solution of finding the TM, or part of it, has to be repeated for every different fibre, for every different length, under each different bending scenario, and for each different temperature, reducing the practicality of this solution. Plöschner et al. (2015) developed a procedure that could also incorporate bending through a precise characterisation of the fibre and a theoretical model.

Methods 2 and 3 are both machine learning based approaches to solve the inverse problem. Both of these approaches train a neural network to approximate the inverse transmission matrix and predict the original images from the speckled patterns. Method 2 consists of using dense linear models, which have no inductive bias (Moran et al., 2018; Fan et al., 2019; Caramazza et al., 2019). Moran et al. (2018) and Caramazza et al. (2019) approached solving the inverse problem through the use of either a Real and Complex linear model with a Hadamard layer to model the power drop off of the spatial light modulator. A dense linear model can account for the difference in spatial arrangement between speckled and original images and is therefore a theoretically sound approach to solving the task. Despite this, at its core, this approach requires the use of a fully connected model, which maps from speckled images to the original images and scales as $\mathcal{O}(N_s^2 N_o^2)$, where N_s is the resolution of the speckled image and N_o is the resolution of the original image. This is a very memory-expensive operation and restricts the scalability of the approach to relatively low resolution images. Furthermore, the high number of trainable parameters of a fully connected model, which has no inductive bias, could lead to the model overfitting to the specific training data.

Method 3 is to use a convolutional machine learning based approach (Borhani et al., 2018; Rahmani et al., 2018). In theory a fully convolutional neural network approach improves upon the scalability issue, through incorporating an inductive bias into the model that is commonly used for learning on images. Despite this, for the task of inverting the transmission effects of optical fibres, the speckled and original images do not have a local relationship. This is an issue for fully convolution based approaches as convolutions have an inductive bias of locality and the input and output data domains do not share the same spatial arrangement. Therefore, in practice, using a full convolutional approach requires the model to have a large number of layers in order to be able to effectively map every pixel in the speckled image to every pixel in the original image. As a result, in practice, fully convolutional approaches do not over come the scalability issues. Borhani et al. (2018) use a convolutional U-Net model to invert the transmission effects. To overcome the issue of mapping between two domains that are not in the same spatial arrangement their model requires 14 hidden layers to learn the inversion of 32×32 resolution images. Rahmani et al. (2018) also use a fully convolutional model, which comprises of 22 convolutional layers, for low resolution, 28×28, images. Therefore, it is not surprising the model can overcome the issues of mapping between two image domains with non-similar spatial arrangements as a mapping from each speckled pixel to each original image pixel is possible. Despite this, the use of a large number of convolutional layers removes the scalability benefit of the approach. There is therefore no evidence that the convolutional based models truly improve the scalability issue as these models are very large considering the resolution of images.

Fan et al. (2019) use a convolutional neural network to invert transmission effects. Here, the convolutions and pooling reduce the speckled resolution before a dense linear layer predicts the original image. As a result the model will be sensitive to the learned downsampling of the convolutions and suffer from similar scalability issues to Moran et al. (2018) due to the inclusion of a dense linear layer.

All of these approaches are mostly expected to work for classes of objects that belong to the class that was used for training (Borhani et al., 2018). Rahmani et al. (2018) make the first attempt to demonstrate generalisation outside of the training domain, although the results demonstrate limited evidence of true generalisation due to the choice of images featuring limited high frequency features and low quality reconstructions. Caramazza et al. (2019) demonstrate stronger generalisation due to the testing data choice, but there is still scope to improve the reconstruction quality outside of the training domain.

4.4.1.4 Equivariance

Cylindrical Harmonics

While the circular harmonics have seen some attention in the deep learning community due to the use in constructing rotational equivariant CNNs, the cylindrical harmonics have seen less attention (Klicpera et al., 2020). The cylindrical harmonics appear as a solution to Bessel functions for integer α and are therefore of interest for problems where information transformation is characterised by such functions. For example Bessel functions are used when solving wave or heat propagation. Bessel functions are solutions for different complex numbers α of Bessel's differential equation:

$$x^2\frac{d^2y}{dx^2} + x\frac{dy}{dx} + (x^2 - \alpha^2)y = 0. \tag{4.7}$$

For integer values of α the Bessel function solutions are a linearly independent set of functions expressed in cylindrical coordinates. Each function consists of the product of three functions. The radially dependent term is typically called the cylindrical harmonics. Further details on the connection between group theory and the Bessel basis functions is provided in Section 4.4.2.1.

4.4.1.5 Generation of Theoretical Transmission Matrices

Light propagation through a MMF is characterised by the transmission modes of the fibre. The fibres considered in this work have a step index profile with the refractive index within the core being constant and at a higher value than the cladding. The analytical solutions for the fibre modes can be determined by solving the Helmholtz equation in cylindrical coordinates, details for which are given in Appendix 4.4.2.1. The problem can therefore be expressed as an eigen-value eigen-function problem, where the eigen-functions are the fibre modes and the eigen-values are the propagation constants, β, of the modes. The solutions for the electric field within the fibre are comprised of Bessel functions of the first kind inside the core and modified Bessel functions of the second kind in the cladding as follows:

$$f_l^{core}(r,\theta) = \frac{J_l(u(\beta)\cdot\frac{r}{R})}{J_l(u(\beta))}e^{\pm il\theta}, \tag{4.8}$$

$$f_l^{clad}(r,\theta) = \frac{K_l(w(\beta)\cdot\frac{r}{R})}{K_l(w(\beta))}e^{\pm il\theta}, \tag{4.9}$$

where r and θ are the radial and azimuthal coordinates, respectively, l is the azimuthal index of the mode, R is the fibre core radius, and u and w are normalised frequencies defined as follows:

$$u(\beta) = R\sqrt{k_0^2 n_{core}^2 - \beta_{lm}^2}, \tag{4.10}$$

and

$$w(\beta) = R\sqrt{\beta_{lm}^2 - k_0^2 n_{clad}^2}, \tag{4.11}$$

where n_{core} and n_{clad} are the refractive indices of the core and cladding, respectively, k_0 is the vacuum wave number, and m is the radial mode index. Further details on the connection between propagation of light through MMFs and group theoretic equivariant neural networks is provided in Appendix 4.4.2.1.

Taking the derivative of Equations 4.8 and 4.9, and equating the resulting functions asserts a smoothness condition on the electric field across the core-cladding boundary. Solving this for all values of β over all possible integer values of l gives the propagation constants. These constants correspond to the fibre modes given by the, now appropriately, parameterised Equations 4.8 and 4.9. Examples of the mode fields are given in Figure 4.4.

A diagonal fibre propagation matrix, F, can then be made from the eigen-values, β, of the Helmholtz problem and the corresponding eigen-functions can be recorded in a matrix, M, which maps the image space to the fibre mode, or group space. We can find the inverse mapping bases from the fibre or group space back to the image space, $M^\dagger$, by taking the conjugate transpose. These three components allow us to construct the TM as:

$$\text{TM} = MFM^\dagger. \tag{4.12}$$

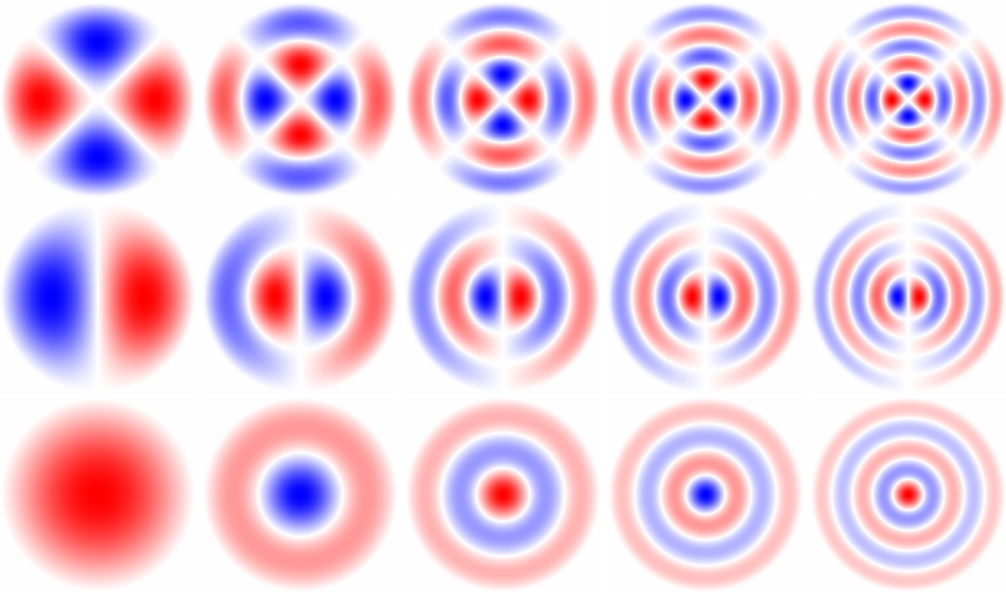


Figure 4.4: Electric field amplitude profiles of the Bessel bases used within our model. Here red is positive, blue is negative and the colour saturation represents the electric field strength. These bases are often known as LP modes.

4.4.2 Model Specification

Our method comprises two models: a Bessel equivariant model and a convolutional post-processing model. The Bessel equivariant model uses a basis set of the form of the cylindrical harmonics to ensure that the model is equivariant to known symmetries in optical fibres. The model learns a diagonal complex mapping function between bases, which from a physics perspective can be interpreted as learning the propagation constants of each mode in the fibre or from an equivariance perspective a weight matrix learning how each filter contributes to the reconstruction task. The Bessel equivariant model is constrained by the basis choice to produce circular images, which can have circular artifacts also seen in a true inversion, see Figure 4.3. The post-processing model is used to fill gaps in the corners, remove circular artifacts, and sharpen the images. The two stage approach is useful in safety-critical applications as users can inspect the output of both models, where the Bessel equivariant model output is less likely to depend on the training data due to the choice of inductive bias.

4.4.2.1 Bessel Equivariant Model

The core of our Bessel equivariant model has an inductive bias that better replicates the physics of the task of inverting the transmission effects of MMF imaging. To achieve this we utilise prior knowledge of MMFs and the modes with which an image propagates through the fibre. Similar to the successes of rotation equivariant neural networks, which make use of circular harmonics to achieve rotation equivariance, we utilise cylindrical harmonics as an inductive bias of the model. This is motivated by the propagation modes of the fibre being characterised by Bessel functions, where the radially dependent solution is given by the cylindrical harmonics. We make the connection to the group theoretic development of equivariant neural networks in Section 4.4.2.1, demonstrating the connection between the group $\mathrm{SO}^+(2,1)$, the light cone, and Bessel function solution to the wave equation.

The first stage of building the Bessel equivariant model is the computation of two sets of basis functions, which is done prior to training and only has to be completed once when creating the model. The first set transforms an input image into a function that lives on the group, while the second transforms from a function that lives on the group back into the image domain. These basis functions are sampled on a grid given by the size of the speckled images for the first basis set and the original images for the second basis set. These bases are not trainable parameters and are not updated during training of the model. These basis functions are an alternative to those used in general rotation equivariant neural networks, which typically offer equivariance to the group $\mathrm{SO}(2)$, except they correctly model the symmetry group, $\mathrm{SO}^+(2,1)$ of the task of inverting transmission effects of a MMF due to the added time dimension. An example of the bases used is given in Figure 4.4. The basis functions used within the model are the same as the fibre modes that are created when constructing a theoretical TM, which is detailed in Section 4.4.1.5.

The second component of the Bessel equivariant model is a diagonal complex-valued weight matrix. This comprises the trainable weights of the Bessel equivariant model. This complex-valued weight matrix is diagonal, as, in theory there is no mode mixing, and therefore the model only needs to learn the complex propagation constant associating each input mode to its corresponding output mode. In practise manufacturing defects, sharp bends, dopant diffusion, elliptical cores, amongst other reasons can cause mode mixing (Carpenter et al., 2014), which can be compensated for within the model by relaxing the diagonal constraint placed upon the weight matrix. The model's weight matrix linearly transforms the function mapped by the first basis set. The model trainable parameters, under the diagonal constraint, are equal to the number of bases within the model.

The overall architecture of the Bessel equivariant model is therefore a mapping from the image space to the fibre mode space provided by the first basis set, then a learned update function provided by the complex weight matrix, and finally a mapping from the fibre model space to the output image space provided by the second basis set. The overall model architecture is given in Figure 4.5.

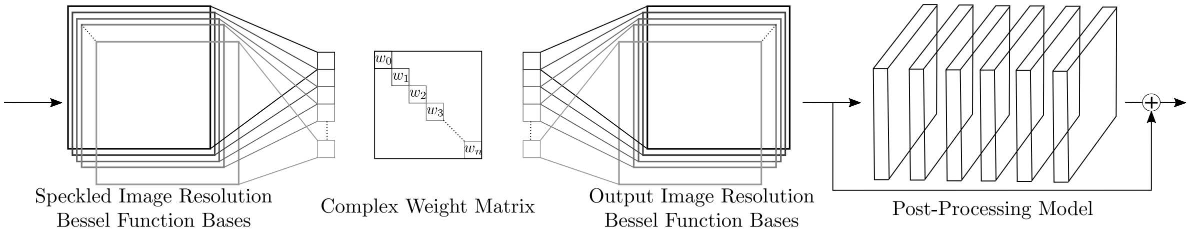

w_0
w_1
w_2
w_3
w_n
Speckled Image Resolution Bessel Function Bases
Complex Weight Matrix
Output Image Resolution Bessel Function Bases
Post-Processing Model

Figure 4.5: Our two stage model architecture comprising of a Bessel equivariant model and post-processing model. Input (left) is speckled images. Output (right) is predicted image. The speckled image resolution Bessel function bases transform the speckled image into a mode space of the group $\mathrm{SO}^+(2,1)$. The complex weight matrix is diagonal. The original image resolution Bessel function bases provides a mapping from the mode space to the output image space. The trainable parameters of the Bessel equivariant model are the diagonal complex weight matrix only. The post-processing model is a convolutional model.

Group Theoretic Understanding of Optical Fibre Transmission Modes

When a light beam propagates in free space or in a transparent homogeneous medium, its transverse intensity profile generally changes. Despite this, there exists certain distributions that do not change intensity profile as they traverse. These non-changing profiles are the transmission modes of the space.

The development of group equivariant networks exploits a similar principle, where these networks are constructed under the more general principle of finding basis functions called irreducible representations of some group. Some examples of this principle include: for the circle S^1 or line $\mathbb{R}$ the irreducible representations are given by complex exponentials $\exp(in\theta)$, for the group $\mathrm{SO}(2)$ the irreducible representations are given by the circular harmonics, for the group $\mathrm{SO}(3)$ the irreducible representations are given by the Wigner D-functions, and for S^2 the irreducible representations are given by the spherical harmonics. Thus a function on the group can be composed as a linear combination of the corresponding irreducible representations. On the other hand, when learning a function it is possible to learn the weightings of each of the irreducible representations of the group to learn that function. Building a model such that its feature space comprises the irreducible representations of the group provides a method of constructing a model that is equivariant to the underlying group of the representations.

Here we seek to show a connection between the known properties of optical fibres and group equivariant networks. We start by providing some details on the propagation of light through a fibre. The propagation of light is governed by the time-independent (Helmholtz) equation, which in cylindrical coordinates is given by Equation 4.13.

$$\frac{\partial^2 E}{\partial r^2} + \frac{1}{r}\frac{\partial E}{\partial r} + \frac{1}{r^2}\frac{\partial^2 E}{\partial \phi^2} + q^2 E = 0. \tag{4.13}$$

The standard approach to solving the above equation is to use the separation-of-variables procedure, which assumes a solution of the form given in Equation 4.14.

$$E_z = AF_1(r)F_2(\phi). \tag{4.14}$$

Due to the circular symmetry of the fibre, each component must not change when the coordinate ϕ is increased by a multiple of 2π. Therefore, we make the following assumption:

$$F_2(\phi) = e^{i\nu\phi}, \tag{4.15}$$

where $\nu \in \mathbb{Z}$.

Substituting Equation 4.14 into Equation 4.13 yields a wave equation of the following form:

$$\frac{\partial^2 F_1}{\partial r^2} + \frac{1}{r}\frac{\partial F_1}{\partial r} + \left(q^2 - \frac{\nu^2}{r^2}\right) F_1 = 0. \tag{4.16}$$

Which is a differential equation for Bessel functions. Solving this both inside the core of the fibre and in the cladding of the fibre provides two solutions. In the core of the fibre, as $r \to \infty$ the guided modes must remain finite. Thus for $r \leq a$ for core radius a the solution is a Bessel function of the first kind:

$$E_z(r \leq a) = AJ_\nu(ur)e^{i\nu\phi}e^{i(\omega t - \beta z)} \tag{4.17}$$

and outside of the core the solution is a modified Bessel function of the second kind:

$$E_z(r \geq a) = CK_\nu(wr)e^{i\nu\phi}e^{i(\omega t - \beta z)}. \tag{4.18}$$

Solving these equations provides the transmission modes of the fibre, i.e. the non-changing distributions.

Next we consider the propagation of light through a fibre from a group theoretic perspective. The indefinite special orthogonal group $\mathrm{SO}(2,1)$ is the group considering two spatial dimensions and one time dimension, and can be realised as:

$$SO(2,1) = \{X \in \mathrm{Mat}_3(\mathbb{R}) | X^t \nu X = \nu, \det(X) = 1\} \tag{4.19}$$

where,

$$\nu = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}. \tag{4.20}$$

To identify the Lie algebra $\mathfrak{so}(2,1)$ we use the tangent space $T_1SO(2,1)$ to $\mathrm{SO}(2,1)$ at the identity 1. We then choose a curve $a : L \to SO(21)$ such that $a'(0) = A$ Then, the characterising equation of A gives:

$$a(t)^T \nu a(t) = \nu. \tag{4.21}$$

Taking the derivative with respect to t gives:

$$a'(t)^T \nu a(t) + a(t)^T \nu a'(t) = 0. \tag{4.22}$$

Then, evaluating the expression at $t = 0$ gives:

$$A^T \nu + \nu A = 0. \tag{4.23}$$

Now we check the linear conditions determined by the above characterisation to then write out the Lie algebra, with ν having a natural block decomposition. Therefore, for a general element $A \in \mathfrak{so}(2,1)$, given as:

$$A = \begin{pmatrix} W & x \\ y^T & z \end{pmatrix}, \tag{4.24}$$

where $W \in M(2,\mathbb{R})$, $x, y \in \mathbb{R}^2$, and $z \in \mathbb{R}$. In this block decomposition $\nu = \begin{pmatrix} \mathbb{I}_2 & 0 \\ 0 & -1 \end{pmatrix}$. Then, Equation 4.23 becomes:

$$\begin{pmatrix} W^T & y \\ x^T & z \end{pmatrix} \begin{pmatrix} \mathbb{I}_2 & 0 \\ 0 & -1 \end{pmatrix} + \begin{pmatrix} \mathbb{I}_2 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} W & x \\ y^T & z \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}. \tag{4.25}$$

Therefore, the following conditions are imposed:

$$W^T = -W \tag{4.26}$$

$$y = x \tag{4.27}$$

$$z = 0. \tag{4.28}$$

Hence, the Lie algebra $\mathfrak{so}(2,1)$ is given by:

$$\mathfrak{so}(2,1) = \left\{ \begin{pmatrix} W & x \\ x^T & 0 \end{pmatrix} : W^T = -W \right\} = \left\{ \begin{pmatrix} 0 & -w & x_1 \\ w & 0 & x_2 \\ x_1 & x_2 & 0 \end{pmatrix} \right\}. \tag{4.29}$$

As a result, we can see that the condition on W is such that $W \in \mathfrak{so}(2, \mathbb{R})$. Therefore, the condition on W characterises the rotational symmetry of the group. Given the goal is make the connection between the group theoretic understanding of group equivariant neural networks and solution to the wave equation in cylindrical coordinates, this is promising as the solutions to Bessel functions have rotational symmetries.

Further, the group $\mathrm{SO}(2, 1)$ is that of two spatial dimensions and one time dimension. The connection between the three spatial dimensional case and the view as a light cone was made in the development of Minkowski spacetime. Here you imagine a cone where the time axis runs from the point of a cone through the centre of the plane drawn on the open end and the two spatial axes form a plane which intersects the cone and is parallel to the plane drawn on the open end. This is known as the future light cone and is an interpretation of how light spreads out after an event occurs. The group actions of the group $\mathrm{SO}^+(2, 1)$, which is the group $\mathrm{SO}(2, 1)$ with the requirements that $t > 0$, act on this space and can be viewed as rotations of the three dimensional Euclidean sphere. A connection can be drawn between this view and the fact that non compact generators of $\mathrm{SO}(n, 1)$ differ from corresponding matrix elements of the same generators of $\mathrm{SO}(n + 1)$, the group of rotations in $n + 1$-dimensional space, by a factor of $\sqrt{(}-1)$ (Wong, 1974). Therefore, the connection can be made between the group actions of $\mathrm{SO}^+(2, 1)$ and the light cone view of light propagation.

A final connection can be made, now that there is a connection between the group action and the light cone, between the light cone and the wave equation. Returning to the wave equation we note that it describes waves travelling with frequency independent speed. The character of the solution is different in odd and even dimensional spaces. In an odd dimensional space a disturbance propagates on the light cone and vanishes elsewhere. On the other hand, in an even dimensional space a disturbance propagates inside the entire light cone. Therefore, in an even dimensional space a disturbed medium never returns to rest. This phenomena if known as geometric dispersion. Here we are interested in even dimensional space as we have two spatial dimensions governing transmission through the fibre along with a third time dimension. We therefore expect the propagation of light through the fibre to be understood by propagation inside the entire light cone. Given the wave equation:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \tag{4.30}$$

a solution is possible through considering the three-dimensional theory if we regard u as a function in three-dimensions and that the third dimension is independent. Therefore, we make the following requirements:

$$u_0(0,x,y) = 0 \tag{4.31}$$

$$u_t(0,x,y) = \phi(x,y). \tag{4.32}$$

Then the three-dimensional solution equation becomes:

$$u(t,x,y) = tM_{ct}[\phi] = \frac{t}{4\pi}\iint_S \phi(x+ct\alpha, y+ct\beta)d\omega, \tag{4.33}$$

where α and β are the first two coordinates on the unit sphere and $d\omega$ is the area element on the sphere. This integral can be written as a double integral over the disc D with centre x, y and radius ct:

$$u(t,x,y) = \frac{1}{2\pi ct}\iint_D \frac{\phi(x+\xi, y+\nu)}{\sqrt{(ct)^2 - \xi^2 - \nu^1}}d\xi d\nu. \tag{4.34}$$

Therefore, it becomes clear that the solution does not only depend on the data on the light cone where

$$(x-\xi)^2 + (y-\nu)^2 = c^2t^2, \tag{4.35}$$

but on the entire data inside the light cone. It can therefore be seen how the solution to the wave equation yields the interpretation of the light cone in the precise way that was yielded by the analysis of the group $\mathrm{SO}^+(2,1)$. This completes the connections between the group action of $\mathrm{SO}^+(2,1)$ and the solution to the Bessel functions that govern the transmission modes within optical fibres. Now we can compose a model in a similar way to how circular harmonics are used to construct rotation equivariant neural networks under the group $\mathrm{SO}(2)$, but by utilising the basis functions found by solving the Bessel function in Equation 4.16 to compose a network which is equivariant to the group $\mathrm{SO}^+(2,1)$. This is useful as it positions the model with respect to other group equivariant neural networks, and provides a model with suitable inductive bias for the task of learning a function approximation to transmission through optical fibres.

4.4.2.2 Post-Processing Model

As demonstrated in Section 4.4.1.2 the limited number of modes means that the transmission effects of a MMF are typically not fully invertible leading to information loss. We would therefore not expect an equivariant model informed by the physics of the task to be able to fully invert the transmission effects. To overcome this we add a second model which is similar to a super-resolution model, except that it does not change the resolution and instead learns to remove circular artifacts, fill in gaps, and sharpen the images. It is a fully convolutional model comprising of convolutional layers and non-linearities. This model takes as input the output of our Bessel equivariant model, which we suspect will be similar to what is achievable by passing the speckled images through an inverted TM, and predicts the original images. This model is composed of eight convolutional layers with ReLU non-linearities. The motivation behind the explicit separation of the inversion and enhancement models is that in some applications, especially safety-critical applications, it is important to be able to see the results of an inversion conditioned on the observations alone, regardless of prior expectations.

4.4.2.3 Training Details

Throughout the paper we make use of three different datasets that are commonly used in machine learning, namely MNIST (LeCun et al., 1998), fMNIST (Xiao et al., 2017), and ImageNet (Deng et al., 2009). MNIST and fMNIST are also commonly used in the study of inverting transmission effects of MMFs due to their low resolutions. MNIST images are 28×28 images of handwritten digits between 0 and 9. fMNIST images are 28×28 images of items of clothing, such as trousers, jumpers, or shoes. ImageNet is a large scale dataset of higher resolution images which we size to be 256×256. We only use a subset of the imageNet dataset to demonstrate the ability to invert such high resolution images. For each of these datasets we split into a training and testing dataset. The training dataset is used for training, while the testing dataset is not used during training and we use this to test the model after training. We use the testing dataset to generate all the predicted images throughout this chapter.

We train each model for 200 epochs with the stochastic gradient descent algorithm. We use the mean squared error as a loss function between the original images and the predicted images. We do not use any regularisation on the model weights. For the Real and Complex linear models and the Bessel equivariant model we train each model for 200 epochs. For the post-processing model, as it is an addition to the Bessel equivariant model, we train this model for 200 epochs after the Bessel equivariant model has already been trained for 200 epochs and the Bessel equivariant model's weights have been frozen. For the training, all models are trained on 1 Titan RTX GPU with 24GiB of memory. We attempted training the complex linear model on a A6000 GPU with 48GB of memory on ImageNet to check if this was possible, but the model required more memory.

We utilise two different sources of data throughout the experimental section. The first of these is collected using a theoretical TM, which was developed specifically for this chapter and makes it easier and faster to create a new dataset through removing the need for experimentation in the lab. We provide details of the theoretical TM generation in Section 4.4.1.5. For this theoretical dataset we have both the amplitude and phase information for the speckled patterns. In addition, we also have access to the original and inverted images, where the inverted images are created by first passing an image through the TM and then back through its inverse. The second source of data was collected by Moran et al. (2018), which is data collected in the lab using a real fibre. For this data we only have the amplitude information for the speckled patterns. In addition, we also have the original images which have constant phase. This data has no licence and it accessible on the projects GitHub page.

4.4.3 Experiments

The experiments section is broadly split into experiments on the data of Moran et al. (2018) collected in in the lab and data generated using a theoretical TM. The data from Moran et al. (2018) was created using a physical optical fibre and we refer to this data source to as lab based data. The code for creating theoretical TMs was developed as part of work for a paper, of which this section is based on. This allows the developed model to be tested on new datsets with much greater ease than would be possible if all datsets had to be collected in the lab. In addition, it makes it possible to test on higher-resolution images, which due to the scaling issues of prior works, had not been attempted in the lab previously and therefore no such dataset exists. We refer to data generated utilising a theoretical TM created with this code as theoretical TM data. Within the lab based data the experiments are further broken down into:

- Experiments which validate that our new approach can invert the transmission effects of lab based data, despite using significantly fewer trainable model parameters
- Experiments on the relaxation of the diagonal constraint of the complex weight matrix within the model to account for losses in lab based systems

Further, the experiments on data from a theoretical TM can be broken down as follows:

- Experiments which validate that our new approach can invert the transmission effects of theoretical data, despite using significantly fewer trainable model parameters
- Experiments which demonstrate the ability of our model to generalise to out-of-distribution data
- Experiments considering the effects of noise in speckled images.

- Experiments which demonstrate the predictive ability of our model on reduced size datasets
- Experiments which demonstrate the predictive ability of our model when the Bessel basis set is under parameterised
- Experiments which validate that our new approach can invert the transmission effects of higher resolution images than has been previously been considered

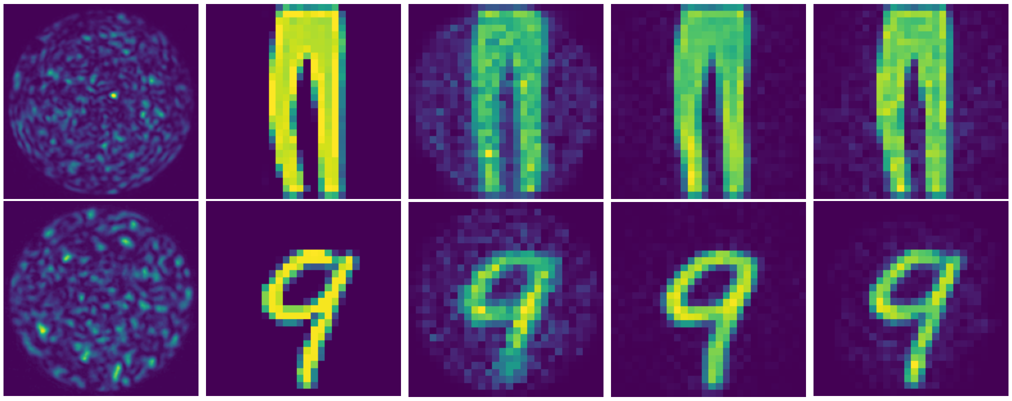
4.4.3.1 Lab Based MMF – Inversions

In the first experiment we compare our model to the complex-valued linear model developed by Moran et al. (2018) by considering both MNIST and fMNIST data. This aims to demonstrate that our model can invert the transmission effects of a fibre in a lab based context. For this we train and test separate models for the MNIST and fMNIST datasets.

Figure 4.6 (d) demonstrates that our model provides clear and accurate reconstructions of the target images whilst using orders of magnitude fewer trainable parameters. In addition, visually, our model (d) produces less noise that the complex linear model (e). Furthermore in (c), despite having some noise in the predictions, the object is clearly visible in the output of the Bessel equivariant model only. Therefore it is demonstrated that the Bessel equivariant model only is able to reconstruct the digit or item such that it is visible, and this is correctly sharpened to a realistic prediction of the original image by our post-processing model. We provide further visualisations in Figures B.1 and B.2 in the appendix.

Table 4.4 shows that the complex linear model outperforms our model in terms of achieving a lower loss function on both MNIST and fMNIST data. Despite this our model is still solving the task of inverting lab based data successfully. Furthermore, Table 4.4 shows that the complex linear model is over fitting to the training as it achieves a lower loss value on the training dataset than the testing dataset, while out model achieves comparable losses on both datasets.

We also compare the number of trainable parameters within the model in Table 4.5. This highlights that our model requires two orders of magnitude fewer parameters that the complex linear model. In addition, it can be seen that without the post-processing model our Bessel equivariant model only requires two orders of magnitude fewer parameters that the complex linear model. Therefore, our model has the potential to scale to higher resolution images where the complex linear model will run out of GPU memory.



(a) Input (b) Target (c) BEM (d) BEM (PP) (e) Complex

Figure 4.6: Comparison of predicted images from inverting transmission effects of a MMF. The upper row is fMNIST data, lower row MNIST data. (a) Input speckled image, (b) the target original image to reconstruct, (c) Output of the Bessel equivariant model, (d) Output of the combination of Bessel equivariant and post-processing model, and (e) the output of the complex-valued linear model.

Table 4.4: Comparison of the loss values of each model trained with MNIST or fMNIST data.

	MNIST		fMNIST	
Model	Train Loss	Test Loss	Train Loss	Test Loss
Complex Linear	0.00396	0.00684	0.00509	0.01061
Bessel Equivariant Diag	0.03004	0.03125	0.02903	0.03140
Bessel Equivariant Diag + PP	0.01317	0.01453	0.01609	0.01749

Table 4.5: Comparison of # trainable parameters in each model trained with MNIST or fMNIST data.

Model	Number of Trainable Parameters (Millions)
Complex Linear	78.826
Bessel Equivariant Diag	0.057
Bessel Equivariant Diag + Post Proc	0.500

4.4.3.2 Lab Based MMF – Accounting for Losses

In the second experiment we explore our model's ability to invert the transmission effects of a real world fibre by relaxing the diagonal constraint of our model. This aims to demonstrate that our model can overcome issues such as manufacturing defects, sharp bends, dopant diffusion, elliptical cores, amongst other reasons which cause reality to deviate from theory. For this we train and test separate models for the MNIST and fMNIST datasets.

As we noted in Section 4.4.2.1, the assumption that the mapping between Bessel bases is diagonal holds in theory, although in practice an imperfect system could lead to a necessity to relax this assumption. Here we explore this for data collected using a real fibre by building four different versions of our model:

1. A model with the assumption of a diagonal weight matrix
2. A model that has the diagonal assumption relaxed to allow five elements either side of the diagonal to be populated
3. A model that has the diagonal assumption relaxed to allow ten elements either side of the diagonal to be populated
4. A model full a fully populated weight matrix

If the fibre and experimental set-up were in an ideal setting we would expect the model with a diagonal weight matrix to be sufficient to produce near perfect reconstructions, up to the optimisation process finding the weights such a model. The second model allowing five elements to be populated either side of the diagonal would allow the model to capture some of the block diagonal structure seen in Carpenter et al. (2014), although this would still occlude the mapping between modes further apart in our mapping space. Similarly, the third model allows for capturing more of the block diagonal structure by allowing ten elements to be populated either side of the diagonal. Finally, the model with a fully populated weight matrix allows the greatest flexibility and has the greatest potential to capture deviations from theory as seen in practice, although, this model does not take advantage of the known diagonal structure that this mapping function has and is therefore over-parameterised.

Figure 4.7 (a, c, e, g) demonstrates that as the diagonal assumption of the complex weight matrix is relaxed, our Bessel equivariant model only produced images closer to that original image through capturing more high frequency detail and predicting less noise. Our models with five and ten elements either side of the diagonal populated with trainable parameters (c, e) improves upon the result achieved by the model with

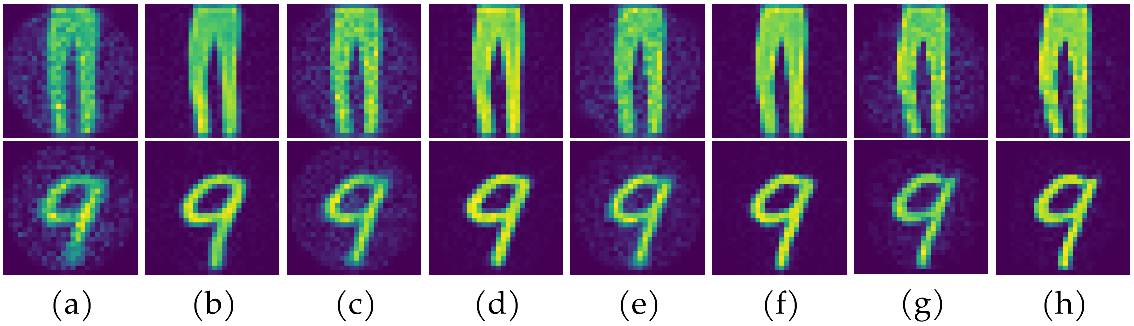


Figure 4.7: Comparison of predicted images from inverting transmission effects of a MMF with varying relaxations on the assumption the fibre has a diagonal fibre propagation matrix. The upper row is fMNIST data, lower row MNIST data. (a, c, e, g) Reconstructions using the Bessel equivariant model only. (b, d, f, h) Reconstructions using the Bessel equivariant model and post-processing model. (a, b) Reconstructions when the complex weight matrix is diagonal. (c, d) Reconstructions when the complex weight matrix has five block diagonal structure. (e, f) Reconstructions when the complex weight matrix has ten block diagonal structure. (g, h) Reconstructions when the complex weight matrix is full.

diagonal weight matrix (a) by both removing noise, better capturing the bend in the trouser leg, and producing a sharper version of the number nine. Further, the model with full populated weight matrix (g) is able to generate digits that like the correct digit. When combined with the post-processing model (b, d, f, h) all variants of the Bessel equivariant model produce clear high-quality reconstructions. We provide further visualisations of the results of models with different diagonal relaxation in Figures B.3 and B.4 in the appendix. This is a promising result as it demonstrates that our model allows for a design choice of complexity, with the option to trade off performance for a reduced memory requirement whilst maintaining accurate results even in the lowest memory configuration; this is something that is not possible with the Real or Complex linear models. Choosing a model with a low number of off-diagonal trainable elements, we believe, is the best trade off between considering losses and imperfections of the real-world while benefiting from the known sparsity of the mapping function between Bessel bases.

Table 4.6 shows the training and testing loss values for each variant of the model, split to show the output of the Bessel equivariant model only and when combined with the post-processing model. This shows that our model with a full mapping matrix between bases, i.e. full relaxation of the diagonal constraint, outperforms all other models. In addition, with a relaxation to allow for a 10 block diagonal structure our model performs comparably with the complex linear model despite requiring two orders of magnitude fewer parameters. The test loss continues to increase as the diagonal constraint is enforced in the Bessel equivariant weight matrix indicating a worsening of the reconstructed image.

Despite this, our model still provides clear reconstructions of the target images as is shown in Figure 4.7.

Table 4.6: Comparison of the loss values of each model trained with MNIST or fMNIST data.

	MNIST		fMNIST	
Model	Train Loss	Test Loss	Train Loss	Test Loss
Complex Linear	0.00396	0.00684	0.00509	0.01061
Bessel Equivariant Diag	0.03004	0.03125	0.02903	0.03140
Bessel Equivariant Diag + PP	0.01317	0.01453	0.01609	0.01749
Bessel Equivariant 5 Off Diag	0.01782	0.01931	0.02057	0.02354
Bessel Equivariant 5 Off Diag + PP	0.00658	0.00740	0.01138	0.01315
Bessel Equivariant 10 Off Diag	0.01362	0.01521	0.01788	0.02140
Bessel Equivariant 10 Off Diag + PP	0.00488	0.00576	0.00943	0.01162
Bessel Equivariant Full	0.00300	0.00684	0.00548	0.01380
Bessel Equivariant Full + PP	**0.00145**	**0.00378**	**0.00306**	**0.00964**

We provide the number of trainable parameters within each model in Table 4.7. This highlights that all of our models except the one with full weight matrix requires two orders of magnitude fewer parameters that the complex linear model. As the diagonal mapping between Bessel bases is relaxed to include 5 or 10-block diagonal structure the number of trainable parameters increases, but is still very low in comparison to the complex linear model. In the most flexible version of our model the number of trainable parameters is comparable with the complex Linear model, although we have demonstrated that this level of flexibility is not required to achieve good image reconstruction.

Table 4.7: Comparison of number of trainable parameters in each model trained with MNIST or fMNIST data when accounting for losses in a physical fibre.

Model	Number of Trainable Parameters (Millions)
Complex Linear	78.826
Bessel Equivariant Diag	0.007
Bessel Equivariant Diag + PP	0.450
Bessel Equivariant 5 Off Diag	0.059
Bessel Equivariant 5 Off Diag + PP	0.502
Bessel Equivariant 10 Off Diag	0.124
Bessel Equivariant 10 Off Diag + PP	0.567
Bessel Equivariant Full	42.665
Bessel Equivariant Full + PP	43.108

4.4.3.3 Theoretical TM - Inversions

This experiment compares each model when trained on a training subset of fMNIST images and their speckled equivalent and tested on a testing subset of fMNIST images and there speckled equivalent. This aims to demonstrate that our model can invert the transmission effects of a theoretical fibre. We provide a comparison of the predictions of a complex valued linear model (Moran et al., 2018), our equivariant model only, and our equivariant model coupled with a post-processing model on fMNIST images. We create the dataset using 28×28 fMNIST images and generate 180×180 speckled images using the theoretical TM.

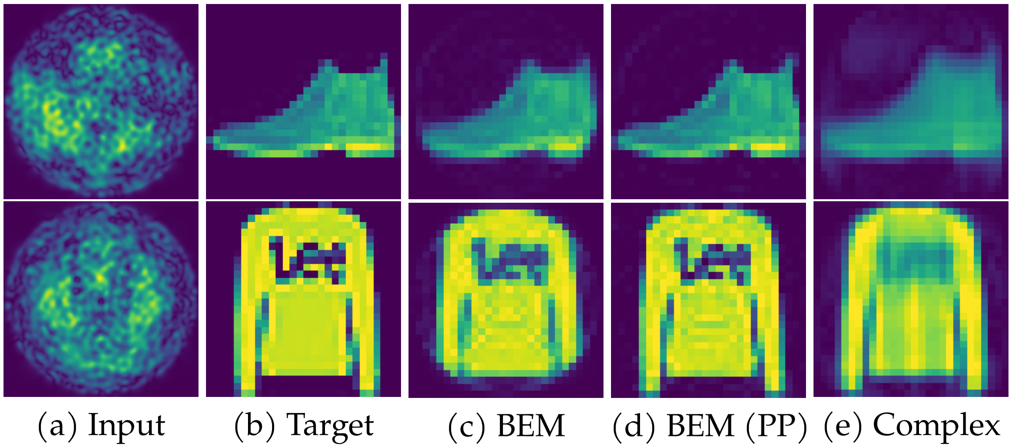


Figure 4.8: Comparison of predicted images from inverting transmission effects of a theoretical MMF with fMNIST data. (a) Input speckled image, (b) Target original image to reconstruct, (c) Output of the Bessel equivariant model, (d) Output of the Bessel equivariant and post-processing model, and (e) Output of the complex valued linear model.

Figure 4.8 shows reconstructions produced by the three models. This demonstrates visually that our model, even without the post-processing part, is able to reconstruct the original images. The post-processing model refines the output of the equivariant model and fills in gaps outside the fibre. Despite it being possible to estimate the general category of clothing from the outputs of the complex linear model, the higher frequency information has been lost. For example, the pattern on the jumper is not as clear as in the original image. On the other hand, our model is able to reconstruct this higher frequency information. We provide further reconstructions in Figure B.5.

Further to the reconstructions, loss values for each model are provided in Table 4.8. This demonstrates that out Bessel equivariant model only achieves a comparable loss value to the complex linear model. Although, this result is a consequence of the circular nature of the Bessel basis functions, and as such the Bessel equivariant model under-performs due to not being able to reconstruct pixel values close to the edge of the image. On the

the hand, the complex linear model under-performs due to failing to reconstruct higher frequency information and over-fitting to the general clothing categories. Finally, when our Bessel equivariant model is combined with the post-processing model it outperforms all other models.

Table 4.8: Comparison of the loss values of each model trained with fMNIST data and the corresponding speckled patterns.

Model	Train Loss	Test Loss
Complex Linear	0.0149	0.0146
Bessel Equivariant	0.0141	0.0139
Bessel Equivariant + Post Proc	**0.0032**	**0.0032**

We provide the number of trainable parameters in Table 4.9. Similarly to learning to reconstruct lab based data, our model requires two orders of magnitude fewer parameters than the complex linear model. Further, when considering only the Bessel equivariant part of the model it requires four orders of magnitude fewer parameters than the complex linear model.

Table 4.9: Comparison of number of trainable parameters in each model trained with fMNIST data and the corresponding 180×180 speckled patterns, created by a theoretical fibre with ~ 1000 modes.

Model	Number of Trainable Parameters (Millions)
Complex Linear	25.402
Bessel Equivariant Diag	0.001
Bessel Equivariant Diag + PP	0.222

4.4.3.4 Theoretical TM – Generalising To Out-Of-Distribution Data

This experiment tests the ability of each model trained in Section 4.4.3.3 to generalise to a new dataset. For this, we take each model trained on reconstructing fMNIST images from their corresponding speckled patterns and test there ability to reconstructing MNIST images from there corresponding speckled patterns. Previous works have demonstrated some ability to generalise to new data classes (Rahmani et al., 2018; Caramazza et al., 2019), although in each case the results are not perfect.

Figure 4.9 shows reconstructions produced by the three models. This demonstrates visually that our Bessel equivariant model without the post-processing model generalises well to an out-of-distribution dataset. On the other hand, the complex linear model somewhat predicts the correct digits. In this out of training domain our post-processing model does not add any value to the original Bessel equivariant model, which has already predicted a good reconstruction. This highlights the benefit of our two-stage modelling approach as we have one robust model and a second model to sharpen the images, as a result if a user believes the situation could be unusual they could reliably use the output of our Bessel equivariant model and not the post-processing part. We provide further reconstructions in Figure B.6.

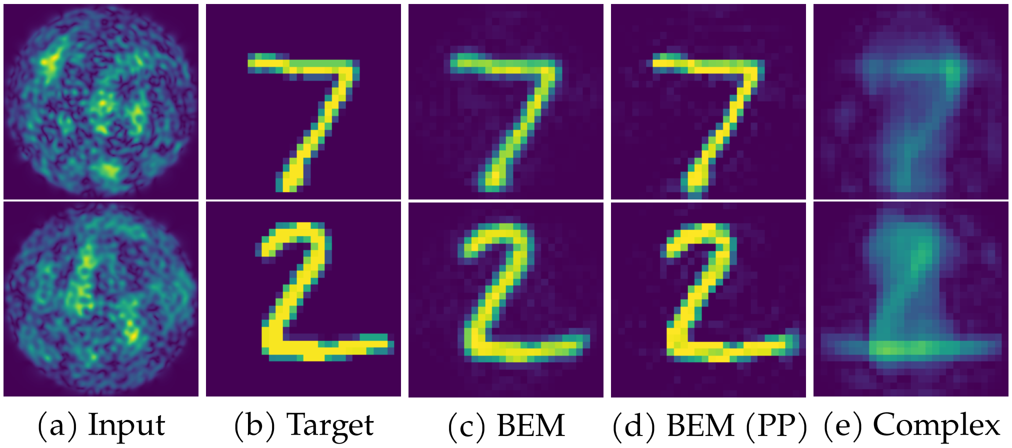


Figure 4.9: Comparison of predicted images from inverting transmission effects of a MMF for models trained with fMNIST data and tested on MNIST data. (a) Input speckled images, (b) Target original images, (c) Output of the Bessel equivariant model, (d) Output of Bessel equivariant and post-processing model, and (e) Output of the complex valued linear model.

Table 4.10 presents the loss values for testing of each models on the MNIST dataset. This shows that our Bessel equivariant model significantly out-performs the complex linear model in generalising to a new dataset. Furthermore, when comparing to the fMNIST testing loss provided in Table 4.8, it can be seen that the loss value for our model is very similar between both datasets. This further demonstrates the ability of our model to better generalise to new datasets.

Table 4.10: Comparison of the loss values of each model trained with fMNIST data.

Model	MNIST Test Loss
Complex Linear	0.0363
Bessel Equivariant	**0.0026**
Bessel Equivariant + Post Proc	0.0028

4.4.3.5 Theoretical TM – Effects of Noise on Speckled Images

Here we compare the effect of noise on the speckled images when using a theoretical TM. In each case the speckled images are saturated at 90% of there maximum value and Gaussian noise is added with different standard deviation values. We demonstrate the reconstruction ability of our model in Figure 4.10 when Gaussian noise with 0.01 standard deviation is used, in Figure 4.11 when Gaussian noise with 0.05 standard deviation is used, in Figure 4.12 when Gaussian noise with 0.1 standard deviation is used, and in Figure 4.13 when Gaussian noise with 0.5 standard deviation is used. Figure 4.10 demonstrates that with a low level of noise added to the speckled patterns our model is largely unaffected. Figures 4.11 and 4.12 show that as more noise is added to the speckled patterns noise begins to appear in the reconstructions of our model, but that image is still easily identifiable. Finally, Figure 4.13 demonstrates that when significant noise is added to the speckled patterns the reconstruction quality begins to drop, although in this instance the speckled pattern is visually indistinguishable from the noise. We also provide the loss values in Table 4.11. These results demonstrate that our approach is robust to noise.

Table 4.11: Comparison of the loss values of each model trained with fMNIST data when the speckled images are saturated at 90% and Gaussian noise is added with standard deviation given in the column Noise Level.

Noise Level	Model	Train Loss	Test Loss
0.01	Bessel Equivariant	0.0140	0.0142
	Bessel Equivariant + Post Proc	0.0033	0.0033
0.05	Bessel Equivariant	0.0147	0.0149
	Bessel Equivariant + Post Proc	0.0043	0.0043
0.1	Bessel Equivariant	0.0166	0.0168
	Bessel Equivariant + Post Proc	0.0049	0.0049
0.5	Bessel Equivariant	0.0373	0.0377
	Bessel Equivariant + Post Proc	0.0164	0.0166

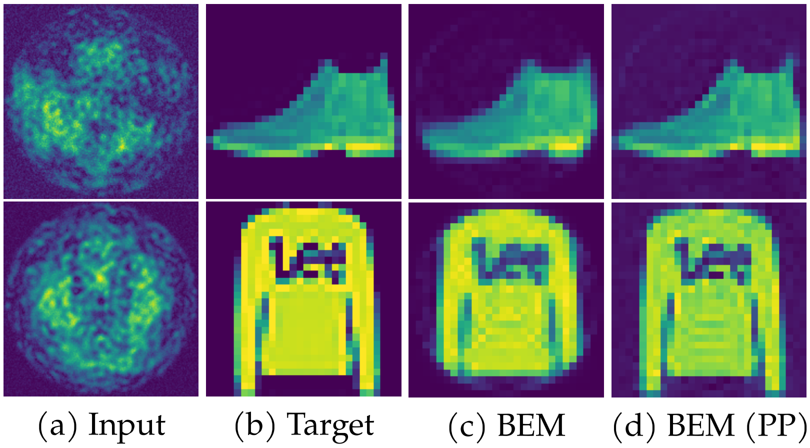


Figure 4.10: Comparison of predicted images from inverting transmission effects of a MMF using fMNIST data created with a theoretical TM when the speckled images are saturated at 90% of their maximum value and Gaussian noise with 0.01 standard deviation added. (a) The input noisy speckled image, (b) the target original image to reconstruct, (c) the output of the Bessel equivariant model, and (d) the output of the combination of Bessel equivariant and post-processing model.

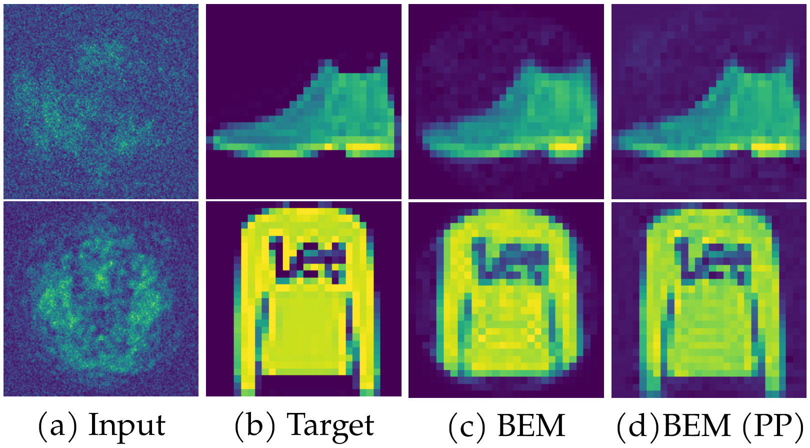


Figure 4.11: Comparison of predicted images from inverting transmission effects of a MMF using fMNIST data created with a theoretical TM when the speckled images are saturated at 90% of their maximum value and Gaussian noise with 0.05 standard deviation added. (a) The input noisy speckled image, (b) the target original image to reconstruct, (c) the output of the Bessel equivariant model, and (d) the output of the combination of Bessel equivariant and post-processing model.

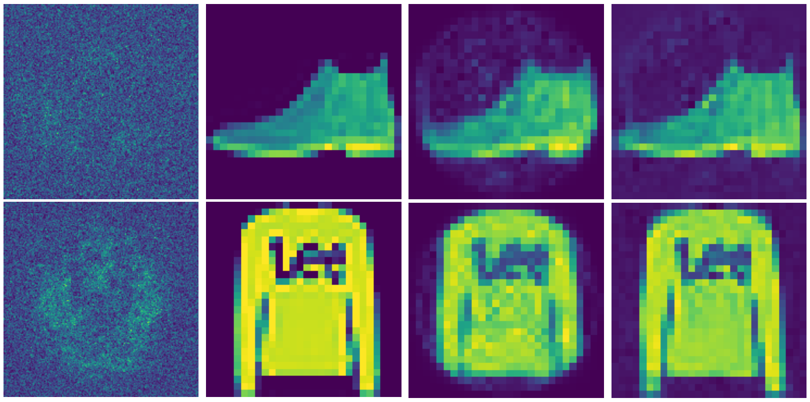

(a) Input (b) Target (c) BEM (d) BEM (PP)

Figure 4.12: Comparison of predicted images from inverting transmission effects of a MMF using fMNIST data created with a theoretical TM when the speckled images are saturated at 90% of their maximum value and Gaussian noise with 0.1 standard deviation added. (a) The input noisy speckled image, (b) the target original image to reconstruct, (c) the output of the Bessel equivariant model, and (d) the output of the combination of Bessel equivariant and post-processing model.

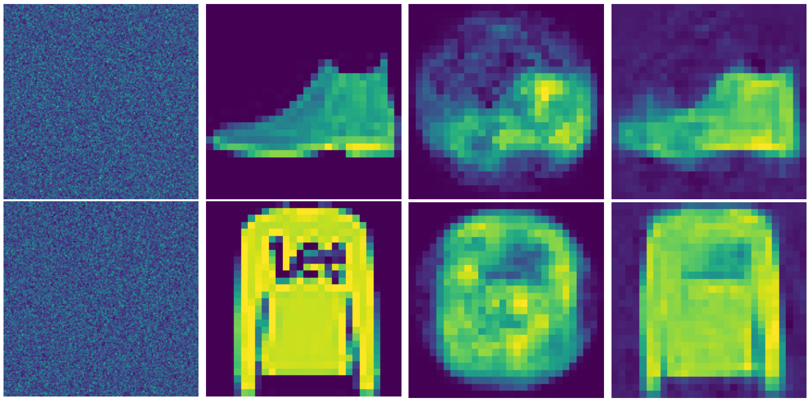
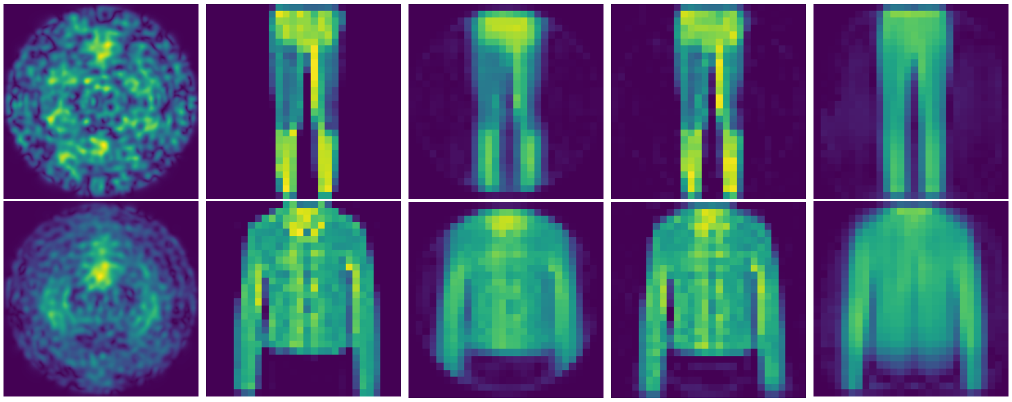

(a) Input (b) Target (c) BEM (d) BEM (PP)

Figure 4.13: Comparison of predicted images from inverting transmission effects of a MMF using fMNIST data created with a theoretical TM when the speckled images are saturated at 90% of their maximum value and Gaussian noise with 0.5 standard deviation added. (a) The input noisy speckled image, (b) the target original image to reconstruct, (c) the output of the Bessel equivariant model, and (d) the output of the combination of Bessel equivariant and post-processing model.

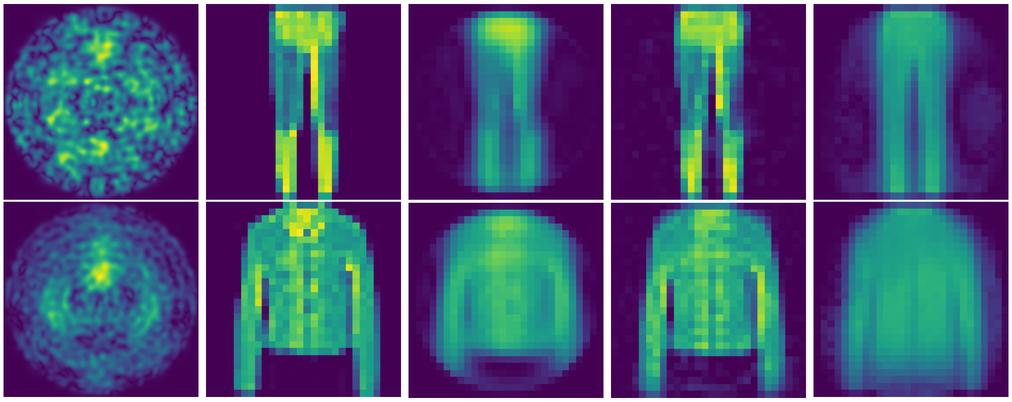
4.4.3.6 Theoretical TM – Reduced Training Dataset Size

In this section we compare the effect of reducing the training dataset size on the reconstruction ability of the models. For this, we make use of the fMNIST dataset created using a theoretical TM and select different size subsets to train on. Figure 4.14 shows the reconstructions when the dataset is reduced in size by a half and includes 6000 training examples. This highlights that the output of our Bessel equivariant model on its own has become more blurred than when the entire dataset is used. Similarly, the complex linear model also now predicts more blurred images. Despite this when using our full model the post-processing model sharpen and correct the prediction to match the target. The result of increased blurring in the predicted images by our Bessel equivariant model only and the complex linear model is increased in Figures 4.15 and 4.16 as the training dataset is reduced in size, although in each the object is still somewhat detectable. Despite this increased blurring of the predicted images, the post-processing model is still able to correct the images to closely resemble the targets. Finally in Figure 4.17, with only 120 training examples, the complex linear model predicts and empty image. Furthermore, our Bessel equivariant model predicts a shape from which the object is not identifiable and the post-processing model goes some way to correcting to the target image, but falls short of producing a good reconstruction. Never-the-less these results demonstrate that our model can learn to reconstruct the target images with less training examples than the complex linear model. This is most likely due to the significantly reduced number of trainable parameters of our model allowing for faster generalisation.



(b) Input (b) Target (c) BEM (d) BEM (PP) (d) Complex

Figure 4.14: Comparison of predicted images from inverting transmission effects of a MMF. The training dataset was reduced from the original size of 12000 to 6000. (a) The input speckled image, (b) the target original image to reconstruct, (c) the output of the Bessel equivariant model, (d) the output of the combination of Bessel equivariant and post-processing model, and (e) the output of the Complex valued linear model. The data used was fMNIST with speckled patterns created with a theoretical TM.



(a) Input (b) Target (c) BEM (d) BEM (PP) (e) Complex

Figure 4.15: Comparison of predicted images from inverting transmission effects of a MMF. The training dataset was reduced from the original size of 12000 to 2400. (a) The input speckled image, (b) the target original image to reconstruct, (c) the output of the Bessel equivariant model, (d) the output of the combination of Bessel equivariant and post-processing model, and (e) the output of the Complex valued linear model. The data used was fMNIST with speckled patterns created with a theoretical TM.

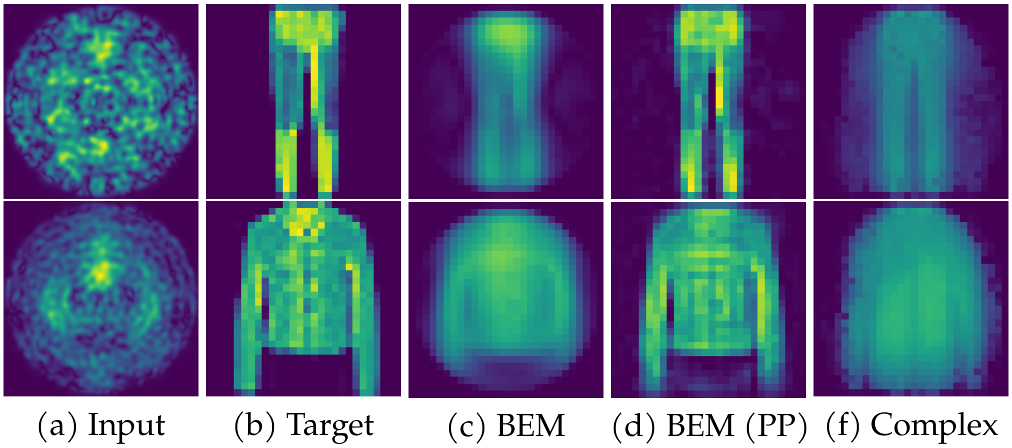


Figure 4.16: Comparison of predicted images from inverting transmission effects of a MMF. The training dataset was reduced from the original size of 12000 to 1200. (a) The input speckled image, (b) the target original image to reconstruct, (c) the output of the Bessel equivariant model, (d) the output of the combination of Bessel equivariant and post-processing model, and (e) the output of the Complex valued linear model. The data used was fMNIST with speckled patterns created with a theoretical TM.

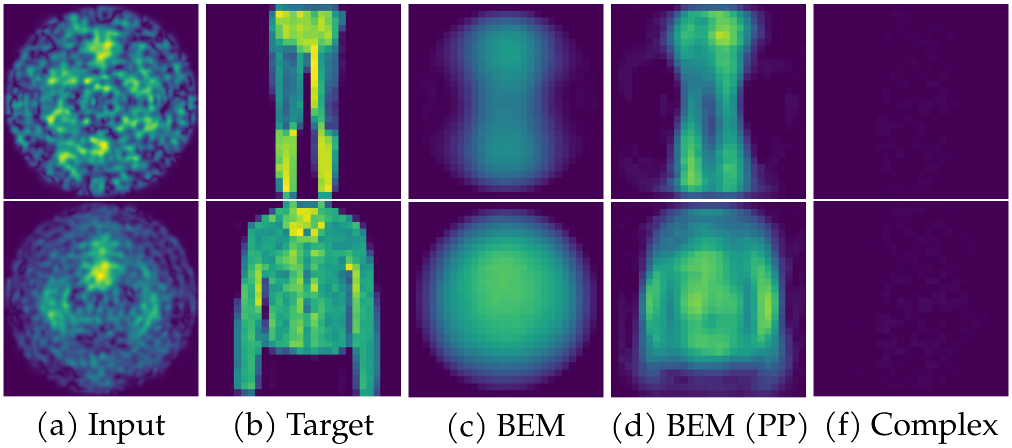


Figure 4.17: Comparison of predicted images from inverting transmission effects of a MMF. The training dataset was reduced from the original size of 12000 to 120. (a) The input speckled image, (b) the target original image to reconstruct, (c) the output of the Bessel equivariant model, (d) the output of the combination of Bessel equivariant and post-processing model, and (e) the output of the Complex valued linear model. The data used was fMNIST with speckled patterns created with a theoretical TM.

4.4.3.7 Theoretical TM – Under Parameterising the Basis Set

In this section we consider the impact of under parameterising the Bessel function bases. This demonstrates the capability of the model when the number of bases is not specified to match the given fibre being used and could represent a situation where details of the fibre being used are not known. For this we reduce the number of radial frequencies that feature in the set of basis functions. The original basis set we utilised for the data collected using a theoretical fibre for fMNIST comprises of 21 radial frequencies and 1061 bases. Here we show the original results using this full bases set in Figure 4.18 column (b). Next we show the results of removing the high frequency bases by only considering the first 14 radial frequencies, which amounts to having 932 bases in Figure 4.18 column (c). In addition, we show the results of removing further high frequency bases by only considering the first 7 radial frequencies, which amounts to having 567 bases in Figure 4.18 column (d). Finally, we show the results of removing further high frequency bases by only considering the first 4 radial frequencies, which amounts to having 322 bases in Figure 4.18 column (e). These results demonstrate that our model behaves in a predictable way when the basis set it under parameterised, namely that as the higher frequency bases are removed the resulting images predicted by the Bessel equivariant model only do not feature high frequency information. We provide further visualisations in Figures B.7, B.8, and B.9 in the appendix. These demonstrate that despite our Bessel equivariant model not being able to reconstruct high frequency information when the basis set is sufficiently under parameterised the post-processing model is able to sharpen the image and fill in some of the high frequency information. Therefore, when using our entire model issues of under parameterising the basis set are minimal and the model is robust to the number radial frequencies chosen.

The loss values for each model is provided in Table 4.12 and visualised in Figure 4.19, which backs up what is seen in Figure 4.18. This further demonstrates that as high frequency bases are removed from the model the loss values increase. Figure 4.18 shows that this loss increase is due to the models inability to reconstruct high frequency details.

Table 4.12: Comparison of the loss values of each model trained with fMNIST data created with a theoretical TM. This shows the impact of under parameterising the Bessel function basis set by removing higher frequency modes. 21 radial frequencies is the natural choice for the fibre considered, hence all values less than 21 give an under parameterised model.

Model	Radial Frequencies	# Modes	Train Loss	Test Loss
Bessel Equivariant	21	1061	0.0141	0.0139
Bessel Equivariant + Post Proc	21	1061	**0.0032**	**0.0032**
Bessel Equivariant	14	932	0.0158	0.0156
Bessel Equivariant + Post Proc	14	932	0.0036	0.0036
Bessel Equivariant	7	567	0.0203	0.0201
Bessel Equivariant + Post Proc	7	567	0.0053	0.0053
Bessel Equivariant	4	332	0.0299	0.0296
Bessel Equivariant + Post Proc	4	332	0.0090	0.0090

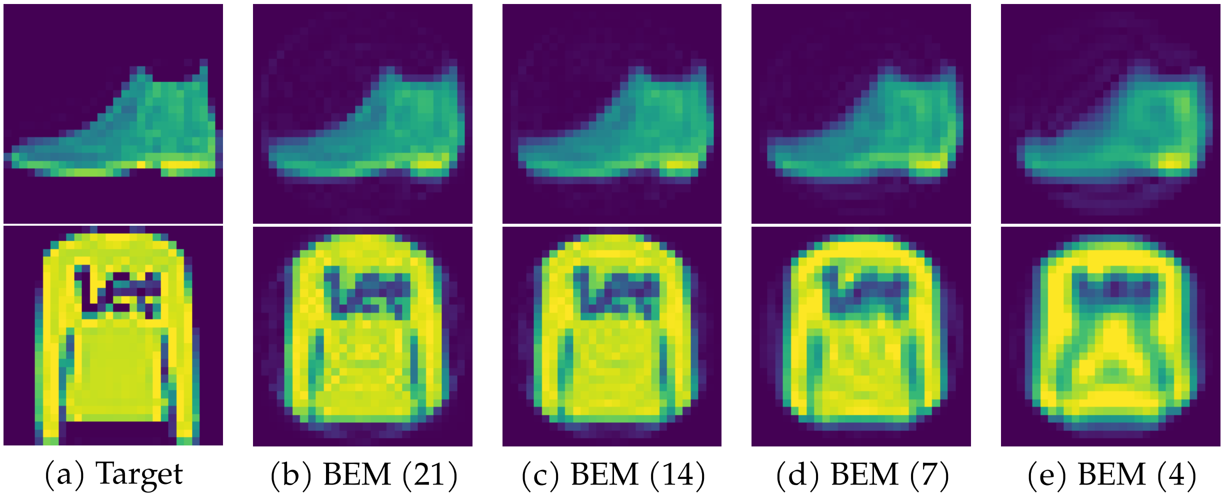


Figure 4.18: Comparison of predicted images from inverting transmission effects of a MMF, when reducing the number of radial frequencies present in the Bessel function bases. (a) The input speckled image. (b) The number of radial frequencies present in the Bessel function bases was left at the original value of 21, which represents 1061 bases. (c) The number of radial frequencies present in the Bessel function bases was reduced from the original value of 21 to 14, which represents a reduction in the number of bases from the original value of 1061 to 930. (d) The number of radial frequencies present in the Bessel function bases was reduced from the original value of 21 to 7, which represents a reduction in the number of bases from the original value of 1061 to 567. (e) The number of radial frequencies present in the Bessel function bases was reduced from the original value of 21 to 4, which represents a reduction in the number of bases from the original value of 1061 to 322. The data used was fMNIST with speckled patterns created with a theoretical TM.

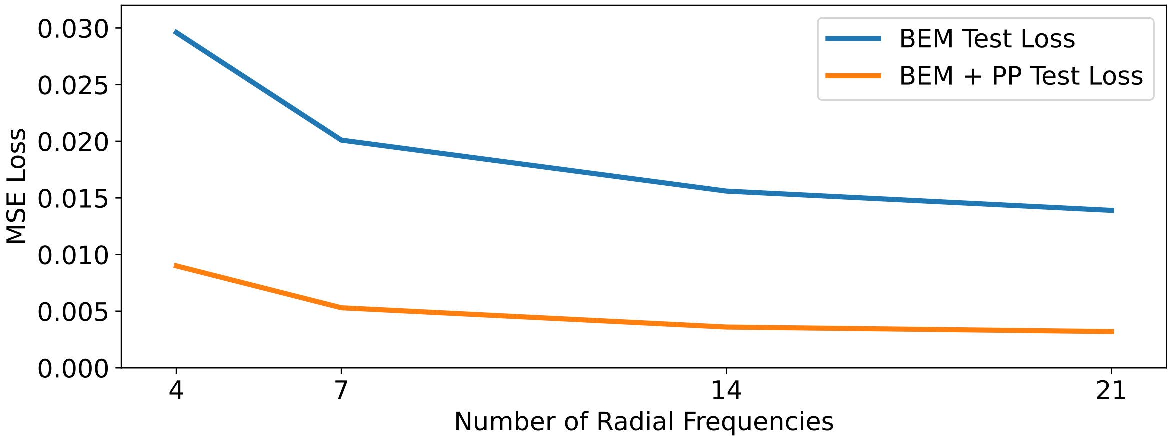


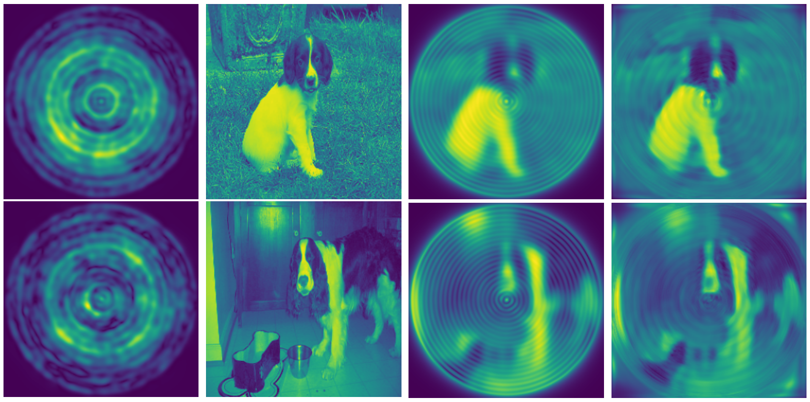
Figure 4.19: Comparison of the loss values of each model trained with fMNIST data created with a theoretical TM. This shows the impact of under parameterising the Bessel function basis set by removing higher frequency modes. 21 radial frequencies is the natural choice for the fibre considered, hence all values less than 21 give an under parameterised model.

4.4.3.8 Theoretical TM – Scaling to Larger Images – ImageNet

Scaling to higher resolution images has been a challenge in MMF imaging due to the non local relationship between pixels in speckled images and their corresponding original image, and as a result the dense mapping between these two domains is high-dimensional. This has prohibited prior methods from being able to reconstruct high resolution images from there corresponding speckled patterns. In this experiment we assess the ability of our model to scale to high resolution images and demonstrate the improved scaling of our model over prior works.

For this we experiment with a subset of images from the ImageNet dataset, where we fix the resolution to be 256×256. Inverting the transmission effects of such higher resolution images has not been previously achievable. We only compare our equivariant model with our equivariant model coupled with a fully convolutional post-processing model. The complex-valued linear model cannot fit in the GPU memory of an A6000 with 48.7GiB of memory with this resolution of image and therefore we cannot show reconstructions from this model.

Figure 4.20 shows that our Bessel equivariant model produces a reconstruction from which it is possible to determine the type of dog and its activity. The reconstruction does not capture all the high frequency information, but this fibre only has $\sim$1000 modes and as a result the true inversion does not correctly reconstruct the original image as was demonstrated in Section 4.4.1.2. Therefore this is a high quality reconstruction. When we combine the two models some of the artifacts of the Bessel equivariant model are removed as the post-processing model fills in information towards the corners and sharpens the image. Further reconstructions are provided in Figure B.10.



(a) Input (b) Target (c) BEM (d) BEM (PP)

Figure 4.20: Comparison of predicted images from inverting transmission effects of a MMF using high resolution ImageNet data. (a) Input speckled image, (b) Target original image to reconstruct, (c) Output of Bessel equivariant model (d) Output of Bessel equivariant and post-processing model.

Table 4.13: Comparison of the loss values of each model trained with ImageNet data.

Model	Train Loss	Test Loss
Bessel Equivariant	0.0541	0.0574
Bessel Equivariant + Post Proc	**0.0161**	**0.0159**

We present the loss values for both models in Table 4.13. This shows the reduction in loss possible by utilising the full model as the Bessel equivariant model on its own is heavily penalised due to not being able to reconstruct the image in the corners.

We compare the models' memory requirements in Figure 4.21. This shows how fewer trainable parameters within our model translates into significantly less memory use than the complex linear model. This effect is seen more drastically when scaling to high resolution images, where the complex linear model runs out of memory on a 24.2GiB Titan RTX for original and speckled images of resolution 180×180 pixels. Our model, in contrast, has been tested on 256×256 pixel images, requiring only 2.1GiB. In addition, despite the reconstruction not tested here, this shows that in theory our model can scale to megapixel images.

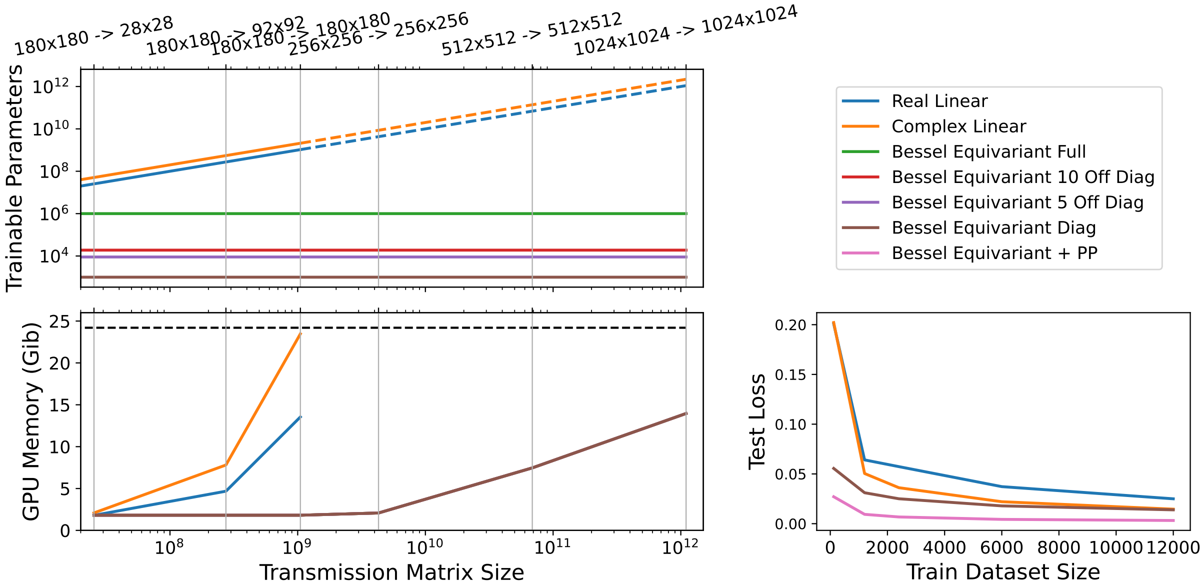


Figure 4.21: (Upper Left) Comparison of number of trainable parameters versus TM size (dashed line indicates model cannot in practice be built due to large memory requirements). (Lower Left) Comparison of required GPU memory against TM size. Vertical grey lines indicate a given image size – our model can process 1024×1024 images on a single consumer-level Titan RTX GPU. (Lower Right) Comparison of reducing the size of the training dataset. All plots are for a 1000 mode fibre.

4.4.4 Discussion

The results demonstrate that developing a neural network model that is equivariant to the symmetries of the task of inversion of MMF transmission significantly simplifies the learning task through reducing the number of parameters in the model. This improves the scalability of the model with respect to the image resolutions and makes the model more robust to out of distribution data. This opens up the possibility of using neural network based methods for the inversion of MMF transmission effects to applications that require the use of higher resolution images. In addition, the improved robustness to out of distribution data is particularly valuable in safety critical applications.

4.5 Conclusions

In Section 4.3 we developed a novel architecture for the task of segmentation of deforestation regions, which models the correct symmetries for the task, where few examples of rotation equivariant networks exist in comparison to non-rotation equivariant networks. Further, we compared this model to a more standardised model that has been widely used in deep learning for image segmentation. This demonstrates the practical benefits of incorporating the correct symmetries for a given task, namely increased accuracy of the predicted segmentation maps and the generation of stable segmentation maps under transformations that naturally occur. Finally, this provides a concrete example of building a rotation equivariant model for a task that has rotational symmetries. This will aid in making these techniques more readily adopted, by reducing the difficultly of implementing such a model for a given task when compared to the non-rotation equivariant counterparts.

In Section 4.4 we develop a novel architecture for the task of inverting the transmission effects of multi-mode optical fibres demonstrating a more interpretable model, better generalisation, improved robustness, and a model that can scale to higher resolution images. We overcome the challenge of bridging the gap between the understanding of multi-mode optical fibres in physics and equivariant neural networks in Section 4.4.2.1. We develop a method for generating theoretical transmission matrices, making conducting varied experiments more accessible in Section 4.4.1.5. We present a new type of model for solving the task of inverting the transmission effects of a MMF through developing a $\mathrm{SO}^+(2,1)$-equivariant neural network, which models the cylindrical symmetries known in light propagation through optical fibres, and combining this with a post-processing network in Sections 4.4.2.1 and 4.4.2.2. The post-processing network will learn features specific to the training data, and will generalise to new image classes as well as a general convolutional network. The use of a theoretical TM allows us to compare an 'ideal' inverse with the output of previous models based on learning a full transmission matrix. This suggests that previous learned transmission matrices were combining elements of the actual transmission matrix with elements of the post-processing network, which would affect their ability to generalise to new images. We also anticipate that in interactive safety-critical applications, users might want the ability to switch between modes, to be sure that the evidence was there, and not overly influenced by the priors in the training data.

We demonstrate that our model improves upon the SOTA complex linear models through incorporating a useful physically informed inductive bias into the equivariant network in Section 4.4.3. Firstly, in Section 4.4.3.1, we show that our model can reconstruct real world data through the use of images and speckled patterns generated by a fibre in the laboratory. In addition, we explore how the diagonal assumption within the complex weight matrix in our model can be relaxed to overcome losses in a real world system in Section 4.4.3.2. Secondly, we demonstrate the ability of our model the invert the transmission effects of the theoretical fibre in Section 4.4.3.3. Further, our equivariant network is shown to perform well on new image classes, providing a model that better generalises to out-of-distribution data in Section 4.4.3.4. We also demonstrate the impact on our model when there is noise in the speckled patterns, of having a reduced training dataset size; and when the Bessel function basis set is under parameterised in Sections 4.4.3.5, 4.4.3.6, and 4.4.3.7 respectively. These experiments demonstrate the robustness of our approach.

Finally, we demonstrate that our approach significantly improves the ability to scale to higher resolution images by improving the scaling law from $\mathcal{O}(N^4)$ to $\mathcal{O}(m)$, where N is pixel size and m is the number of fibre modes. We provide a comparison between our model and prior works on both data created using a real fibre and a theoretical TM, demonstrating in both cases that our model solves the task while using significantly less trainable parameters than the complex real model throughout Section 4.4.3. In addition, in Section 4.4.3.8, we demonstrate results on high resolution 256×256 images, which has previously been unachievable due to the growth of parameters with previous models. The dramatic reduction in the number of parameters for each fibre configuration opens the way for future models which can learn mappings for high-resolution images, from a wider set of perturbed fibre poses and combine these using architectures such as VAEs.

5

Rotational Symmetries in 3D Hand Reconstruction

In this chapter we develop a novel rotation equivariant model for the task of generating 3D hand meshes from 2D RGB images. The model comprises four main components: a rotation equivariant encoder, a mapping function from the base space of 2D grids to 2D vector spaces, a projection function from a 2D vector space to a 3D vector space, and a 3D rotation equivariant decoder. Each component is constructed to guarantee rotation equivariance around the 2D axes of the input images. We demonstrate that the proposed architecture generalises to new orientations better than prior works. We also show the mesh reconstruction accuracy on three benchmark datasets comparing to prior works.

5.1 Introduction

Vision based models for 3D hand mesh generation are currently receiving a lot of interest as they facilitate a wide range of applications. Some of these include virtual reality (VR), augmented reality (AR), and sign language recognition. Currently, two main approaches exist to tackle the difficulties of 3D mesh generation from 2D RGB images. One approach uses deformable models which exploit a prior distribution in 3D shape shape. These models can be trained on relatively small datasets, although typically they struggle to generalise to a diverse range of hand meshes. As such, they often do not accurately generate hand meshes of correct shape and pose. The second approach is to use neural network models, which are more general and do not exploit a 3D shape space prior. The neural network model is generally in the form of a convolutional encoder and graph or mesh neural network decoder. These models are able to more accurately predict the correct shape and pose of the hands, although these models currently require large amounts of data to train. One solution to this is to make use of synthetic data as a pre-training step. Although, the use of synthetic data often results in a model that does not generalise well to real-world examples. On the other hand, collecting large datasets of real-world data is expensive and time consuming, prohibiting this as a viable option in many cases. The lack of robustness presented from using synthetic data or the expense of gathering real-world data may limit the number of applications of the approach.

In this work we overcome the issue of requiring large amounts of data by building a neural network model with a suitable inductive bias. We quantitatively demonstrate the effectiveness of this approach on real-world data, outperforming previous state-of-the-art methods. Further, we qualitatively demonstrate the benefit of our choice of inductive bias and compare to alternative models. We follow the approach of using a neural network based decoder, rather than a deformable model with 3D shape prior, to overcome the issue with generalisation. In addition to building a model than can generalise, we constrain the weight matrices throughout the model to enforce rotational equivariance. This improved weight sharing in the model reduces the need for pre-training on large synthetic datasets and generalises well when trained only on a small real-world dataset. The resulting rotational equivariance of the model, which is a suitable inductive bias for generating 3D meshes from RGB images due to the symmetries present in the data, ensures that a 2D rotation of the input image corresponds to an equivalently rotated 3D mesh. To the best of our knowledge this is the first work proposing rotation equivariance for the generation of 3D meshes from RGB images. Furthermore, we compare models with both an MLP and GNN decoder to explore the benefit of using a decoder with locality as an inductive bias. This highlights some disadvantages to using a GNN decoder, and motivates us to build a MLP based model with rotational equivariance.

5.2 Background

5.2.1 Rotational Equivariance

The background on rotation equivariance on 2D grids, namely rotation equivariance for images, has been considered in Chapter 4.2. Therefore, we will not repeat this here and refer the reader to Chapter 4.2 for the background in rotation equivariance on 2D grids. Instead, this section provides the background on rotation equivariant networks for 3D vector spaces and more general methods of constructing equivariant neural networks.

In general for data living on a 3D continuous real base space the group considered is that of 3D rotations, $\mathrm{SO}(3)$, 3D rotations and translations, $\mathrm{SE}(3)$, or 3D rotations, reflections and translations, $\mathrm{E}(3)$ (Thomas et al., 2018; Fuchs et al., 2020; Satorras et al., 2021; Köhler et al., 2019; Schütt et al., 2017). The group representations $g \in \mathrm{SO}(3)$ are orthogonal matrices that can be decomposed as:

$$\rho(g) = Q^T \left[\oplus_l D^l(g)\right] Q, \tag{5.1}$$

where Q is a change-of-basis matrix and D^l is a Wigner-D matrix (Thomas et al., 2018; Fuchs et al., 2020). The Wigner-D matrices D map elements of the group $\mathrm{SO}(3)$ to $(2l+1) \times (2l+1)$-dimensional matrices. Each D_l is an irrep of the group $\mathrm{SO}(3)$. Scalars map to the trivial irrep and vectors in 3D space map to non-trivial irreps, namely the real Wigner D-matrices:

$$D^0(g) = 1 \quad and \quad D^n(g) = \mathcal{R}(g), \tag{5.2}$$

where $\mathcal{R}$ is a rotation matrix given by the action of $g \in \mathrm{SO}(3)$ on a vector in $\mathbb{R}^3$. Equation 5.1 shows that the group representation is a direct sum of irreps, where the number of irreps used l can be chosen.

A weight matrix kernel in an $\mathrm{SO}(3)$ equivariant linear layer is therefore constructed from the choice of group representations ρ. The weight matrix kernel is written as:

$$W^{lk} : \mathbb{R}^3 \to \mathbb{R}^{(2l+1)\times(2k+1)}, \tag{5.3}$$

which is a mapping from the corresponding vector space of the k^{th} irrep to the corresponding vector space of the l^{th} irrep. Weiler et al. (2018), Thomas et al. (2018), and Kondor (2018) showed that the kernel lies in the span of an equivariant basis $\{W_J^{lk}\}_{J=|k-l|}^{k+l}$. Therefore, the weight matrix in an $\mathrm{SO}(3)$ equivariant network is a learnable linear combination of such basis kernels

$$W_{\mathrm{SO}(3)} = \sum_{J=|k-l|}^{k+l} \psi_J^{lk} W_J^{lk}, \tag{5.4}$$

where W_J^{lk} is the weight matrix kernel, commonly referred to as a basis, and ψ_J^{lk} is a learnable function for the J^{th} coefficient.

The basis kernels W_J^{lk} can be constructed from spherical harmonics. The spherical harmonics $Y_m^{(l)}$ are functions defined on the surface of a sphere and they form a complete set of orthogonal functions, and thus an orthonormal basis. Spherical harmonics are basis functions for irreducible representations of $\mathrm{SO}(3)$. For the spherical harmonics Y_m^l l is a non-negative number and m is an integer between $-l$ and l. The real spherical harmonics for $l = 0$, namely scalar functions, and $l = 1$, namely functions in 3D vector space are:

$$Y^{(0)}(\hat{r}) \propto 1 \quad Y^{(1)}(\hat{r}) \propto \hat{r}, \tag{5.5}$$

where $\hat{r}$ is a vector in 3D space normalised to unit length.

These functions are $\mathrm{SO}(3)$ equivariant; that is, for all $g \in \mathrm{SO}(3)$ and $\hat{r}$,

$$Y_m^{(l)}(\mathcal{R}(g)\hat{r}) = \sum_{m'} D^l_{mm'}(g) Y_{m'}^{(l)}(\hat{r}). \tag{5.6}$$

Therefore, the weight matrices of an $\mathrm{SO}(3)$ equivariant linear layer are restricted such that,

$$W_J^{lk} = \sum_{-J}^{J} Y_{Jm} Q_{Jm}^{lk}. \tag{5.7}$$

Thomas et al. (2018) and Fuchs et al. (2020) provide methods for constructing such networks. Here each basis kernel $W_J^{lk} : \mathbb{R}^3 \to \mathbb{R}^{(2l+1)\times(2k+1)}$ is formed by taking a linear combination of Clebsch-Gordan matrices Q_{Jm}^{lk} of shape $(2l+1)\times(2k+1)$, where the J, m^{th} linear combination coefficient is the m^{th} dimension of the J^{th} spherical harmonic. As a result an $\mathrm{SO}(3)$ equivariant network can be constructed from the spherical harmonics.

Following this, work has been conducted on a more general method of constructing equivariant models Lang & Weiler (2020), Ravanbakhsh et al. (2017). However these methods still require considerable mathematical work limiting there general application. On the other hand, a general framework for constructing rotation-reflection equivariant convolutional networks for the group $\mathrm{E}(2)$ including subgroups was presented in Weiler & Cesa (2019). Further, a general framework for solving equivariance constraints for a range of different groups was presented for a multilayer perceptron (MLP) based architecture by Finzi et al. (2021). This method therefore recovers the same bases as other equivariant networks.

5.2.2 Graph Neural Networks

GNNs were first introduced as a means to learn on graph structured data using neural networks where the data is irregularly structured Gori et al. (2005), Scarselli et al. (2008), Li et al. (2015), Duvenaud et al. (2015). GNNs become more widely used when they were developed to scale better with the size of input graph Welling & Kipf (2016), Veličković et al. (2018), Bronstein et al. (2017), Gilmer et al. (2017a), Defferrard et al. (2016). Since then GNNs have been used to learn about graphs Chen et al. (2020), Simonovsky & Komodakis (2017), Morris et al. (2019), Xu et al. (2019), Mitton et al. (2021), point clouds Wu et al. (2019), Hertz et al. (2020), Zhao et al. (2021), Fuchs et al. (2020), Li & Lee (2019), and meshes Feng et al. (2019), Hanocka et al. (2019), Hanocka et al. (2020).

Graph- mesh- and point-convolutions are relevant here as they can be used in the decoder as an inductive bias to build into the model. They provide the inductive bias of locality in the model, meaning that features are updated only based on neighbouring nodes. On the other hand, using an MLP does not provide any locality inductive bias in the model, as the features at every node are updated based on the features of every other node.

5.2.3 3D Hand Mesh Generation

Generating meshes from 2D images has seen increased attention in recent years Chatzis et al. (2020). Using a convolutional encoder is the approach taken in all prior works to embed the image into a latent space, with differing choices of convolutional architecture. The main difference between prior works comes in the form of the decoder. One approach is to use a deformable model which has been fit to landmarks, exploiting a prior distribution in 3D shape space Baek et al. (2019), Boukhayma et al. (2019), Hasson et al. (2019), Zhang et al. (2019), Kulon et al. (2019). This has the advantage that the model will generate realistic looking hands. On the other hand, this approach has a weakness that the model will tend to generate hands only from the prior distribution and therefore the generated meshes may not accurately reflect the input image.

Another approach is to use more general models as the decoder with no prior distribution guiding the model to generate hands. One such method uses a ResNet-50 encoder and spacial mesh convolutional decoder (Kulon et al., 2020). Another method uses a stacked hourglass convolutional encoder and graph convolutional decoder utilising Chebyshev polynomials (Ge et al., 2019). While these approaches more accurately generate hand meshes with the correct shape and pose of the input image, they generally require large amounts of synthetic data as a pre-training step. This is not advantageous as it requires significant effort in creating a synthetic dataset. In addition, these methods are not robust at generating real-world meshes due to the reliance on large datasets. Current approaches (Ge et al., 2019; Kulon et al., 2020) choose a small sized neighbourhood for the update functions in the decoder, and do not consider a larger neighbourhood. This locality is lost if the graph chosen is a fully connected graph. Current methods allow longer range dependency between nodes in the graph by stacking multiple GNN layers. No attention appears to be paid to the impact of this choice and whether there is a benefit to using larger neighbourhoods during update functions or even fully connected graphs. Further, it can be noted that using an MLP layer instead of fully connected GNN could be seen as an alternative approach to allowing long range dependency on node features.

5.3 Rotation Equivariant 3D Hand Mesh Generation From 2D Images

5.3.1 Model Specification

5.3.1.1 Overview

We follow the model structure of previous works on learning to generating 3D hand meshes from RGB images (Kulon et al., 2020; Ge et al., 2019) consisting of a convolutional encoder to embed the image into a latent space and a decoder to generate the 3D meshes. The input to the model is RGB images of hands and the model outputs predicted 3D meshes of the hand. The decoder predicts the point positions of a mesh, where we follow previous work (Ge et al., 2019; Kulon et al., 2020) and output the points of the mesh in a fixed order, thus making use of a predefined topology.

Unlike previous works we consider the symmetries in the problem and create a fully rotation equivariant model. This comprises of an encoder and decoder, similarly to prior works, except in this instance both are rotation equivariant. Further, our model has two new modules a vector mapping function and a 3D projection function. Therefore our model comprises of four main components:

1. a rotation equivariant encoder
2. a rotation equivariant mapping function from functions defined on a 2D grid to functions defined on 2D continuous vector space
3. a rotation equivariant projection function from 2D to 3D
4. a rotation equivariant decoder

The encoder maps from images to functions defined on the group $C_8 \rtimes \mathbb{R}^2$. The vector mapping function maps from the output latent space of the encoder, which is a function on the group $C_8 \rtimes \mathbb{R}^2$ to 2D vector spaces which are acted upon by representations of the group $\mathrm{SO}(2)$. The 3D projection function maps from a vector feature space acted upon by representations of the group $\mathrm{SO}(2)$ to both an $\mathrm{SO}(2)$ invariant scalar function and a vector space acted upon by representations of the group $\mathrm{SO}(2)$. This allows us to introduce the third inferred dimension of the 3D space in a controlled way, namely by using the $\mathrm{SO}(2)$ invariant scalar function as the depth prediction. Finally the decoder is an $\mathrm{SO}(3)$ equivariant MLP model. By ensuring rotation equivariance at each stage of the model a 2D rotation of the input image corresponds to a 2D rotation of the output mesh about the depth axis. An overview of the model is provided in Figures 5.1, 5.2, 5.3, and 5.4.

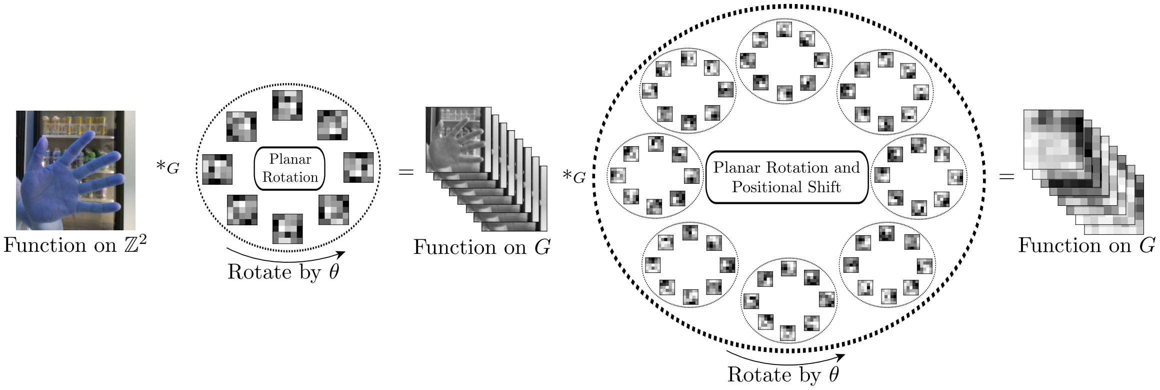


Figure 5.1: The first of four components is the 3D hand mesh generation model is the encoder. The encoder is a rotation equivariant convolutional neural network that takes as input 2D RGB images of hands and outputs a latent space that comprises of functions on the group $C_8 \rtimes \mathbb{R}^2$.

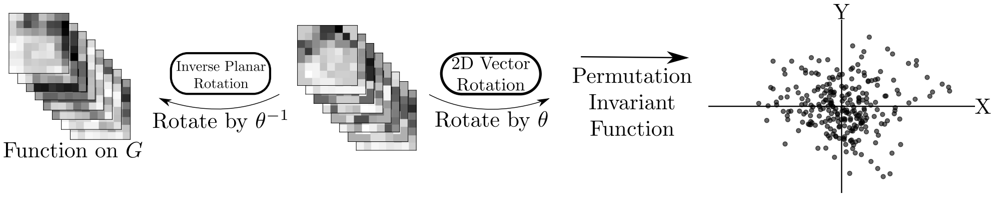


Figure 5.2: The second of four components is the 3D hand mesh generation model is the vector mapping function. The vector mapping function takes as input the latent functions and outputs 2D vector spaces of point vectors.

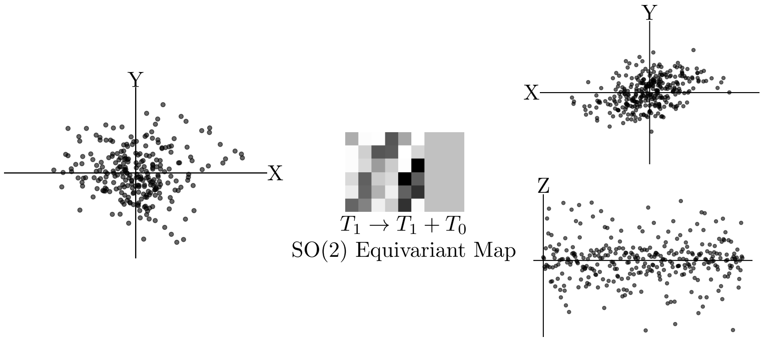


Figure 5.3: The third of four components is the 3D hand mesh generation model is the 3D projection function. The 3D projection function takes as input 2D vector spaces and outputs both a 2D vector space and a 1D vector, which are interpreted as the X-Y and Z dimensions in 3D space respectively. This function uses representations of the group $\mathrm{SO}(2)$ to map from a vector representation, T_1, to both a vector representation, T_1, and scalar representation, T_0.

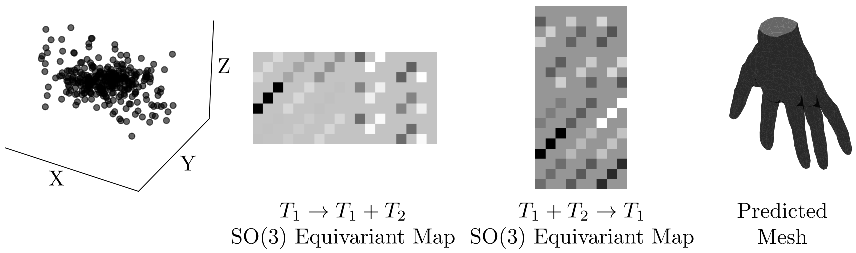
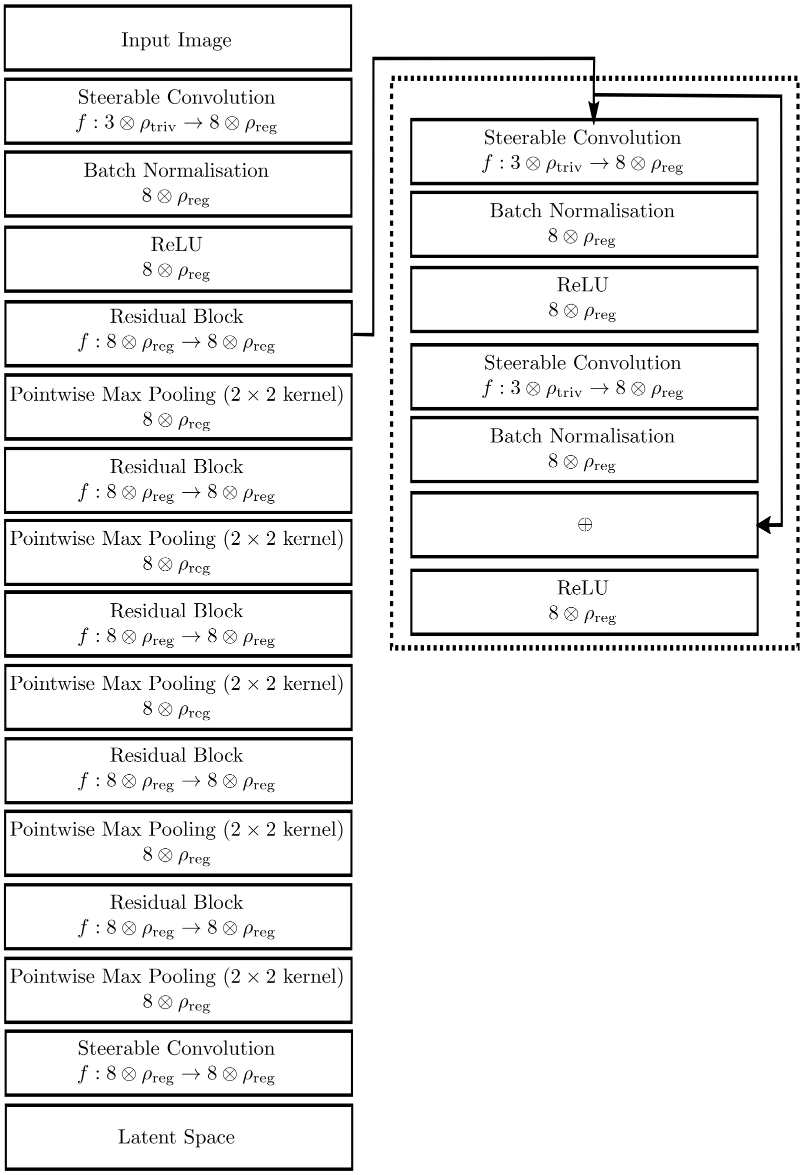


Figure 5.4: The fourth of four components is the 3D hand mesh generation model is the decoder. The decoder is a $\mathrm{SO}(3)$ equivariant function that maps from 3D points to 3D points, where the output is the hand mesh.

5.3.1.2 Encoder

For the encoder we use a steerable convolutional neural network (CNN) Cohen & Welling (2017), Weiler & Cesa (2019), which is translation-rotation equivariant. We chose to make the encoder equivariant to the group C_8, the group of rotations by $45°$. We enforce that the feature spaces transforms under the regular representation, ρ_{reg}, of the group C_8. This defines the transformation law of the feature fields in the network. Therefore, the steerable CNN has a feature space of steerable fields $f : \mathbb{R}^2 \to \mathbb{R}^8$, which associate an 8-dimensional vector $f(x) \in \mathbb{R}^8$ to each point x in the base space. The transformation law of the feature fields in the convolutional model is given by $f(x) := \rho(g) \cdot f(g^{-1}(x - t))$. This states that it transforms the feature fields by moving the feature vectors from x to their new position $g^{-1}(x-t)$, which both translates them by t and rotates the orientation by g^{-1}, and acts on them by $\rho(g)$, which permutes the order of the feature vector depending on the group element g.

For the encoder we define a residual block, which comprises of two steerable convolution layers, two batch normalisation layers Ioffe & Szegedy (2015), and two ReLU layers Nair & Hinton (2010) with skip connection He et al. (2016). The encoder comprises of a steerable convolution, batch normalisation, and ReLU layer followed by five residual blocks and final steerable convolution. The encoder is detailed in Figures 5.1 and 5.5.



Input Image
Steerable Convolution $f: 3\otimes\rho_{\text{triv}}\to 8\otimes\rho_{\text{reg}}$
Batch Normalisation $8\otimes\rho_{\text{reg}}$
ReLU $8\otimes\rho_{\text{reg}}$
Residual Block $f: 8\otimes\rho_{\text{reg}}\to 8\otimes\rho_{\text{reg}}$
Pointwise Max Pooling (2 × 2 kernel) $8\otimes\rho_{\text{reg}}$
Residual Block $f: 8\otimes\rho_{\text{reg}}\to 8\otimes\rho_{\text{reg}}$
Pointwise Max Pooling (2 × 2 kernel) $8\otimes\rho_{\text{reg}}$
Residual Block $f: 8\otimes\rho_{\text{reg}}\to 8\otimes\rho_{\text{reg}}$
Pointwise Max Pooling (2 × 2 kernel) $8\otimes\rho_{\text{reg}}$
Residual Block $f: 8\otimes\rho_{\text{reg}}\to 8\otimes\rho_{\text{reg}}$
Pointwise Max Pooling (2 × 2 kernel) $8\otimes\rho_{\text{reg}}$
Residual Block $f: 8\otimes\rho_{\text{reg}}\to 8\otimes\rho_{\text{reg}}$
Pointwise Max Pooling (2 × 2 kernel) $8\otimes\rho_{\text{reg}}$
Steerable Convolution $f: 8\otimes\rho_{\text{reg}}\to 8\otimes\rho_{\text{reg}}$
Latent Space
Steerable Convolution $f: 3\otimes\rho_{\text{triv}}\to 8\otimes\rho_{\text{reg}}$
Batch Normalisation $8\otimes\rho_{\text{reg}}$
ReLU $8\otimes\rho_{\text{reg}}$
Steerable Convolution $f: 3\otimes\rho_{\text{triv}}\to 8\otimes\rho_{\text{reg}}$
Batch Normalisation $8\otimes\rho_{\text{reg}}$
$\oplus$
ReLU $8\otimes\rho_{\text{reg}}$

Figure 5.5: Diagram of the encoder inside the model. This shows the representation space used at each layer in the model and the total number of layers. The right hand column details the components of the residual block, while the left hand column shows the entire structure of the encoder.

5.3.1.3 Vector Mapping

The vector mapping layer converts latent functions defined on the group $C_8 \rtimes \mathbb{R}^2$ to a 2D continuous latent vector space, which is acted upon by representations of the group $\mathrm{SO}(2)$. We achieve this by first inverting the action of the group on the base space, which, noting that the transformation law of the feature fields in the encoder is given by $f(x) := \rho(g) \cdot f(g^{-1}(x-t))$, involves applying the transformation $f(x) := f(g(x))$, where the group action is $g \in C_8$. We then transform the latent functions defined on the image domain, $\mathbb{Z}_2$, to latent functions defined on a 2D continuous vector space, $\mathbb{R}^2$. This is achieved by reshaping the latent functions from $(b \times m \times g \times k \times k)$ to $(b \times k \times k \times g \times n \times 2)$ by interpreting the learned latent function's feature dimension as n points in a 2D continuous vector space; where b is the batch size, m is the feature dimension, g is the group dimension, k is the dimension of the grid, and n is the halved feature dimension. Following this, we apply the corresponding group action $g^{-1} \in C_8 \leq \mathrm{SO}(2)$, where it is noteworthy that this does not require any prior knowledge of the orientation of the image. This corresponding group action is achieved by using representations of the group $\rho(g) \in \mathbb{R}^{2\times 2}$. Finally, we apply a permutation invariant function over the group axis as we specify that the output graph should be of fixed topology. This permutation invariant function ensures the topology of the output graph is not dependent on the permutations of the encoder's transformation law. Here we chose to use a summation as the permutation invariant function. The output of the vector mapping function is therefore a 2D continuous vector space, where the feature vectors correspondingly rotate when the input image is rotated. This mapping from 2D grids to 2D continuous vector space is implicitly done in prior works (Kulon et al., 2020; Ge et al., 2019), where this switch of input domain is given no consideration. Whereas, here we we develop a novel vector mapping function to ensure that rotation equivariance is maintained. The vector mapping function is detailed in Figure 5.2.

5.3.1.4 3D Projection

The 3D projection function maps from a function on a 2D continuous vector space acted upon by representations of the group $\mathrm{SO}(2)$ to both an $\mathrm{SO}(2)$ invariant scalar function and a function on a 2D continuous vector space acted upon by representations of the group $\mathrm{SO}(2)$. In tensor notation this is a mapping from $T_1^{\mathrm{SO}(2)}$ to $T_1^{\mathrm{SO}(2)} \oplus T_0^{\mathrm{SO}(2)}$. We treat the scalar representation as a learned depth estimate and therefore treat it as the third dimension of the 3D point cloud. We therefore reshape the $\mathrm{SO}(2)$ vector and scalar functions into a 3D vector space. The 3D projection function is detailed in Figure 5.3.

5.3.1.5 Decoder

The decoder maps the latent function, which is a function on a 3D continuous vector space acted upon by representations of the group $\mathrm{SO}(3)$, into the vertex points of the output mesh. The decoder is $\mathrm{SO}(3)$-equivariant and therefore guarantees that a rotation of the input image corresponds to a rotation of the predicted 3D mesh about depth axis introduced by the 3D projection function. The decoder comprises of three layers mapping between functions that live on a 3D continuous vector space acted upon by representations of the group $\mathrm{SO}(3)$. The resulting output of the model is a fixed topology mesh that comprises position vectors in 3D space. This decoder comprises of equivariant multi-layer perceptron layers Finzi et al. (2021). The architecture of the decoder is presented in Figures 5.4 and 5.6.

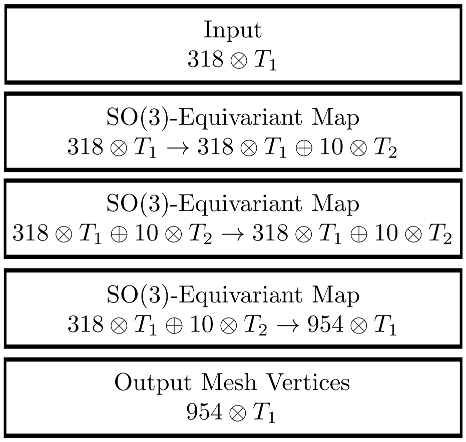

Input
$318 \otimes T_1$
SO(3)-Equivariant Map
$318 \otimes T_1 \to 318 \otimes T_1 \oplus 10 \otimes T_2$
SO(3)-Equivariant Map
$318 \otimes T_1 \oplus 10 \otimes T_2 \to 318 \otimes T_1 \oplus 10 \otimes T_2$
SO(3)-Equivariant Map
$318 \otimes T_1 \oplus 10 \otimes T_2 \to 954 \otimes T_1$
Output Mesh Vertices
$954 \otimes T_1$

Figure 5.6: Diagram of the decoder used inside the model. This shows the representation space used at each layer in the decoder.

5.3.1.6 Model Training

The loss function can consist of some combination of a vertex reconstruction loss, a Laplacian loss, and a hand prediction loss. The vertex reconstruction loss is a mean squared error (MSE) loss. This enforces that points in the predicted mesh are close to points in the true mesh. The MSE loss is given by:

$$\mathcal{L}_v = \frac{1}{N} \sum_{i=1}^{N} \left\| v_i^{3\mathrm{D}} - \hat{v}_i^{3\mathrm{D}} \right\|_2^2 , \tag{5.8}$$

where $v_i^{3\mathrm{D}}$ and $\hat{v}_i^{3\mathrm{D}}$ are the ground truth and predicted vertex locations respectively.

The Laplacian loss is introduced to preserve the local surface smoothness of the mesh.

The Laplacian loss is given by:

$$\mathcal{L}_l = \frac{1}{N} \sum\nolimits_{i=1}^{N} \left\| \delta_i - \sum\nolimits_{v_k \in \mathcal{N}(v_i)} \delta_k \Big/ B_i \right\|_2^2, \tag{5.9}$$

where $\delta_i = v_i^{\mathrm{3D}} - \hat{v}_i^{\mathrm{3D}}$ is the offset between the ground truth and predicted vertex locations, $\mathcal{N}(v_i)$ is the set of neighbouring vertices of v_i, and B_i is the number of vertices in the set $\mathcal{N}(v_i)$.

The hand prediction loss is a binary cross entropy (BCE) loss. This is used to predict whether a hand is present in the image and is used for multi hand prediction tasks. The BSE loss is given by:

$$\mathcal{L}_b = \frac{1}{N} \sum_{i=1}^{N} y_i \cdot \log(x_i) + (1 - y_i) \cdot \log(1 - x_i), \tag{5.10}$$

where x is the model prediction and y is the true value.

For each dataset a different loss function can be used, where a hyperparameter is used as a weighting factor for each of the losses. The weighting factor used for each loss function for each dataset is provided in Table 5.1.

Dataset	M/S MSE Loss	Laplacian Loss	BCE Loss	R MSE Loss
Real-World	10^0	10^1	0	0
FreiHAND	10^0	10^1	0	0
InterHand 2.6M	10^3	0	10^0	10^{-1}

Table 5.1: Details on the hyperparameters weighting each loss function for each dataset. A hyperparameter of zero indicates that the loss function is not used for that dataset. In the heading row M stands for mesh, S stands for skeleton, and R stands for root. M/S MSE Loss is the loss function used for the reconstruction of the mesh or skeleton of the hand and R MSE Loss is the loss function used for a single 3D root position of the hand.

For each of the datasets used we also provide further training details in Table 5.2.

Dataset	Optimiser	Epochs	Initial lr	lr Decay	lr Decay Rate	Batch Size
Real-World	Adam	300	10^{-5}	0.5	$\forall 100$	8
FreiHAND	Adam	46	10^{-4}	0.1	38	32
InterHand 2.6M	Adam	20	10^{-5}	0.1	15 & 17	64

Table 5.2: Training details for each dataset. Details incldue the optimiser used, number of epochs trained for, initial learning rate (lr), decay factor of the lr, rate at which the lr is decayed, and the batch size used.

5.3.2 Experiments

5.3.2.1 Real-World Dataset

We experiment on the real-world dataset from Ge et al. (2019), which we split into 500 training and 83 testing examples of images and meshes. The input images are of size of 256×256 and the meshes to be predicted have 954 vertices. We report the mesh error as the evaluation metric, where the mesh error is the average error in Euclidean space between corresponding vertices in each generated 3D mesh and its ground truth 3D mesh, namely the mesh MSE loss in Section 5.3.1.6.

Choice of Decoder

We first experiment with the choice of decoder by using a non-rotation equivariant CNN encoder, vector mapping function and 3D projection function, and two different decoders. This ensures the model is inline with state-of-the art prior methods (Ge et al., 2019; Kulon et al., 2020) and allows for the comparison exclusively between decoders. The two decoders considered are:

- Multi-Layer Perceptron (MLP)
- Graph Neural Network (GNN)

The state-of-the art prior methods (Ge et al., 2019; Kulon et al., 2020) utilise a GNN decoder with node neighbourhoods chosen to be relatively small in comparison to the entire set of vertices in the output mesh. No consideration is given to the neighbourhood size used in the graph decoder and whether a global update could improve upon the local updates of a GNN. A MLP could be viewed as a fully connected GNN update with separate update on each edge. The main drawback of an MLP is the fixed size outputs of the model and in general GNNs are used due to the variable size of the data. Here when generating 3D meshes the output is of fixed size and hence one benefit of using a GNN is moot. The remaining advantage of using a GNN over an MLP is that the GNN has an inductive bias of locality. We experimentally compare using a GNN and MLP decoder in Table 5.3. This shows that the MLP decoder achieves a lower mesh error indicating the MLP model learns to generate more accurate meshes. This indicates that the inductive bias of locality does not improve mesh prediction over the MLP model. As we seek to build the inductive bias of rotational equivariance into our model, which aims to improve the generalisation of the vertex position prediction, we conclude that an MLP decoder is desirable given its improved performance.

Table 5.3: Average mesh error tested on the testing set of the real-world dataset Ge et al. (2019).

Method	Fixed orientation mesh error [mm]
MLP	**7.36**
GNN	8.08

Rotation Equivariant Model

We experiment with three different models two non-rotation equivariant models, named MLP and GNN, and a rotation equivariant model, named EMLP. The method named MLP has a non-rotation equivariant CNN encoder, vector mapping function and 3D projection function, and an MLP decoder. The method named GNN has a non-rotation equivariant CNN encoder, vector mapping function and 3D projection function, and a GNN decoder. Finally, the method named EMLP is the model detailed in Section 5.3.1; it has a rotation equivariant CNN encoder, vector mapping function, 3D projection function, and decoder, where each component is detailed in Figures 5.1, 5.2, 5.3, and 5.4 respectively. All three models have an identical training scheme, detailed in Section 5.3.1.6, and they are tested on the same 83 testing examples. Table 5.4 demonstrates that the model with an MLP based decoder performs better on the fixed orientation testing dataset than the model with GNN based decoder and the rotation equivariant EMLP model. One explanation for this is that the MLP is the most unconstrained and flexible of the three models. As a result, it is better able to learn to reconstruct hand meshes when the hand poses remain similar and in similar orientations. On the other hand, when considering a testing dataset which contains hands in a rotated orientation, both the MLP and GNN models perform very poorly, while the EMLP model maintains a good performance level. This demonstrates the improved generalisation of a model that accounts for suitable symmetries in the data. Overall, the consistency of the rotation equivariant model across all testing datasets results in it outperforming the other models for mesh reconstruction. This indicates our choice of model for generating 3D meshes is beneficial and rotation equivariance is a useful inductive bias to build into the model.

Table 5.4: Average mesh error on the testing set of the real-world dataset (Ge et al., 2019) in the column fixed orientation mesh error. Also, a rotated version of this testing dataset in the column rotated orientation mesh error, where the rotations used are $90°$, $180°$, and $270°$.

Method	Fixed orientation mesh error [mm]	Rotated orientation mesh error [mm]
MLP	**7.36**	136.27
GNN	8.08	140.52
EMLP	10.13	**10.82**

Following this, we conduct an ablation study to assess the benefit of each new component introduced within our model. We use the MLP baseline from Table 5.4, as it outperformed the GNN model, and similarly compare to the EMLP model, which is our fully rotation equivariant model comprising all components introduced in Section 5.3.1. Table 5.5 highlights the benefit of each component introduced in our model with the fully rotation equivariant model achieving a mesh error an order of magnitude lower than any other combination of model components on the rotated orientation testing dataset. This demonstrates that each new component introduced within our model is required for improved generalisation to new rotated orientations. Despite the better ability to reconstruct rotated hand meshes, the fully equivariant model under-performs the other models on the fixed orientation dataset. This is most likely a result of the model not being as flexible as any other combinations, due to each model having a similar number of parameters, which results in it not being able to as accurately capture the fixed orientation meshes.

Table 5.5: Average mesh error on the testing set of the real-world dataset (Ge et al., 2019) for fixed/rotated orientation mesh error. Conv is a non rotation equivariant convolutional encoder, EConv is a rotation equivariant convolutional encoder, MLP is a non rotation equivariant multi-layer perceptron, and EMLP is a rotation equivariant multi-layer perceptron.

Encoder	Projection function	Decoder	Fixed orientation mesh error [mm]	Rotated orientation mesh error [mm]
Conv	MLP	MLP	7.36	136.27
Conv	EMLP	EMLP	6.82	143.28
Conv	MLP	EMLP	6.94	138.93
Conv	EMLP	MLP	6.32	139.27
EConv	MLP	MLP	7.84	141.54
EConv	EMLP	MLP	7.27	143.99
EConv	MLP	EMLP	7.73	151.78
EConv	EMLP	EMLP	10.13	**10.82**

In addition to the performance of the three networks, we present the number of trainable parameters in Table 5.6. This shows that all three models have a similar number of trainable parameters. We aimed to build all three models such that they had a similar number of trainable parameters to ensure a fair comparison between each model.

Table 5.6: Number of trainable parameters in each component of the model in millions.

Method	Encoder	3D Projection	Decoder	Total
MLP	0.17	2.79	9.38	12.34
GNN	0.17	2.79	8.76	11.72
EMLP	0.62	2.81	8.92	12.35

Furthermore, we compare our rotation equivariant model to other prior works in Table 5.7.

This demonstrates that our method produces superior results to previous methods on a real-world testing dataset.

Table 5.7: Average mesh error tested on the testing set of the real-world dataset. Results for prior methods are taken from Ge et al. (2019).

Method	Mesh error [mm]
MANO-based Romero et al. (2017)	20.86
Direct LBS Cai et al. (2018)	13.33
Graph CNN Ge et al. (2019)	12.72
Ours	**10.13**

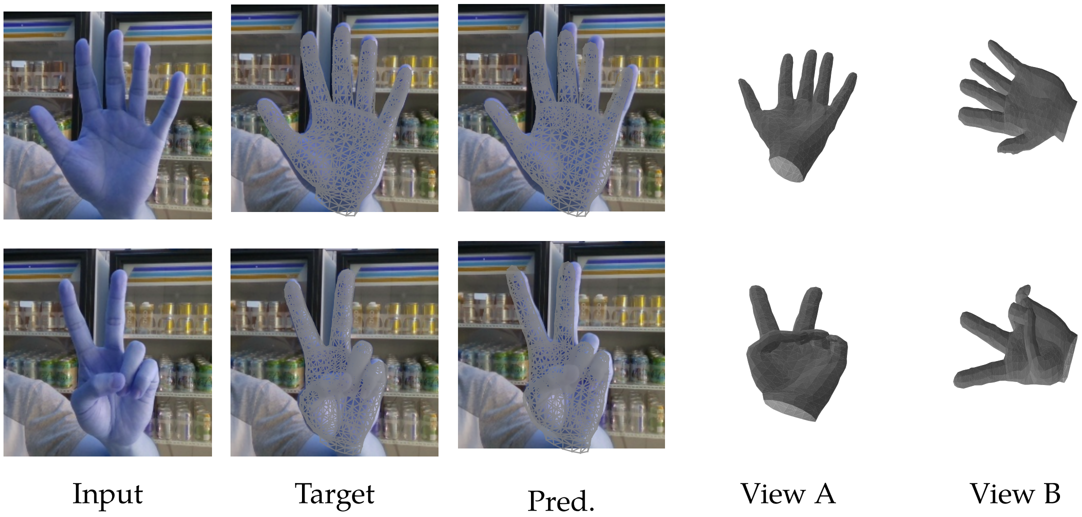


Figure 5.7: Qualitative mesh reconstruction results on the testing dataset from the real-world dataset of Ge et al. (2019) comparing our model's predictions to the target meshes.

Finally, we qualitatively validate our method by comparing the target mesh to the mesh predicted by our model in Figure 5.7. We present the image input into the model with both the target and predicted meshes projected onto the input image. Further, we show the mesh generated by our model for the image in two different view points. This shows that the generated meshes accurately capture the pose and shape required and the generated meshes are smooth. Further, we provide additional reconstruction in Figure C.1. In addition to qualitative analysis on the testing dataset, we also compare how our model performs to the two non-rotation equivariant models with MLP and GNN decoders on a testing dataset where the hands have been rotated by $90°$. As is demonstrated in Figure 5.8 our EMLP model generates meshes that accurately capture the pose and and shape of the hand. On the contrary, the MLP and GNN model fail to capture the pose correctly and do not predict realistic looking hand meshes in many cases. This result is an effect of the type of inductive bias that we chose to build into our model. These results demonstrate that the inductive bias of locality does not assist the model in generalising the hands in orientations outside of the training dataset. Further, the MLP

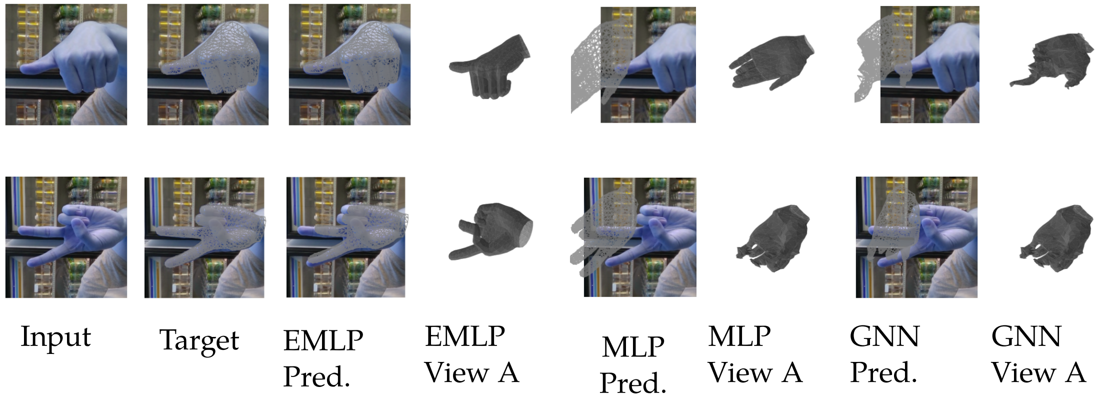


Figure 5.8: Qualitative mesh reconstruction results on rotated testing hand images from the testing dataset for the EMLP, MLP, and GNN models.

model with no inductive bias appears to generate more realistic looking hands than the GNN model, despite generating hands of incorrect pose. Comparing both models to our EMLP model highlights the value of building useful inductive bias into the model, especially those that reflect some known symmetry in the data.

5.3.2.2 FreiHAND Dataset

FreiHAND (Zimmermann et al., 2019) is a dataset comprising of 2D images accompanied with the corresponding 3D hand meshes and hand pose data. It consists of 130,240 training examples with 3,960 examples reserved for testing. It contains difficult poses and interactions with objects, which makes this a more challenging dataset than the real-world one. The 3D meshes are generated by iteratively fitting a hand shape model and therefore are not as accurate as in the real-world dataset. As a result of the hand-object interactions within the dataset, we expect the encoder to be more important, as the images contain more complex information, such as the varied objects and occlusions of the hands. Nevertheless, the dataset is a widely used benchmark, and as such we also test our method on it.

We modified the model presented in Section 5.3.1 in line with the model presented by (Chen et al., 2021) by adding a second encoder, while maintaining each of our rotation equivariant components. The second encoder predicts a segmentation map of the hand that is concatenated with the input features of the original encoder. The original model then operates as describes in Section 5.3.1. The addition of the second encoder is to give the model more capacity in the encoder to be able to overcome the additional complexity in the input images.

We compare to previous methods in Table 5.8, where we consider models that have a similar capacity to our model, namely those with a ResNet 18, Resnet 50 or equivalent encoder. Due to the diversity of the backgrounds and objects in the hand images, we believe that pre-training the encoder on a large image dataset is required to encode useful hand information. This is due to a large pre-trained encoder already containing the necessary weights to identify objects that are present in addition to the hands, whilst an encoder that has not been pre-trained will lack this prior information. We pretrained our rotation equivariant encoder on imagenet using the ffcv package (Leclerc et al., 2022) for 24 epochs. Typically, ResNet models are trained on imagenet for more epochs and therefore it is likely our model would perform better if the pretraining was conducted for longer. For comparison we also show the result of our model when trained with an encoder that has not been pretrained on imagenet. The results demonstrate that our method when the encoder is pretrained is competitive with state-of-the-art methods trained using 4 and 8 GPUs despite our method only using 1 GPU, which suggests the generalisation benefit of rotation equivariance is a promising direction.

Table 5.8: Performance comparison with previous methods on FreiHAND dataset (Zimmermann et al., 2019). Methods in comparison are: MANO (Romero et al., 2017), Hasson et al (Hasson et al., 2019), Boukhayma et al (Boukhayma et al., 2019), FreiHAND (Zimmermann et al., 2019), Kulon et al (Kulon et al., 2020), Pose2Mesh (Choi et al., 2020), I2LMeshNet (Moon & Lee, 2020), CMR-G (Chen et al., 2021)

Method	Encoder	PA-MPVPE ↓	F@5mm ↑	F@15mm ↑
MANO		14.4	0.416	0.880
Hasson et al	ResNet18	13.2	0.436	0.908
Boukhayma et al	ResNet50	13.0	0.435	0.898
FreiHAND	ResNet50	10.7	0.529	0.935
Kulon et al	ResNet50	8.6	0.614	0.966
Pose2Mesh	ResNet18	7.8	0.674	0.969
CMR-G	ResNet18	7.6	-	-
I2LMeshNet	ResNet50	7.6	0.681	0.973
Ours - pre-train	EqResNet18	7.9	0.663	0.968
Ours - no pre-train	EqResNet18	8.9	0.608	0.957

We also provide examples of the mesh reconstructions precdicted by our model overlayed onto the input hand images in Figure 5.9. This demonstrates that our model can accurately reconstruct the hand meshes with more challenging background and with objects within the hands. We provide further reconstruction in Figure C.3.

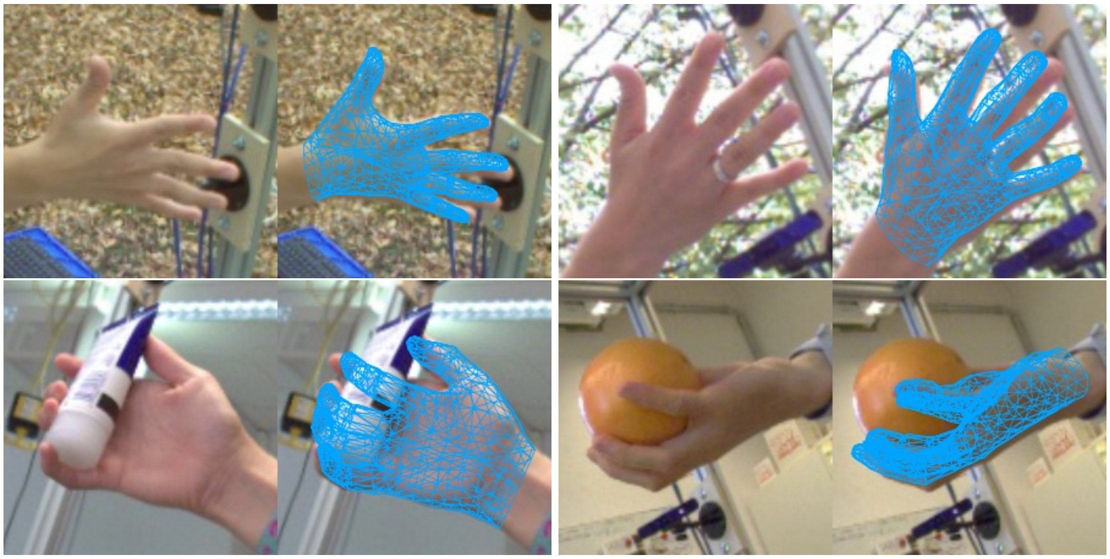


Figure 5.9: Qualitative mesh reconstruction results on the testing dataset from the FreiHand dataset of Zimmermann et al. (2019).

5.3.2.3 InterHand2.6m Dataset

We also experiment on the InterHand2.6m dataset (Moon et al., 2020). This dataset comprises of 2D images and accompanying 3D hand pose. Therefore, instead of reconstructing the 3D hand mesh, here the model is required to predict the 3D hand pose from the image. Each pose is a skeleton of the hand consisting of 21 3D points. Further, each image can consist of one or two hands, and when there are two hands in the image they can be interacting. Therefore, the task is to predict the 42 3D points comprising both hands, the root depth, and whether each hand is present in the image. As a result, we modify our decoder for this dataset such that it outputs 42 3D points, the root depth, and binary prediction of whether each hand is present. This requires changing the output vector space of the decoder from $954 \bigotimes T_1$ to $42 \bigotimes T_1 \bigoplus 1 \bigotimes T_0 \bigoplus 2 \bigotimes T_0$.

The dataset consists of 1.4 million training examples, 380 thousand validation examples, and 849 thousand testing examples. This makes it a large dataset, especially in comparison to the real-world dataset. Similarly to the FreiHAND dataset, occlusion can occur in the images when one hand overlaps the other.

We compare to previous methods in Table 5.9. This demonstrates that our method performs competitively with leading methods, ranking 3rd best in terms of joint predictive accuracy. A key difference between our architecture and the majority of other methods, including NRSfM (Deng et al., 2022) and IntagHand (Li et al., 2022), is that they use a ResNet50 encoder pre-trained on Imagenet (Deng et al., 2009) and the equivalent rotation equivariant pre-trained ResNet50 model does not exist. As a result, our encoder is a rotation equivariant ResNet18 model, which is not pre-trained. Therefore, the base starting point of our encoder has a worse ability to detect and classify objects in images. As a result, we believe our model could generate more accurate hand poses by making use of a larger pre-trained encoder.

Table 5.9: Performance comparison with previous methods on InterHand2.6m dataset (Moon et al., 2020). Methods in comparison are: Zimmermann et al (Zimmermann & Brox, 2017), Zhou et al (Zhou et al., 2020), IHMR (Rong et al., 2021), Boukhayma et al (Boukhayma et al., 2019), Moon et al (Moon et al., 2020), Spurr et al (Spurr et al., 2018), Zhang et al (Zhang et al., 2021), PRN (Park et al., 2020), DNRSfM (Kong & Lucey, 2019), C3dpo (Novotny et al., 2019), NRSfM (Deng et al., 2022), and IntagHand (Li et al., 2022)

Method	MPJPE ↓
Zimmermann et al	36.4
Zhou et al	23.5
IHMR	17.1
Boukhayma et al	16.9
Moon et al	16.9
Spurr et al	15.4
DNRSfM	13.8
Zhang et al	13.1
C3dpo	9.8
NRSfM	8.9
IntagHand	8.8
Ours	9.3

5.3.3 Discussion

The results show that enforcing rotational symmetries within the model ensures the predicted hand meshes transform in a predictable way under rotation and that no undesirable deformations occur in the predicted meshes. The advantage of this is most noticeable when training on a small dataset, which is likely when using real world data due to the cost of data collection. This implies that incorporating rotational symmetries into hand mesh prediction models improves the robustness of such models, ensuring predicted pose meshes as stable under rotation.

5.4 Conclusions

We present a novel framework for generating 3D hand meshes from 2D RGB images that is composed of equivariant layers that ensure the entire model is rotation equivariant. To the best of our knowledge this is the first model for generating 3D meshes from 2D RGB images that considers equivariance. We provide a comparison between and MLP and GNN based decoder, showing that an MLP model generates more accurate smoother meshes than a GNN model. This improvement in mesh generation justifies our choice of using an MLP based model. Further, we demonstrate that rotation equivariance is a suitable inductive bias to build into each component of the model by outperforming, in terms of reconstruction accuracy, other leading methods on a real-world dataset. In addition, this improves robustness by removing undesirable deformations of the generated meshes under rotation of the input image, which we both quantitatively and qualitatively compare with non-rotation equivariant models. Furthermore, we demonstrate competitive mesh reconstruction ability with state-of-the-art methods on two widely used, large scale, benchmark datasets. The theoretical match of prior knowledge about the problem, and the improved empirical performance compared to the state of the art, support the use of these new inductive biases in models generating 3D hand meshes from 2D RGB images.

6

Automorphism Symmetries in Subgraph Graph Neural Networks

In this chapter we develop a novel graph neural network that is equivariant to automorphism groups of subgraphs. This model initially comprises of a subgraph selection policy to split the input graphs into a collection of subgraphs, which uniquely stores subgraphs in separate bags based on their automorphism groups. The input features of the subgraphs are updated through multiple automorphism equivariant layers, before the subgraphs are pooled and graph level features are output. We demonstrate that the model is more scalable than global permutation equivariant models. We also show that the model performs competitively across a range of graph benchmark tasks.

6.1 Introduction

Machine learning on graphs has received much interest in recent years with many graph neural network (GNN) architectures being proposed. One such method, which is widely used, is the general framework of message passing neural networks (MPNN). In an MPNN, the graph consists of a set of nodes which have a feature space associated to them, and an adjacency matrix which details the connectivity of the graph. It is also possible to have edge feature spaces such that the edges do not only provide connectivity, but also have feature vectors associated to them. Then in an MPNN, node feature vectors in each layer of the network are updated according to

$$m_v = \sum_{n \in \mathcal{N}(v)} M(h_v, h_n, e_{vn}) \tag{6.1}$$

and

$$h_v = U(h_v, m_v), \tag{6.2}$$

where $\mathcal{N}(v)$ is the set of nodes in the neighbourhood of nodes v, h is the vector space of nodes in a specific hidden layer, and e is the feature vector space of the edge connecting two nodes. As a result MPNNs provide both a useful inductive bias and scalability across a range of domains (Gilmer et al., 2017b).

However, Xu et al. (2019); Morris et al. (2019) showed that models based on a message passing framework with permutation invariant aggregation functions, for example the summation in Equation 6.1, have expressive power at most that of the Weisfeiler-Lehman (WL) graph isomorphism test (Weisfeiler & Leman, 1968). Therefore, there exist many non-isomorphic graphs that a model of this form cannot distinguish between.

Many methods have been proposed to design a GNN that improves the expressive power of MPNNs. Although—most often—an increase in expressivity must be traded off against scalability. We present the background into existing methods which attempt to tackle this question in Section 6.2.

We build on the developments of permutation equivariance to design a framework to create provably more expressive and scalable graph networks. We achieve this through incorporating symmetry structures in graphs, by considering a graph equivariant update function which operates over subgraphs. Our framework, *Subgraph Permutation Equivariant Networks* (*SPEN*), is developed from the observation that operating on subgraphs both improves the scalability and expressive power of higher-dimensional GNNs, whilst also unlocking a natural choice of automorphism group, further increasing the expressive power of the network. Our framework consists of:

1. encoding the graph as a bag of bags of subgraphs
2. utilising a k-order permutation equivariant base encoder
3. constraining the linear map to be equivariant to the automorphism groups of the bags of subgraphs

Subgraphs each have a symmetry group and our framework captures this in two ways. Each subgraph has a permutation symmetry, which is induced by a permutation of the nodes in the graph. In addition, there is a symmetry across subgraphs whereby subgraphs are associated to an automorphism group. We therefore construct a neural network comprising of layers that are equivariant to both permutations of nodes and the automorphism groups of subgraphs. We achieve this by utilising a permutation equivariant base encoder with feature space constrained by the direct sum of different order permutation representations. Further, we constrain the linear map within each layer of the network to be equivariant to the automorphism groups of the bags of subgraphs. This means that subgraphs belonging to different automorphism groups are processed by a kernel with different weights, while for subgraphs belonging to the same automorphism group the kernel shares weights. This leads to us creating a subgraph extraction policy which generates a bag of bags of subgraphs, where each bag of subgraphs corresponds to a different subgraph automorphism group. Our main contributions within this chapter are:

- an automorphism equivariant compatible subgraph extraction method.
- a novel choice of automorphism groups with which to constrain the linear map to be equivariant to
- a more scalable framework for utilising higher-dimensional permutation equivariant GNNs
- a more expressive model than higher-dimensional permutation equivariant GNNs and subgraph MPNNs
- A theoretical analysis of the proposed model in terms of scalability and an application of the theoretical analysis from Bevilacqua et al. (2022) and de Haan et al. (2020) to demonstrate the expressivity of our model.
- a demonstration that our method is statistically indistinguishable from state-of-the-art methods on benchmark graph classification tasks.

Throughout this chapter we are following the notation presented by (Bevilacqua et al., 2022) for how we present what a subgraph is and for the expressivity analysis.[1]

6.2 Background

6.2.1 Graph Neural Network Definitions

In this work we consider graphs as concrete graphs and utilise subconcrete graphs in our framework. The subgraphs are extracted as k-ego network subgraphs. A concrete graph is defined as:

Definition 20. A *Concrete Graph* (de Haan et al., 2020) G is a finite set of nodes $\mathcal{V}(G) \subset \mathbb{N}$ and a set of edges $\mathcal{E}(G) \subset \mathcal{V}(G) \times \mathcal{V}(G)$.

The set of node IDs may be noncontiguous and we make use of this here as we extract overlapping subgraphs and perform the graph update function on bags of subgraphs.

[1] One of contributions of this chapter namely operating on subgraphs, which inherit ids from the original graph was developed concurrently to (Bevilacqua et al., 2022), see `https://openreview.net/forum?id=7oyVOECcrt` for the initial version of our work.

The natural numbers of the nodes are essential for representing the graphs in a computer, but hold no information about the underlying graph. Therefore, the same underlying graph can be given in may forms by a permutation of the ordering of the natural numbers of the nodes. Throughout the paper we refer to concrete graphs as graphs to minimise notation.

In practice, graphs can have both node and edge features, which are both vector spaces. Throughout this chapter we represent graphs in tensor format. This is due to the use of tensor format making the notation connecting graph feature spaces and the group representations acting on them easier.

Definition 21. In tensor format the values of G are encoded in a tensor $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}(G)| \times |\mathcal{V}(G)| \times d}$, where the node features are encoded along the diagonal and edge features are encoded in off-diagonal positions.

Following our use of concrete graphs we also make use of subconcrete graphs, which are defined as:

Definition 22. A *subconcrete Graph H* is created by taking a node $i \in \mathcal{V}(G)$, and extracting the nodes $j \in \mathcal{V}(G)$ and edges $(i, j) \subset \mathcal{V}(G) \times \mathcal{V}(G)$, according to some subgraph selection policy.

An important part of the developed framework in this chapter is the use of subgraphs. We consider the subgraph selection policy as a k-ego-network policy. For brevity we refer to subconcrete graphs as subgraphs throughout the paper.

Definition 23. A *k-Ego Network* of a node is its k-hop neighbourhood with induced connectivity.

This chapter focuses on symmetries between graphs. For this we considered the automorphism symmetries of graphs through the consideration of the automorphism group. Here we consider the automorphism group of the permutation group by considering graphs with the same structure up to some permutation as part of the same automorphism group. We define a naturality constraint for a linear map as this governs a symmetry or equivariance condition up to graph isomorphism.

Definition 24. The *naturality* constraint for a linear map states that for a graph G and linear map $f_G : \rho(G) \to \rho'(G)$ the following condition holds for every graph isomorphism ϕ (de Haan et al., 2020):

$$\rho'(\phi) \circ f_G = f_{G'} \circ \rho(\phi).$$

When considering the permutation symmetries of a graph and looking at the automorphism group, there can be no automorphic mapping between a graph with four nodes and a graph with five nodes. This is due to there being no permutation of the four nodes features such that they yield the five nodes features.

In this chapter we are interested in the symmetries of the symmetric group S_n. This constraint can be solved for different order tensor representations (Maron et al., 2018; Finzi et al., 2021). We present the space of linear layers mapping from k-order representations to k'-order representations in Figure 6.4.

6.2.2 Expressive Graph Neural Networks

More expressive graph neural networks (GNNs) exist which can be grouped into three different groups:

1. those which design higher-dimensional GNNs
2. those which use positional encodings through pre-coloring nodes
3. those which use subgraphs/local equivariance

Several architectures have been proposed of the type 1, which design a high-order GNN equivalent to the hierarchy of k-WL tests (Maron et al., 2018, 2019; Morris et al., 2019, 2020b; Keriven & Peyré, 2019; Azizian & Lelarge, 2021). Despite being equivalent to the k-WL test, and therefore having provably strong expressivity, these models lose the advantage of locality and linear complexity. As such, the scalability of such models poses an issue for their practical use. Maron et al. (2018) showed that the basis space for permutation equivariant models of order k is equal to the $2k^{th}$ Bell number, which results in a basis space of size 2 for order-1 tensors, 15 for order-2 tensors, 203 for order-3 tensors, and 4140 for order-4 tensors, demonstrating the practical challenge of using higher-dimensional GNNs. Several architectures have also been proposed of type 2 where authors seek to introduce a pre-coloring or positional encoding that is permutation invariant. These comprise of pre-coloring nodes based on pre-defined substructures (Bouritsas et al., 2022) or lifting graphs into simplicial- (Bodnar et al., 2021b) or cell- (Bodnar et al., 2021a) complexes. These methods require a pre-computation stage, which in the worst-case, finding substructures of size k in a graph of n nodes, is $\mathcal{O}(n^k)$. Finally, subgraphs/local equivariance of type 3 have been considered to find more expressive GNNs. Local graph equivariance requires a (linear) map that satisfies an automorphism equivariance constraint. This is due to the nature of graphs having different local symmetries on different nodes/edges. This has been considered by de Haan et al. (2020) though imposing an isomorphism/automorphism constraint on edge neighbourhoods and by Thiede et al. (2021) and Xu et al. (2021) by selecting specific automorphism groups and lifting the graph to these. Although the choice of automorphism group chosen by de Haan et al. (2020) leads to little weight sharing and requires the automorphism constraint to be parameterized, while those proposed by Thiede et al. (2021) and Xu et al. (2021) do not guarantee to capture the entire graph and requires a hard-coded choice of automorphism group. Xu et al. (2021) also propose a method for searching across different subgraph templates, although this still requires some hard-coding of the subgraph structures. Operating on subgraphs has been considered as a means to improve GNNs by dropping nodes (Papp et al., 2021; Cotta et al., 2021), dropping edges (Rong et al., 2020), utilising ego-network graphs (Zhao et al., 2022), and considering the symmetry of a bag of subgraphs (Bevilacqua et al., 2022).

6.2.3 Previous Methods

Here we provide further details on some of the key models from the previous section. This aims to help position our model with respect to other methods. The methods detailed below generally fall into model types 1 and 3 above.

6.2.3.1 Global Equivariant Graph Networks

Global Permutation Equivariance. Global permutation equivariant models have been considered by Hartford et al. (2018), Maron et al. (2018), Maron et al. (2019), and Albooyeh et al. (2019), with Maron et al. (2018) demonstrating that for order-2 layers there are 15 operations that span the full basis for an permutation equivariant linear layer. These 15 basis elements are shown in Figure 6.4 with each basis element given by a different color in the map from representation $\rho_2 \rightarrow \rho_2$. Despite these methods, when solved for the entire basis space, having expressivity as good as the k-WL test, they operate on the entire graph. Operating on the entire graph features limits the scalability of the methods. In addition to poor scalability, global permutation is a strong constraint to place upon the model. In the instance where the graph is flattened and an MLP is used to update node and edge features the model would have n^4 trainable parameters, where n is the number of nodes. On the other hand, a permutation equivariant update has only 15 trainable parameters and in general $15 \ll n^4$.

Viewing a global permutation equivariant graph network from a category theory perspective there is one object with a collection of arrows representing the elements of the group. Here the arrows or morphisms go both from and to this same single object. The feature space is a functor which maps from a group representation to a vector space. For a global permutation equivariant model the same map is used for every graph.

$$\begin{tikzcd}
G \arrow[loop above, "g_1"] \arrow[loop left, "g_2"] \arrow[loop right, "e"]
\end{tikzcd}$$

Symmetric Group

Global Naturality Global natural graph networks (GNGN) consider the condition of naturality (de Haan et al., 2020). GNGNs require that for each isomorphism class of graphs there is a map that is equivariant to automorphisms. This naturality constraint is given by the condition $\rho'(\phi) \circ K_G = K_{G'} \circ \rho(\phi)$, which must hold for every graph isomorphism $\phi : G \to G'$ and linear map K_G (de Haan et al., 2020). While the global permutation equivariance constraint requires that all graphs be processed with the same map, global naturality allows for different, non-isomorphic, graphs to be processed by different maps and as such is a generalisation of global permutation equivariance. As is the case for global permutation equivariant models, GNGNs scale poorly as the constraint is placed over the entire graph and linear layers require global computations on the graphs.

Viewing a GNGN from a category theory perspective, (de Haan et al., 2020), there is a different object for each concrete graph, which form a groupoid. Then, there is a mosphism or arrow for each graph isomorphism. These can either be automorphisms, if the arrow maps to itself, or isomorphisms, if the arrow maps to a different object. The feature spaces are functors which map from this graph category to the category of vector spaces. The GNG layer is a natural transformation between such functors consisting of a different map for each non-isomorphic graph.

$$\begin{tikzcd}
G_1 \arrow[loop above] \arrow[loop left] \arrow[rr, shift right] \arrow[dr, shift left] & & G_2 \arrow[loop above] \arrow[loop right] \arrow[ll, shift right] \arrow[dl, shift left] \\
& G_3 \arrow[loop left] \arrow[loop below] \arrow[ul, shift left] \arrow[ur, shift left] &
\end{tikzcd}$$

Groupoid of Concrete Graphs
(de Haan et al., 2020)

6.2.3.2 Local Equivariant Graph Networks

Local equivariant models have started to receive attention following the successes of global equivariant models and local invariant models. The class of models that are based on the WL test are not in general locally permutation equivariant, in that they still use a message passing model with permutation invariant update function. Despite this, many of these models inject permutation equivariant information into the feature space, which improves the expressivity of the models (Bouritsas et al., 2022; Morris et al., 2020b; Bodnar et al., 2021b,a). The information to be injected into the feature space is predetermined in these models by a choice of what structural or topological information to use.

Covariant Compositional Networks In contrast to using results from the WL test covariant compositional networks look at permutation equivariant functions, but they do not consider the entire basis space as was considered in Maron et al. (2018). Instead they consider four equivariant operations (Kondor et al., 2018). This means that the permutation equivariant linear layers are not as expressive as those used in the global permutation equivariant layers. Furthermore, in a covariant compositional networks the node neighbourhood and feature dimensions grow with each layer, which can be problematic for larger graphs and limits their scalability.

Local Naturality A local natural graph network (LNGN) (de Haan et al., 2020) uses a message passing framework. The increased expressivity comes from the specification of the constraint that node features transform under isomophisms of the node neighbourhood. Therefore, a different message passing kernel is used on non-isomorphic edges. In practice, this leads to little weight sharing in graphs that are quite heterogeneous and as such, the layer is re-interpreted such that a message from node p to node q, $k_{pq}v_p$, is given by a function $k(G_{pq}, v_p)$ of the edge neighbourhood G_{pq} and feature value v_p at p. In comparison to our method, LNGN amounts to choosing a different automorphism group, where an LNGN is equivariant to the edge subgraph when performing the message passing. On the other hand, we use the k-ego network subgraphs of each node in the graph. As was noted by the authors of LNGN, their choice of automorphism group leads to little weight sharing and requires parameterising. For some datasets, namely those which are particularly heterogeneous, our choice of automorphism group requires parameterising. In practice, this was not required on most datasets, which allows us to use the true automorphism group. We also utilise a different base update function in comparison to LNGNs, where we use a higher-order permutation equivariant update function and an LNGN uses a GCN.

Viewing an LNGN from a category theoretic perspective (de Haan et al., 2020), there is a groupoid of node neighbourhoods — where morphisms are isomorphisms between node neighbourhoods — and a groupoid of edge neighbourhoods — where morphisms are isomorphisms between edge neighbourhoods. In addition, there is a functor mapping from edge neighbourhoods to the node neighbourhood of the start node, and a functor mapping similarly, but to the tail node of the edge neighbourhood. The node feature spaces are functors mapping from the category of node neighbourhoods to the category of vector spaces. Further, composition of two functors creates a mapping from edge neighbourhoods to the category of vector spaces. An LNG kernel is a natural transformation between these functors.

$$\begin{tikzcd}
N_1 \arrow[loop above] \arrow[loop below] \arrow[rr, shift right] \arrow[dr, shift left] & & N_2 \arrow[loop above] \arrow[loop right] \arrow[ll, shift right] \arrow[dl, shift left] \\
& N_3 \arrow[loop left] \arrow[loop right] \arrow[ul, shift left] \arrow[ur, shift left] &
\end{tikzcd}$$

Groupoid of Node Neighbourhoods
(de Haan et al., 2020)

$$\begin{tikzcd}
E_1 \arrow[loop above] \arrow[loop below] \arrow[rr, shift right] \arrow[dr, shift left] & & E_2 \arrow[loop above] \arrow[loop right] \arrow[ll, shift right] \arrow[dl, shift left] \\
& E_3 \arrow[loop left] \arrow[loop right] \arrow[ul, shift left] \arrow[ur, shift left] &
\end{tikzcd}$$

Groupoid of Edge Neighbourhoods
(de Haan et al., 2020)

Autobahn Another local equivariant graph network, which makes use of an automorphism group symmetry, is that of Autobahn (Thiede et al., 2021). Autobahn uses the automorphism groups of cycles and paths. This choice again differs from the automorphism group that we use. A key difference can be seen that our method is general in the way it can be used on any graph dataset, while Autobahn is designed specifically for molecular datasets. Our choice of automorphism group and subgraph selection policy ensures that the information of every node is made use of in the model, which Autobahn's choice of automorphism group can lead to nodes being ignored.

Equivariant Subgraph Aggregation Networks ESAN (Bevilacqua et al., 2022) is another method that utilises subgraphs. ESAN considers a range of different subgraph selection policies and explores the expressive power of each. In contrast our approach considers one subgraph selection policy, which is specifically chosen due to it leading to a natural choice of automorphism equivariance constraint that we found requires less parameterisation than previous works. This presents a key different between our approach and ESAN, namely that we consider automorphism groups of subgraphs while

ESAN does not. A further difference is that we integrate higher-dimensional permutation equivariant feature spaces into our networks.

k-reconstruction GNNs k-reconstruction GNNs (Cotta et al., 2021) also consider subgraphs, although here vertex removed subgraphs are considered, which is different to the subgraph choice we make. Removing single nodes from a graph would not yield the same improvement in scalability as we demonstrate in our method due to each subgraph being almost the same size as the original graph, while we demonstrated using k-ego network subgraphs yields subgraphs which are much smaller than the original graph in practise. Furthermore, in our work we show how the combination of subgraph selection policy, automorphism equivariance constraint, and higher-order permutation GNN update functions improves expressivity beyond 1-WL, while k-reconstruction GNNs compare to 1-WL.

GRaph AutomorPhic Equivalence network The GRaph AutomorPhic Equivalence network (GRAPE) model (Xu et al., 2021) is somewhat similar to Autobahn in that they use subgraph templates to select the automorphism constraint, although it appears that for GRAPE this is chosen to be more general than for Autobahn. This still differs from our subgraph selection policy and has to be pre-determined before building a model. On the other hand, our method, using a k-ego network policy, has the automorphism constraint driven by the data, ensuring the approach is applicable across a range of graph datasets.

6.3 Subgraph Permutation Equivariant Networks

6.3.1 Model Specification

We first outline necessary definitions that our method builds upon. Following this we present the two main concepts of our SPEN model. Firstly, SPEN has a k-ego net subgraph extraction method, which are collected into bags determined by their automorphism group. Secondly, SPEN has an automorphism equivariant graph neural network. This chapter presents the core concepts of the model which contribute to the improved scalability and expressivity. The overall architecture of the SPEN model is presented in Figure 6.1. Further, the breakdown of an automorphism equivariant layer within our framework is presented in Figure 6.2. In addition, in Figure 6.3 we detail an example breakdown of one of the mapping functions used within our model. Finally, Figure 6.4 shows a visualisation of some of the bases used within the mapping functions in the automorphism equivariant functions. The key definitions required to understand our framework are provided in Chapter 6.2.1.

6.3.1.1 Subgraph Selection Policy

subgraphs can be extracted from a graph in a number of ways, by removing nodes, by removing edges, extracting connectivity graphs at nodes, or extracting connectivity graphs at edges to name a few. In this work we focus on k-ego network subgraphs. These are subgraphs extracted by considering the k-hop connectivity of the graph at a selected node and extracting the induced connectivity. The subgraph selection policy of k-ego networks therefore extracts a subgraph for each node in the original graph.

In this work we process graphs as bags of subgraphs, where the notation of a bag of subgraphs was introduced by Bevilacqua et al. (2022). In general the size of the subgraphs, $|H|$, extracted for a graph are not all the same size, and thus $|H|$ varies from subgraph to subgraph. We therefore go further than representing each graph as a bag of subgraphs and represent each graph as a bag of bags of subgraphs, where each bag of subgraphs hold subgraphs of the same size, i.e. S_{H^i} is the bag of subgraphs for subgraphs of size $|H| = i$. The graph is therefore represented as the bag of bags of subgraphs $G \equiv \{\{S_{H^i} \ldots S_{H^k}\}\} \equiv \{\{\{\{H_1^i, ..., H_a^i\}\}, ..., \{\{H_1^k, ..., H_c^k\}\}\}\}$, for subgraphs H, with bags of subgraphs which are each of size $a, ..., c$ containing subgraphs of sizes $i, .., k$ respectively.

In Chapter 6.3.2 we demonstrate how our choice of subgraph selection policy improves the expressivity of the overall model. Further, in Chapters 6.3.2 and 6.3.3 we show that the choice of subgraph selection allows the method to scale better than global approaches. Finally, in Chapter 6.3.3.1 we show that using k-ego network subgraphs yields subgraphs which are typically much smaller than the original graph. This both feeds into the improved ability of our method to scale to larger graphs and provides a smaller more compact automorphism group compared to an approach such as node removed subgraphs. As a result, our method requires less parameterisation of the automorphism group than other automorphism equivariant approaches.

6.3.1.2 Automorphism Equivariant Graph Network Architecture

The input data represented as a bag of bags of subgraphs has a symmetry group of both the individual subgraphs and of the bags of subgraphs. We construct a graph neural network that is equivariant to this symmetry. This can be broken down into three parts:

1. the automorphism symmetry of the bags of subgraphs
2. the permutation symmetry of subgraphs within bags of subgraphs
3. the original graph permutation symmetry

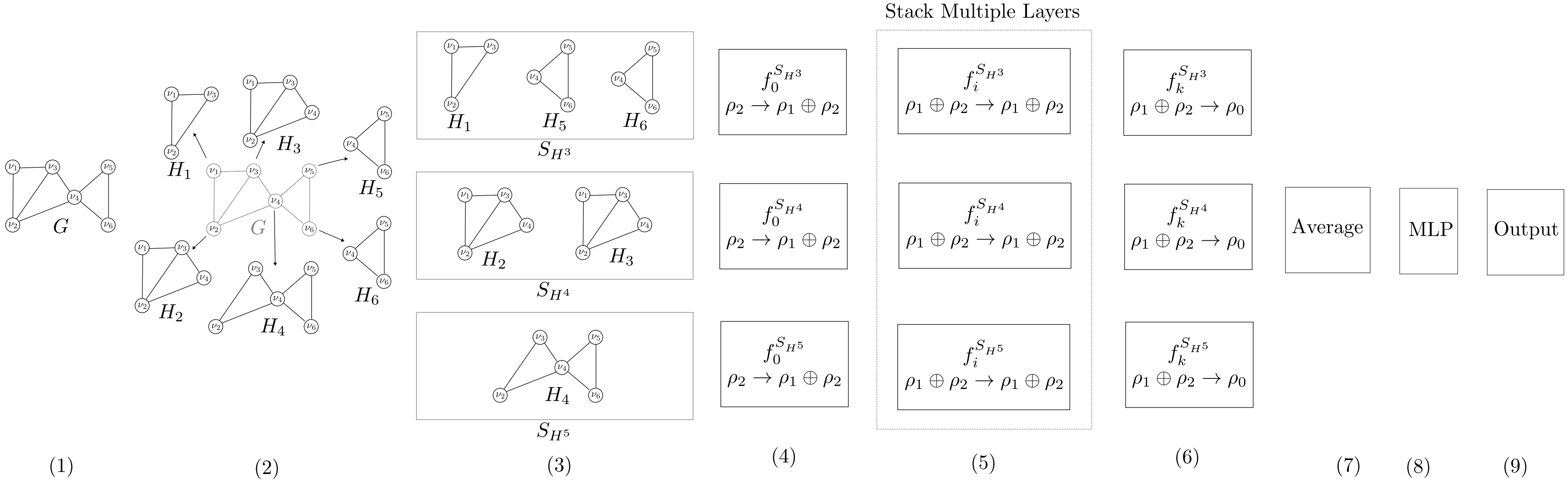
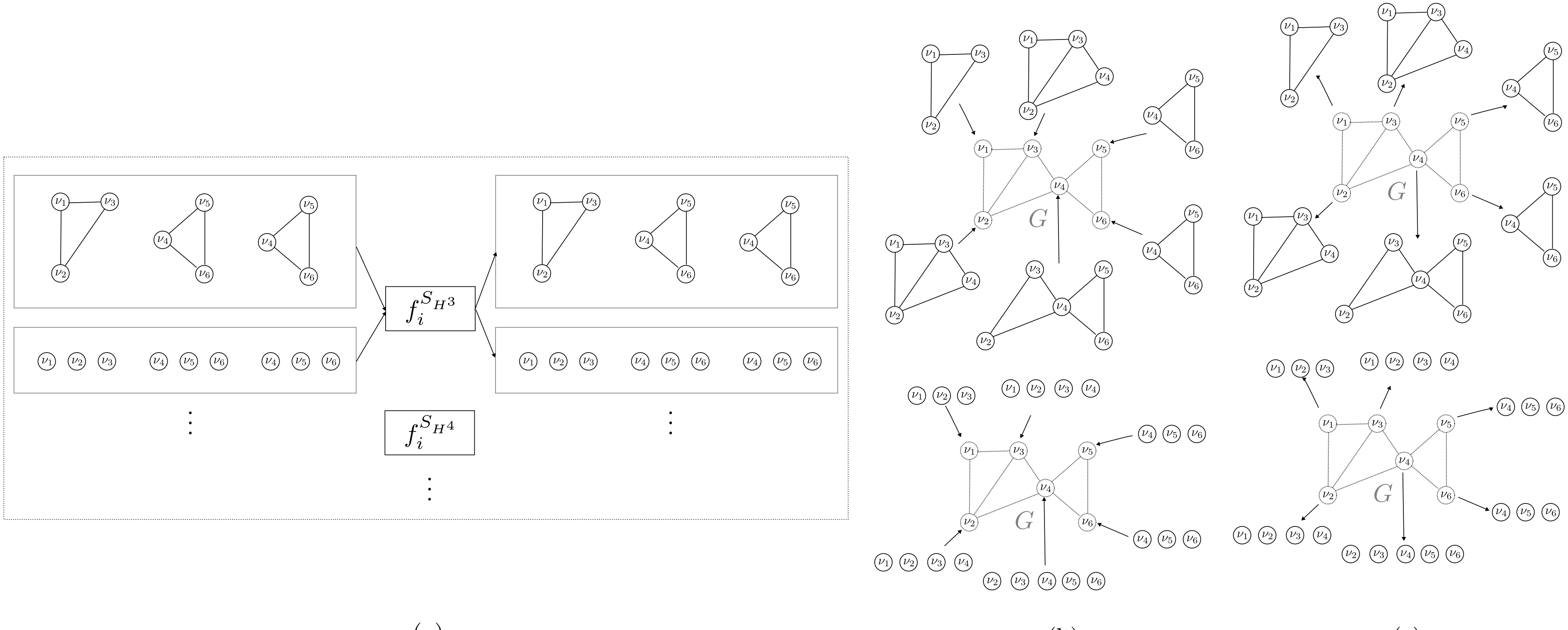

G
H_1
H_2
H_3
H_4
H_5
H_6
ν_1 ν_2 ν_3 ν_4 ν_5 ν_6
S_{H^3}
S_{H^4}
S_{H^5}
$f_0^{S_{H^3}}$ $\rho_2 \to \rho_1 \oplus \rho_2$
$f_0^{S_{H^4}}$ $\rho_2 \to \rho_1 \oplus \rho_2$
$f_0^{S_{H^5}}$ $\rho_2 \to \rho_1 \oplus \rho_2$
Stack Multiple Layers
$f_i^{S_{H^3}}$ $\rho_1 \oplus \rho_2 \to \rho_1 \oplus \rho_2$
$f_i^{S_{H^4}}$ $\rho_1 \oplus \rho_2 \to \rho_1 \oplus \rho_2$
$f_i^{S_{H^5}}$ $\rho_1 \oplus \rho_2 \to \rho_1 \oplus \rho_2$
$f_k^{S_{H^3}}$ $\rho_1 \oplus \rho_2 \to \rho_0$
$f_k^{S_{H^4}}$ $\rho_1 \oplus \rho_2 \to \rho_0$
$f_k^{S_{H^5}}$ $\rho_1 \oplus \rho_2 \to \rho_0$
Average
MLP
Output
(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)

Figure 6.1: (1-2) Splitting the graph into subgraphs. (3) Place subgraphs into bags , where each bag holds subgraphs of a specific size. (4) Process the bags of subgraphs with an automorphism equivariant linear map. (5) Stack multiple layers each comprising of an automorphism equivariant mapping function. (6) Add a final automorphism equivariant mapping function. (7) Each automorphism group is averaged. (8) An MLP is used to update the feature space.



(a) (b) (c)

Figure 6.2: This figure breaks down what a single automorphism equivariant layer within our model looks like. In Figure 6.1 this corresponds to looking inside a single dashed box in (5). Here we see in (a) that the input to the layer is a vector space transforming under the group representation $\rho_1 \oplus \rho_2$ corresponding to graphs and sets as inputs. These processed by automorphism equivariant update functions f, where there is an $f^{S_{H^i}}$ for each automorphism group. (b) Following the automorphism equivariant update function we re-insert the vector space features back into their respective nodes in the original graph, and (c) re-extract back into the respective subgraphs. This can be seen as a form of narrowing and promotion and allows information to propagate between subgraphs.

An overview of the architecture is detailed in Figure 6.1, where each component is described as follows. (1-2) The first component of our SPEN model comprises of splitting the graph G into subgraphs $H_1 \dots H_n$, for a graph with n nodes. For this we use a k-ego network policy extracting a subgraph for each node in the input graph. (3) Secondly, we place subgraphs $H_1 \dots H_n$ into bags $S_{H^i} \dots S_{H^j}$, where each bag holds subgraphs of a specific size, with $i \dots j$ being the different sizes $|H|$ of subgraphs. The extracted subgraphs are used as fully-connected graphs with zero features for non-edges; this results in each bag of subgraphs representing an automorphism group. Here it is worth noting that the figure shows three bags of subgraphs or three automorphism groups, while in general there does not have to just be three automorphism groups and this can vary. (4) We then process the bags of subgraphs with an automorphism equivariant linear map. This comprises of multiple separate functions f, with a different function processing each automorphism group, i.e. $f_0^{S_{H^3}}$ is the function mapping the automorphism group corresponding to the bag S_{H^3} of subgraphs in layer 0. This is a map from a 2-order permutation representation, i.e. graphs, to the direct sum of a 1-order and 2-order permutation representation. (5) We then stack multiple layers each comprising of an automorphism equivariant mapping function. Each of the automorphism groups is updated with a function mapping from the direct sum of a 1-order and 2-order permutation representation to the direct sum of a 1-order and 2-order permutation representation. (6) The final layer in the model is again an automorphism equivariant mapping function were each automorphism group is mapped from the direct sum of a 1-order and 2-order permutation representation to a 0-order representation. (7) Each automorphism group is averaged. (8) An MLP is used to update the feature space.

Automorphism Symmetry

We have defined the subgraph selection policy used, which results in the input graph, G, being represented as a bag of bags of subgraphs, $\{\{S_{H^i} \dots S_{H^k}\}\}$. As each bag of subgraphs holds subgraphs of a different size, each forms a different automorphism group. Therefore, we have different feature spaces for different subgraph sizes and different representations acting on these, i.e. $\rho(S_{H^i}) \neq \rho(S_{H^j})$. A linear layer acting on subgraphs should therefore operate differently on subgraphs from different automorphism groups. This is demonstrated in Figure 6.1 (4,5,6) by having different linear map f for each bag of subgraphs ($f_i^{S_{H^3}}$, $f_i^{S_{H^4}}$, $f_i^{S_{H^5}}$). This is defined more rigorously through the concept of naturality. Given a linear layer mapping from a feature space acted upon by ρ_m to a feature space acted upon by ρ'_m for each subgraph H a (linear) map can be detailed as $f_H : \rho_m(H) \to \rho'_m(H)$. However, given two isomorphic subgraphs H and H' are the same graph up-to some bijective mapping, we want f_H and $f_{H'}$ to process the feature spaces in an equivalent manner. This naturality constraint is therefore similar to the global naturality constraint on graphs G, except for subgraphs, H:

$$\rho'(\phi) \circ f_H = f_{H'} \circ \rho(\phi). \tag{6.3}$$

This constraint (Equation 6.3) says that if we first transition from the input feature space acted upon by $\rho(H)$ to the equivalent input feature space acted upon by $\rho'(H)$ via an isomorphism transformation $\rho(\phi)$ and then apply $f_{H'}$ we get the same thing as first applying f_H and then transitioning from the output feature space acted upon by $\rho'(H)$ to $\rho'(H')$ via the isomorphism transformation $\rho'(\phi)$. Since $\rho(\phi)$ is invertible, if we choose f_H for some H then we have determined $f_{H'}$ for any isomorphic H' by $f_{H'} = \rho'(\phi) \circ f_H \circ \rho(\phi)^{-1}$. Therefore, for any automorphism $\phi : H \to H$, we get an equivariance constraint $\rho'(\phi) \circ f_H = f_H \circ \rho(\phi)$. Thus, a layer in the model must have, for each automorphism group, a map, f_H, that is equivariant to automorphisms. Therefore, our choice of subgraph selection policy, extracting bags of subgraphs, aligns with the naturality constraint, in that we require a mapping function $f^{S_{H^i}}$, for each bag of subgraphs.

Permutation Symmetries Within Bags of Subgraphs

The order of subgraphs in each bag of subgraphs is arbitrary and changes if the input graph is permuted. This ordering comes from the need to represent the graph in a computer and the ordering is tracked through the use of concrete graphs. It would therefore be undesirable for the output prediction to be dependent upon this arbitrary ordering. This is overcome in the choice of insert and extract functions used to share information between subgraphs demonstrated in Figure 6.2. At the end of a linear layer in our model node and edge features from the original graph can be represented multiple times, i.e. they occurs in multiple subgraphs. We therefore average these features across subgraphs through insert and extract functions, and in doing this ensure the output is invariant to the ordering of subgraphs in each bag.

Subgraph Linear Maps

Each bag of subgraphs has a symmetry group that is given by permutation of the order of nodes in a graph. This group is denoted S_n for a graph of n nodes. The group S_n acts on on the graph via $(\sigma \cdot A)_{ij} = A_{\sigma^{-1}(i)\sigma^{-1}(j)}$. subgraphs, H, therefore have a symmetry group $S_m \leq S_n$ and we are interested in constructing graph neural network layers equivariant to this symmetry group. The graph is an order-2 tensor and the action of the permutation group can be generalised to differing order tensors. For example, the set of nodes in a graph is an order-1 tensor. For the case of a linear mapping from order-2 permutation representations to order-2 permutation representations, the basis space was shown to

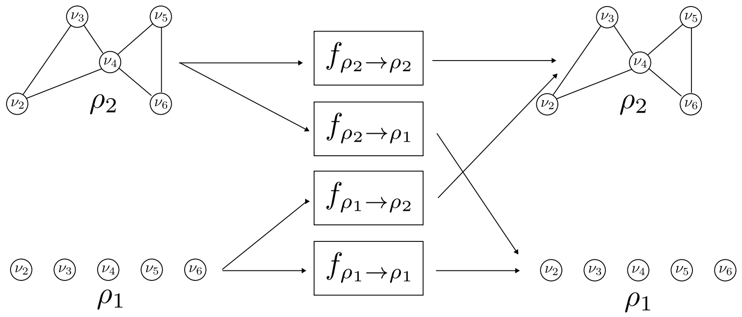


Figure 6.3: An example of what a function box in Figure 6.1 breaks down into. This example is for a function $f_i^{S_H}$ mapping from a representation $\rho_1 \oplus \rho_2 \to \rho_1 \oplus \rho_2$. This demonstrates there is a mapping function from ρ_2 to ρ_2, from ρ_2 to ρ_1, from ρ_1 to ρ_2, and from ρ_1 to ρ_1, as well as ρ_2 is a group representation for graphs and ρ_1 is a group representations for sets.

comprise of 15 elements by Maron et al. (2018). Similarly, the constraint imposed by equivariance to the group of permutations can be solved for different order representation spaces and we provide an example of all mappings between representation spaces from order 0-2 in Figure 6.4. Further, we are not restricted to selecting a single input-output order permutation representation space and can construct permutation equivariant linear maps between multiple representations separately through the direct sum $\oplus$. For example, the direct sum of order 1 and 2 representations is given by $\rho_1 \oplus \rho_2$. We utilise this to build the linear mapping functions, $f^{S_{H^i}}$, shown in Figures 6.1 (4,5,6) and 6.2 (a) to use linear maps between feature space acted on by different representations ρ. An example of how this map breaks down into individual functions mapping from and to a graph feature space acted on by 2-order permutation representations and a set feature space acted on by 1-order permutation representations is provided in Figure 6.3. Furthermore, the bases utilised within each of these functions, mapping between feature spaces acted upon by different order representations, is given in Figure 6.4.

Due to the construction of a subgraph the subgraphs inherit node ids from the original graph. Therefore, a permutation of the order of the nodes in the original graph corresponds to an equivalent permutation of the ordering of the nodes in the subgraphs. In addition, as the permutation action on the graph does not change the underlying connectivity, the subgraphs extracted are individually unchanged up-to some isomorphism. Therefore, a permutation of the graph only permutes the ordering at which the subgraphs are extracted.

Algorithm Summary

The proposed approach is to split the input graph into a set of subgraphs. Collect subgraphs that are the same size. The process these with a GNN model using different

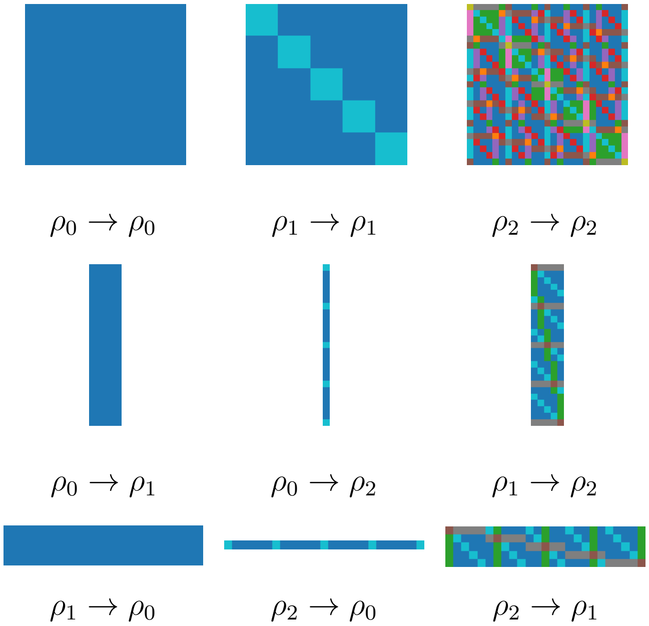


Figure 6.4: Bases for mappings to and from different order permutation representations, where ρ_k is a k-order representation. Each color in a basis indicates a different parameter. $\rho_2 \to \rho_2$ is a mapping from a graph to a graph, and has 15 learnable parameters. Further, there are mappings between different order representation spaces and higher order representation spaces.

permutation representation spaces, which uses the same weights for subgraphs of the same size and different weights for subgraphs not of the same size. Finally, the subgraphs are pooled, averaged, and the features updated with an MLP before predicting the graph class.

6.3.2 Analysis of Expressivity and Scalability

In this section we study both the expressive power of our architecture following the theoretical analysis outlined in Bevilacqua et al. (2022) by its ability to provably separate non-isomorphic graphs and the scalability by its ability to process larger graphs that its predecessor. The expressivity analysis follows from prior work of (Bevilacqua et al., 2022) and (Zhao et al., 2022).

6.3.2.1 WL Test and Expressive Power

The Weisfeiler-Lehman (WL) test (Weisfeiler & Leman, 1968) is a graph isomorphism test commonly used as a measure of expressivity in GNNs. This is due to the similarity between the iterative color refinement of the WL test and the message passing layers of a GNN. The WL test is a necessary but insufficient condition, which is not able to distinguish between all non-isomorphic graphs. The WL test was extended to the k-WL test, which provides increasingly more powerful tests that operate on k-tuples of nodes.

WL analogue for subgraphs. One component of our model is the idea of operating on subgraphs rather than the entire graph, more specifically our architecture operates on ego-network subgraphs. We follow the theoretical analysis outlined in Bevilacqua et al. (2022) to verify that operating on subgraphs will improve the expressive power of the base model within our automorphism equivariant model. Therefore, following Bevilacqua et al. (2022) we present a color-refinement variant of the WL isomorphism test that operates on a bag of subgraphs.

Definition 25. The subgraph-WL test utilises a color refinement of $c^{t+1}_{v,S} = \mathrm{HASH}(c^t_{v,S}, \mathcal{N}^t_{v,S}, C^t_v)$, which is a simplification of that outlined in (Bevilacqua et al., 2022), where $\mathrm{HASH}(\cdot)$ is an injective function, $\mathcal{N}^t_{v,S}$ is the node neighbourhood of v within the ego-network subgraph S, and C^t_v is the multiset of v's colors across subgraphs.

Theorem 6.3.1. subgraph-WL is strictly more powerful than 1&2-WL.

The proof of Theorem 6.3.1 begins with the definition of vertex coloring, required for the WL test.

Definition 26. (Vertex coloring). A vertex coloring is a function mapping a graph and one of its nodes to a "color" from a fixed color palette (Rattan & Seppelt, 2021; Bevilacqua et al., 2022).

Generally, a vertex coloring is a function $c : \mathcal{V} \to C,\ (G, v) \mapsto c^G_v$, where $\mathcal{V}$ is the set of all possible tuples of the form (G, v) with $G = (V, E)$ the set of all finite graphs and $v \in \mathcal{V}$ (Bevilacqua et al., 2022).

Next, we define a color refinement step. This is required to update the coloring of each vertex, which is the iterative process the WL test goes through to determine if two graphs are non-isomorphic.

Definition 27. Vertex color refinement. Let c, d be two vertex colorings. We say that d refines c when for all graphs $G = (V^G, E^G)$, $H = (V^H, E^H)$ and all vertices $v \in V^G$, $u \in V^H$ we have that $g^G_v = d^H_u \Rightarrow c^G_v = c^H_u$. This is written as $d \sqsubseteq c$ (Bevilacqua et al., 2022).

It is also given that when working with a specific graph pair G^1, G^2, the refinement d of c is written $d \sqsubseteq_{G^1,G^2} c$, when, in particular, it holds that $\forall v \in V^{G^1}, u \in V^{G^2}$, $d^{G_1}_v = d^{G_2}_u \Rightarrow c^{G_1}_v = c^{G_2}_u$ (Bevilacqua et al., 2022).

The 1-WL test represents a graph as a multiset (or histogram) of colors associated with its nodes. This coloring induces a partitioning of the nodes into color classes, where two nodes belong to the same partition iff they have the same coloring. The algorithm starts from some initial coloring and iteratively updates the coloring, leading to, at each step where the algorithm does not terminate, a finer-grained node partitioning. Each of these iterations is a refinement step, since, if c indicates the coloring computed at iteration t then the subsequent coloring at iteration $t+1$ is given by $c^{t+1} \sqsubseteq_{G,G} c^{t}$

Definition 28. subgraph-1-WL (Zhao et al., 2022). subgraph-1-WL generalises the 1-WL test by replacing the color refinement step $c_v^{t+1} = \mathrm{HASH}(Star^t(v))$ with $c_v^{t+1} = \mathrm{HASH(G^t[\mathcal{N}_k(v)]), \forall v \in \mathcal{V}}$. Where $G[\mathcal{N}_k(v)]$ is the k-hop egonet.

We start by proving that subgraph-WL is at least as expressive as 1-WL. For this we first characterise our subgraph-WL to make the comparison between a refinement strategy for a bag of subgraphs and those which operate on graphs.

Definition 29. Subgraph-WL node refinement. For a graph $G = (V, E)$ we denote S_G as a bag of subgraphs generated by taking the k-hop ego net of each node $v \in V$. The color refinement for node v at time step $t \geq 0$, C_v^t, is given by the set of node colors across the subgraphs, denoted as $\{\{c_{v,H}^t\}\}_{H \in S_G}$.

Definition 30. The definition of color refinement $b \sqsubseteq a$ (Bevilacqua et al., 2022) is that for all graphs $G^1 = (V^1, E^1)$ and $G^2 = (V^2, E^2)$ and all nodes $v \in V^1$, $w \in V^2$ that, $b_v = b_w \Rightarrow a_v = a_w$.

For such a node refinement policy, inclusive of node refinement across subgraphs, Bevilacqua et al. (2022) show that, for b the node coloring from a subgraph-WL refinement and a the node coloring from a WL refinement, $b \sqsubseteq a$. It then follows that for all graphs $G^1 = (V^1, E^1)$ and $G^2 = (V^2, E^2)$ and all nodes $v \in V^1$, $w \in V^2$ that $b_v = b_w \Rightarrow a_v = a_w$.

Lemma 6.3.2. subgraph-WL is at least as powerful as subgraph-1-WL in distinguishing non-isomorphic graphs.

We follow the notation of (Bevilacqua et al., 2022) and denote denote a colorings by the subgraph-1-WL algorithm, b colorings on each subgraph by the subgraph-WL algorithm, and c coloring on each node within a subgraph by the subgraph-WL algorithm. We also denote S^1, S^2 as the bags of subgraphs from G^1, G^2 respectively. If $|S^1| \neq |S^2|$ then the two graphs are trivially distinguished by subgraph-WL. In the case where $|S^1| = |S^2|$ we seek to show that if subgraph-1-WL (Zhao et al., 2022) identifies non-isomorphic graphs, then so does subgraph-WL.

First, subgraph-1-WL at time step t deems two graphs non-isomorphic if the following two are assigned two different multisets of node colors (Bevilacqua et al., 2022):

$$\{\{a_v^t | v \in \mathcal{V}^v\}\} \neq \{\{a_w^t | w \in \mathcal{V}^w\}\},$$

while subgraph-WL deems them non-isomorphic when the following two are assigned two different multisets of subgraph colors (Bevilacqua et al., 2022):

$$\{\{b^t_{S_k^1}\}\}_{k=1}^m \neq \{\{b^t_{S_h^2}\}\}_{h=1}^m.$$

If it is given that subgraph-1-WL distinguishes between two graphs at iteration T, then by Definition 30 $b^T \sqsubseteq a^T$. In addition, Bevilacqua et al. (2022) prove that for such a coloring at T a subgraph refinement policy such as subgraph-WL is refined by the coloring generated at $T+1$ on any pair of subgraphs: $\forall H_1, H_2 \in S^1 \cup S^2 \; c^{T+1} \sqsubseteq_{H_1,H_2} b^T$. The proof follows from the definition of the refinement step given by (Bevilacqua et al., 2022) that in an algorithm for a bag of subgraphs, namely, the inclusion of a term which refines over the multiset of node colors across subgraphs implies that if $C_v^T = C_u^T$ then $b_v^T = b_u^T$. This gives that if subgraph-1-WL can distinguish between two graphs at time step T then the subgraph refinement policy yields distinct colors to any pair of subgraphs. Therefore, subgraph-WL can distinguish between two graphs that subgraph-1-WL can and is at least as expressive.

This provides the necessary detail for the proof of Theorem 6.3.1. To prove that subgraph-WL is strictly more powerful than 1&2-WL we could instead prove that subgraph-1-WL is strictly more powerful than 1&2-WL and then by Lemma 6.3.2 the proof that subgraph-WL is strictly more powerful than 1&2-WL is complete. In fact, Zhao et al. (2022) prove that subgraph-1-WL is strictly more powerful than 1&2-WL by presenting a pair of non-ismorphic graphs that subgraph-1-WL distinguishes but 1-WL cannot. Therefore, we can conclude that our subgraph-WL is strictly more powerful than 1&2-WL.

Therefore we know that even for a simple 1-WL expressive function in the GNN, such as message passing, our model, splitting the graph into subgraphs, is immediately more expressive than 1&2-WL.

Comparing SPEN to the WL test. We have already shown that when considering a graph update function that operates on a bag of k-ego network subgraphs, even if the update function itself has limited expressivity, it is more expressive than 1&2-WL. SPEN utilises a natural permutation equivariant update function through operating on a bag of bags of subgraphs. The naturality constraint of our model states that each automorphism group of subgraphs should be processed by a different (linear) map. In addition, we utilise higher-dimensional GNNs. Both of these choices are expected to increase the expressive power of our model.

Proposition 6.3.3. For two non-isomorphic graphs G^1 and G^2 subgraph-WL can successfully distinguish them if (1) they can be distinguished as non-isomorphic from the multisets of subgraphs and (2) $\mathrm{HASH}(\cdot)$ is discriminative enough that $\mathrm{HASH}(c^t_{v,S^1}, \mathcal{N}^t_{v,S^1}, C^t_v) \neq \mathrm{HASH}(c^t_{v,S^2}, \mathcal{N}^t_{v,S^2}, C^t_v)$.

This implies that, despite the subgraph policy increasing the expressive power of the model, it is still limited by the ability of the equivalent to the $\mathrm{HASH}(\cdot)$ function's ability to discriminate between the bags of subgraphs. The naturality constraint of our model processing each automorphism group with a different higher-dimensional GNN is therefore expected to increase the expressive power of our model over subgraph methods utilising an MPNN.

Theorem 6.3.4. SPEN is strictly more powerful than subgraph MPNN.

We demonstrate the claim of Theorem 6.3.4 similarly to Bouritsas et al. (2022); de Haan et al. (2020). We use a neural network with random weights on a graph and compute a graph embedding. We say the neural network finds two graphs to be different if the graph embeddings differ by an ℓ_2 norm of more than $\epsilon = 10^{-3}$ of the mean ℓ_2 norms of the embeddings of the graphs in the set. The network is said to be most expressive if it only finds non-isomorphic graphs to be different. We test this by considering a set of 100 random non-isomorphic non-regular graphs, a set of 100 non-isomorphic regular graphs, a set of 15 non-isomorphic strongly regular graphs,[2] and a set of 100 isomorphic graphs. Table 6.1 shows that a simple invariant message passing (GCN) (Welling & Kipf, 2016) as well as a simple invariant message passing model operating on subgraphs (SGCN), which we created, are unable to distinguish between regular and strongly regular graphs. Further, it is shown that PPGN (Maron et al., 2019) can distinguish regular graphs but not strongly regular graphs, although a variant of PPGN that uses high order tensors should also be able to distinguish strongly regular graphs. On the other hand, our SPEN model is able to distinguish strongly regular graphs. Therefore, our model is able to distinguish non-isomorphic graphs that a subgraph MPNN cannot and is strictly more powerful.

6.3.2.2 Scalability

Global permutation equivariant models of the form found by Maron et al. (2018) operate over the entire graph. They therefore scale with $\mathcal{O}(n^2)$, for graphs with n nodes. Our method operates on k-ego network subgraphs where a subgraph is produced for each

[2] See http://users.cecs.anu.edu.au/~bdm/data/graphs.html.

Table 6.1: Rate of pairs of graphs in the set of graphs found to be dissimilar in expressiveness experiment. An ideal method only find isomorphic graphs dissimilar. A score of 1 implies the model can find all graphs dissimilar, while 0 implies the model finds no graphs dissimilar.

Model	Random	Regular	Str. Regular	Isom.
GCN	1	0	0	0
SGCN	1	0	0	0
PPGN	1	0.97	0	0
SPEN	1	0.98	0.97	0

node in the original graph. Our method therefore scales with $\mathcal{O}(nm^2)$, where m is the number of nodes in the k-ego network subgraph. It is therefore clear that if $n = m$, theoretically, our method scales more poorly than global permutation equivariant models, although this would imply the graph is fully-connected and every subgraph is identical. In this situation extracting subgraphs is irrelevant and only 1 subgraph is required (the entire graph). Hence if $n = m$ our method scales with that of global permutation equivariant models. The more interesting situation, which forms the majority of graphs, is when $n \neq m$. When $m \ll n$ our method scales more closely with methods that scale linearly with the size of the graph and it is for this type of data that our method offers a significant improvement in scalability over global permutation equivariant models.

We empirically show how SPEN and global permutation equivariant methods scale depending on the size of n and m by analysing the GPU memory usage of both models across a range of random regular graphs. We utilise random regular graphs for the scalability test as it allows for precise control over the size of the overall graph and subgraphs. We compare the GPU memory usage of both models across a range of graph sizes with a subgraph size of $m = 3$, 6, and 9. Through analysing the graphs in the TUDataset, which we make use of when experimenting on graph benchmarks, we note that the average subgraph sizes range between 3 and 10 (see Table 6.2), justifying the choice of subgraphs in the scalability tests. Figure 6.5 shows that the Global Permutation Equivariant Network (GPEN) (Maron et al., 2018) cannot scale beyond graphs with 500 nodes. On the other hand, our method (SPEN) scales to larger graphs of over an order of magnitude larger. In the situation where $m = 3$ GPEN can process graphs of size up to 500 nodes, while our SPEN can process graphs of size up to 10,000 nodes using less GPU memory.

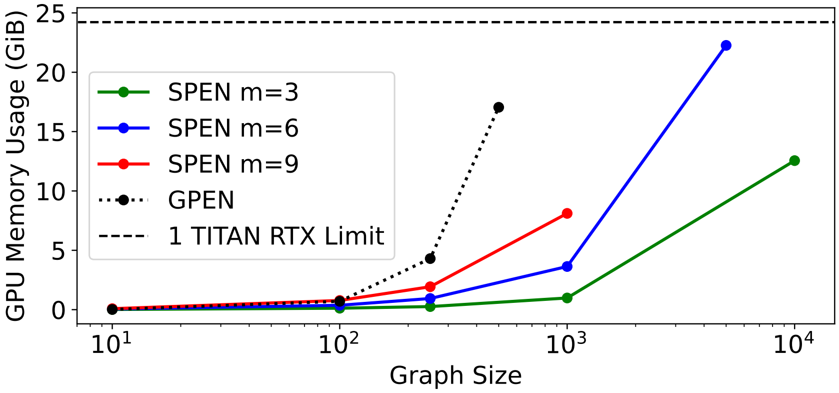


Figure 6.5: Computational cost of global permutation equivariant model (GPEN) and our (SPEN) model with a very similar number of model parameters for varying average size graphs. For this test we constructed random regular graphs of varying size using the NetworkX package (Hagberg et al., 2008). For SPEN subgraphs were constructed using a 1-hop ego network policy. As is demonstrated by the log-axis, SPEN can process graphs an order of magnitude larger than global methods.

6.3.3 Experiments

6.3.3.1 Dataset and Implementation Details

TU Datasets

We tested our method on a series of 7 different real-world graph classification problems from the TU Datasets benchmark of (Yanardag & Vishwanathan, 2015). Five of these datasets originate from bioinformatics, while the other two come from social networks. To highlight some features of each dataset, we note that both MUTAG and PTC are very small datasets, with MUTAG only having 18 graphs in the test set when using a 10% testing split. Further, the Proteins dataset has the largest graphs with an average number of nodes in each graph of 39. Also, NCI1 and NCI109 are the largest datasets having over 4000 graphs each, as a result it is expected that this dataset will lead to less spurious results. Finally, IMDB-B and IMDB-M generally have smaller graphs, with IMDB-M only having an average number of 13 nodes in each graph. The small size of graphs coupled with having 3 classes appears to make IMBD-M a challenging problem. We present the range of graph sizes and subgraph sizes when utilising a 1-ego network subgraph extraction policy in Table 6.2.

Model Architecture

We consider the input graphs as an input feature space that is an order 2 representation. For each local permutation equivariant linear layer we use order 1 and 2 representations as the feature spaces. This allows for projection down from graph to node feature spaces

Table 6.2: Different range of graph sizes and subgraph sizes for each dataset considered from TU datasets.

Dataset	MUTAG	PTC	PROTEINS	NCI1	NCI109	IMDB-B	IMDB-M
Graph Sizes	10-28	2-109	4-620	3-111	4-111	12-136	7-89
subgraph Sizes	2-5	2-5	1-26	2-5	1-6	1-135	1-88
Mean Subgraph Size	3.2	3.0	4.7	3.2	3.2	9.8	10.1

through the basis for $\rho_2 \to \rho_1$, projection up from node to graph feature spaces through the basis for $\rho_1 \to \rho_2$, and mappings across the same order representations through $\rho_2 \to \rho_2$ and $\rho_1 \to \rho_1$. The final local permutation equivariant linear layer maps to order 0 representations through $\rho_2 \to \rho_0$ and $\rho_1 \to \rho_0$ for the task of graph level classification. In addition to the graph layers, we also add 3 MLP layers to the end of the model.

Despite these specific choices, which were made to provide a baseline of our method for comparison to existing methods, the framework we present is much more general and different representation spaces can be chosen. Therefore, different permutation representation spaces, $\rho_1 \oplus \rho 2 \oplus \cdots \oplus \rho_i$, can be chosen for different layers in the model and a different k value can be chosen when creating the subgraphs.

For all experiments we used a 1-hop ego networks as this provides the most scalable version of our method. We trained each model for 50 epochs on all datasets using the Adam optimizer. We considered the evaluation procedure as was conducted in (Bevilacqua et al., 2022; Xu et al., 2019; Yanardag & Vishwanathan, 2015; Niepert et al., 2016). Specifically, we conducted 10-fold cross validation and reported the mean and standard deviation of validation accuracies across the 10 folds. Specific model architecture details are provided in Table 6.3 where they vary across each dataset.

The model is constrained to be equivariant to the automorphism groups of the bags of subgraphs. For MUTAG, PTC, NCI1, and NCI109 we directly constrain the model to the automorphism groups of the bags of subgraphs. For PROTEINS, IMDB-B, and IMDB-M, there exist some bags of subgraphs which comprise of a single subgraph. As this would lead to no weight sharing between these subgraphs and and any other subgraphs we parameterize the automorphism constraint to bunch bags of subgraphs which contain few subgraphs.

Table 6.3: Specific model architecture details for each dataset considered from TU-Datasets.

Dataset	MUTAG	PTC	PROTEINS	NCI1	NCI109	IMDB-B	IMDB-M
# Layers	4	4	6	6	6	6	6
Feature Dimension	16	16	32	64	64	32	32
$\ell 2$ regularisation	0	0	0.02	0	0	0	0

6.3.3.2 Graph Benchmarks

TU Datasets

We perform experiments with our method to compare to leading methods. For this we are looking to see how global permutation equivariant methods compare to our method? How our approach compares in terms of validation accuracy on real graph benchmarks with state-of-the-art methods? How our method scales wen compared with global permutation methods on real benchmark tasks?

We compare to a wide range of alternative methods, including subgraph based methods, higher-dimensional GNN methods, and automorphism equivariant methods. We focus specifically on IGN (Maron et al., 2018) as this method uses an order-2 permutation equivariant tensor representation space for the linear map and is therefore the most similar to our base GNN model. Bevilacqua et al. (2022) test their DSS-GNN on multiple different subgraph policies and here we compare to the method utilising k-hop ego networks as this is the most similar variant to our method.

Table 6.5 compares our SPEN model to a range of other methods on benchmark graph classification tasks from TUDatasets (Morris et al., 2020a). We perform statistical significance analysis using Welch's ANOVA method for comparing multiple means with different variances. We consider the null hypothesis that the means are equal and to reject this null hypothesis the p value is required to be below 0.05. We highlight in grey all the methods indistinguishable from the state-of-the-art method. See Appendix D.1 for more details on the statistical significance analysis. Comparing out method (SPEN) to a higher-dimensional global permutation equivariant (IGN) demonstrates that our method significantly outperforms the base GNN model on four datasets and is statistically indistinguishable on the remaining three. Table 6.5 also highlights that our method is statistically indistinguishable from the state-of-the-art result on six out of seven datasets, and achieves a larger mean on three of these datasets. This demonstrates that our method performs competitively across the range of datasets. The strong results produced by our method suggests that our framework's improved expressivity is beneficial for learning on graph classification tasks. Further, our statistical significance analysis highlights that previous methods which claim state-of-the-art are not achieving this in a statistically meaningful way.

This approach is most suitable for the types of graphs present in the TUDatasets presented in this section. Namely a dataset of smaller graphs that may have some non-isomorphic graphs that would not be distinguished by a message passing neural network. Despite this approach improving scalability over the base GNN, it is unlikely to scale to very large graphs unless they are particularly sparse due to the dense representation of graphs. Furthermore, if an N node graph is fully connected this approach does not make sense as splitting into subgraphs would yield N copies of the same subgraph, which is the original graph. This also applies for graphs close to fully connected, where many of the subgraphs would simply be repeats of each other.

Scalability on Graph Benchmark Datasets

Figure 6.6 demonstrates that the improved scalability of our methods on regular graphs carries over onto graphs on real-world benchmarks. This show that as the size of the graphs grows global permutation equivariant methods rapidly consume large amounts of GPU memroy. On the other hand, due to using subgraphs, our method scales to larger graphs while using significantly less memory. This highlights that our method offers a significant improvement in scalability over global permutation equivariant models.

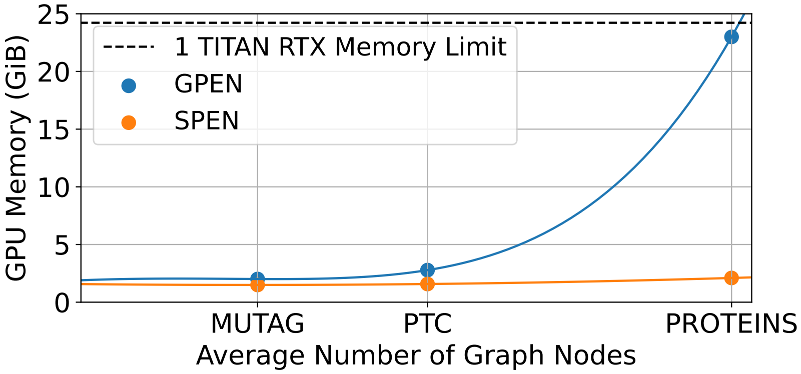


Figure 6.6: Computational cost of a global permutation equivariant model (GPEN) and our method (SPEN) with a very similar number of model parameters and batch size for datasets with varying average size graphs from the TUDatasets. For SPEN subgraphs were constructed using a 1-hop ego network policy.

Subgraph Compute Run-time

The current implementation of the model computes the subgraphs on the fly, although this could be moved into a pre-processing stage which would speed up run-time of the model and slow down the pre-processing stage. Here, in Table 6.4, we provide the run-time of the computation of the subgraphs for each dataset to provide an idea of how long this process takes in our 1-hop SPEN model.

Table 6.4: Run-time to compute subgraphs for each dataset from TUDatasets when using a 1-hop ego-net subgraph selection policy.

Dataset	subgraph Compute Run-time [s]
MUTAG	0.90
PTC	2.26
PROTEINS	14.25
NCI1	31.21
NCI109	30.96
IMDB-B	17.79
IMDB-M	19.75

6.3.4 Discussion

The results show that building models that operate on k-ego network subgraphs improves the scalability of a GNN over the base GNN function. Although, the graphs considered are still small in comparison to some graphs in the wild, such as social network graphs. For very large graphs with widely varying node degrees the automorphsim constraint would most liekly need to be parameterised, which may impact the model performance. Despite this, utilising k-ego network subgraphs, higher order permutation representation feature spaces, and automorphism symmetries allows for the development of GNN models that perform strongly on real-world graph benchmarks.

Table 6.5: Comparison between our SPEN model and other deep learning methods. Larger mean results are better with the standard deviation around the mean given in (). Methods in comparison are: GDCNN (Zhang et al., 2018), PSCN (Niepert et al., 2016), DCNN (Atwood & Towsley, 2016), ECC (Simonovsky & Komodakis, 2017), DGK (Yanardag & Vishwanathan, 2015), DiffPool (Ying et al., 2018), CCN (Kondor et al., 2018), IGN (Maron et al., 2018), GIN (Xu et al., 2019), 1-2-3 GNN (Morris et al., 2019), PPGN (Maron et al., 2019), LNGN (GCN) (de Haan et al., 2020), GSN (Bouritsas et al., 2022), SIN (Bodnar et al., 2021b), CIN (Bodnar et al., 2021a), and DSS-GNN (GC) (EGO) (Bevilacqua et al., 2022). All scores statistically indistinguishable (via Welch's ANOVA) from the highest mean in each benchmark have a gray background, whilst the highest mean values are in bold.

Dataset	MUTAG	PTC	PROTEINS	NCI1	NCI109	IMDB-B	IMDB-M
size	188	344	1113	4110	4127	1000	1500
classes	2	2	2	2	2	2	3
avg node #	17.9	25.5	39.1	29.8	29.6	19.7	13
			Results				
GDCNN	85.8 (1.7)	58.6 (2.5)	75.5 (0.9)	74.4 (0.5)	NA	70.0 (0.9)	47.8 (0.9)
PSCN	89.0 (4.4)	62.3 (5.7)	75 (2.5)	76.3 (1.7)	NA	71 (2.3)	45.2 (2.8)
DCNN	NA	NA	61.3 (1.6)	56.6 (1.0)	NA	49.1 (1.4)	33.5 (1.4)
DGK	87.4 (2.7)	60.1 (2.6)	75.7 (0.5)	80.3 (0.5)	80.3 (0.3)	67.0 (0.6)	44.5 (0.5)
CCN	91.6 (7.2)	70.6 (7.0)	NA	76.3 (4.1)	75.5 (3.4)	NA	NA
IGN	83.9 (13.0)	58.5 (6.9)	76.6 (5.5)	74.3 (2.7)	72.8 (1.5)	72.0 (5.5)	48.7 (3.4)
GIN	89.4 (5.6)	64.6 (7.0)	76.2 (2.8)	82.7 (1.7)	NA	75.1 (5.1)	52.3 (2.8)
PPGN v1	90.5 (8.7)	66.2 (6.5)	**77.2** (**4.7**)	83.2 (1.1)	81.8 (1.9)	72.6 (4.9)	50 (3.2)
PPGN v2	88.9 (7.4)	64.7 (7.5)	76.4 (5.0)	81.2 (2.1)	81.8 (1.3)	72.2 (4.3)	44.7 (7.9)
PPGN v3	89.4 (8.1)	62.9 (7.0)	76.7 (5.6)	81.0 (1.9)	82.2 (1.4)	73.0 (5.8)	50.5 (3.6)
LNGN (GCN)	89.4 (1.6)	66.8 (1.8)	71.7 (1.0)	82.7 (1.4)	83.0 (1.9)	74.8 (2.0)	51.3 (1.5)
GSN-e	90.6 (7.5)	68.2 (7.2)	76.6 (5.0)	83.5 (2.3)	NA	**77.8** (**3.3**)	**54.3** (**3.3**)
GSN-v	92.2 (7.5)	67.4 (5.7)	74.6 (5.0)	83.5 (2.0)	NA	76.8 (2.0)	52.6 (3.6)
SIN	NA	NA	76.5 (3.4)	82.8 (2.2)	NA	75.6 (3.2)	52.5 (3.0)
CIN	92.7 (6.1)	68.2 (5.6)	77.0 (4.3)	83.6 (1.4)	**84.0** (**1.6**)	75.6 (3.7)	52.7 (3.1)
DSS (EGO)	91.5 (4.9)	68.0 (6.1)	76.6 (4.6)	83.5 (1.1)	82.5 (1.6)	76.3 (3.6)	53.1 (2.8)
SPEN	**93.3** (**6.5**)	**71.3** (**9.7**)	74.8 (3.2)	**83.7** (**1.5**)	83.4 (1.2)	75.2 (3.1)	48.7 (2.0)

6.3.5 Future Directions

Many expressive GNN approaches use the WL test as a measure of expressivity. Despite this, tools exist such as Nauty (McKay & Piperno, 2014) exist that can compute the automorphism groups of graphs. Nauty cannot be used as a replacement for the neural networks developed as these need to be able to learn to classify or segment different graphs, something Nauty cannot do. Despite this, considering the approach Nauty takes to determine automorphism groups could lead to the development of more expressive GNNs.

6.4 Conclusion

We present a novel graph neural network framework for building models that operate on k-ego network subgraphs, which respect both the permutation symmetries of individual subgraphs and is equivariant to the automorphism groups across bags of subgraphs. The choice of subgraph policy leads to a novel choice of automorphism groups for the bags of subgraphs. The framework is more scalable than global higher-dimensional GNNs through the use of subgraphs and we have both theoretically and experimentally demonstrated this. We have shown that SPEN is provably more expressive than the base higher-dimensional permutation equivariant GNN and subgraph MPNNs through the choice subgraph selection policy, permutation equivariant base GNN, and automorphism equivariant kernel constraint. We have demonstrated the expressivity of the framework. Finally, we have shown that SPEN performs competitively across multiple graph classification benchmarks, achieving statistically indistinguishable accuracy compared to the state-of-the-art method on six out of seven datasets. We believe that our framework is a step forward in the development of graph neural networks, demonstrating improved expressivity, scalability, and experimentally achieving strong performances on benchmark datasets.

7

Conclusion

We have developed novel deep learning models that exploit symmetries in high-dimensional data across a range of scientific disciplines. We explored applications that use different data domains — including grids, continuous spaces and subgraphs — and developed neural networks that are equivariant to the corresponding group symmetry. Through this, we developed the theory of equivariant neural networks for new symmetry groups, providing connections between the theoretical understanding from the domain of interest and the group theoretic understanding of equivariant networks. This overcomes the challenges of creating neural networks comprising of novel layers outwith common machine learning frameworks, and provides example frameworks for building such models. We show that equivariant models — using a suitable inductive bias — given by considering the symmetries of the task — improves the models classification, segmentation, or generative ability. Further, we demonstrate that equivariant models are able to better generalise to out-of-distribution data, scale better to high-dimensional data, and are more robust than their non-equivariant counterparts. In addition, we demonstrate that models with an inductive bias taken from the symmetries of the data are more interpretable than unconstrained models, and better align with the intuition of domain experts.

In chapter 3 we present a novel model for the classification of age groups and species of mosquitoes. We present a model and training regime that overcomes the challenge of working with very small unbalanced data through the use a convolutional neural network and transfer learning. This enables the model to generalise to small genetically similar datasets, which was previously unachievable. We also show that this model classifies mosquito age groups and species with higher accuracy than models with no inductive bias. We further show that the trained model is sensitive to regions within the input spectra that correspond to regions that have chemical relevance, suggesting that the model is learning useful features. We compared our model to an equivalent model which does not utilise the inductive bias provided by convolutions. We evaluate the practical relevance of the model by predicting the age structure of wild mosquitoes, demonstrating that model can predict a similar distribution to what is seen in the wild with only a small quantity of data. Finally, the proposed model can be used to assist malaria mosquito surveillance and will hopefully assist in assessing the effectiveness of malaria control interventions.

In chapter 4 we develop two rotation equivariant models for images. The first model demonstrates the advantages of using a rotation equivariant model when there is rotational symmetries within the data for the task of segmentation of deforestation regions. For the second model we develop the theory of cylindrical symmetries for equivariant neural networks, presenting a novel model for the task of inverting optical fibre transmission effects. The development of this model bridges the gap between the theoretical understanding of multi-mode optical fibres in the physics community and group theoretic understanding of equivariant neural networks. We develop new datasets based on theoretical transmission matrices, which allows us to test the model on previously unachievable resolutions of images. We demonstrate that our equivariant approach to solving the task provides a more interpretable model, with desirable characteristics for applications of multi-mode fibre imaging. Our equivariant model better generalises to out-of-distribution data, is robust to noise in the speckled images, can learn and generalise with a smaller training dataset, and is robust to under parameterising the bases functions. We also show that the model can invert the transmission effects of a physical fibre in the laboratory. We demonstrate that our model can operate on higher resolution images than was previously possible, which opens up multi-mode optical fibre imaging to new applications where more detail is required in the predicted images. The dramatic reduction in the number of parameters within our model for each fibre configuration opens the way for future models that can learn mappings for high-resolution images, from a wider set of perturbed fibre poses and combine these using architectures such as VAEs.

In chapter 5 we develop a novel model for generating 3D hand meshes from 2D image, which ensures the entire model is rotation equivariant guaranteeing a rotation of the input hand corresponds to a rotated mesh being generated. We provide a comparison between and MLP and GNN based decoder, showing that an MLP decoder generates more accurate, smoother meshes than a GNN model. We demonstrate that rotation equivariance is a suitable inductive bias to build into each component of the model by demonstrating the stability of reconstruction under rotation on a small real-world dataset and demonstrating that our model outperforms, in terms of reconstruction accuracy, other leading methods. We also demonstrate competitive mesh reconstruction ability with state-of-the-art methods on two widely used, large scale, benchmark datasets. The use of rotation equivariance as an inductive bias in models for generating 3D hand meshes from 2D RGB images is supported by the robustness of the model when generating meshes at differing orientations.

In chapter 6 we develop a novel graph neural network framework for building models that operate on k-ego network subgraphs, which respect both the permutation symmetries of individual subgraphs and is equivariant to the automorphism groups across bags of subgraphs. The choice of subgraph policy leads to a novel choice of automorphism groups for the bags of subgraphs. The framework scales to larger graphs better than global higher-dimensional GNNs as a result of operating on subgraphs. The ability to scale to larger graphs is both theoretically and experimentally demonstrated. We have shown that SPEN is more expressive than the base higher-dimensional permutation equivariant GNN and subgraph MPNNs through the choice subgraph selection policy, permutation equivariant base GNN, and automorphism equivariant kernel constraint. We have also shown that SPEN performs competitively across multiple graph classification benchmarks, achieving statistically indistinguishable accuracy compared to the state-of-the-art method on six out of seven datasets.

Bibliography

Marjan Albooyeh, Daniele Bertolini, and Siamak Ravanbakhsh. Incidence networks for geometric deep learning. *arXiv preprint arXiv:1905.11460*, 2019.

Almut Arneth, Fatima Denton, Fahmuddin Agus, Aziz Elbehri, Karl Heinz Erb, B. Osman Elasha, Mohammad Rahimi, Mark Rounsevell, Adrian Spence, Riccardo Valentini, et al. Framing and Context. In *Climate change and land: An IPCC special report on climate change, desertification, land degradation, sustainable land management, food security, and greenhouse gas fluxes in terrestrial ecosystems*, pp. 1–98. Intergovernmental Panel on Climate Change (IPCC), 2019.

James Atwood and Don Towsley. Diffusion-convolutional neural networks. In *Advances in neural information processing systems*, pp. 1993–2001, 2016.

Kemen G. Austin, Amanda Schwantes, Yaofeng Gu, and Prasad S. Kasibhatla. What Causes Deforestation in Indonesia? *Environmental Research Letters*, 14(2), 2019.

Waiss Azizian and Marc Lelarge. Expressive power of invariant and equivariant graph neural networks. In *International Conference on Learning Representations*, 2021.

Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Pushing the envelope for RGB-based dense 3D hand pose estimation via neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1067–1076, 2019.

Chris Bass, Martin S Williamson, Craig S Wilding, Martin J Donnelly, and Linda M Field. Identification of the main malaria vectors in the anopheles gambiae species complex using a taqman real-time pcr assay. *Malaria journal*, 6(1):1–9, 2007.

John C Beier. Malaria parasite development in mosquitoes. *Annual review of entomology*, 43(1):519–543, 1998.

WN Beklemishev, TS Detinova, and VP Polovodova. Determination of physiological age in anophelines and of age distribution in anopheline populations in the ussr. *Bulletin of the World Health Organization*, 21(2):223, 1959.

Beatrice Bevilacqua, Fabrizio Frasca, Derek Lim, Balasubramaniam Srinivasan, Chen Cai, Gopinath Balamurugan, Michael M. Bronstein, and Haggai Maron. Equivariant subgraph aggregation networks. In *International Conference on Learning Representations*, 2022.

Samir Bhatt, DJ Weiss, E Cameron, D Bisanzio, B Mappin, U Dalrymple, KE Battle, CL Moyes, A Henry, PA Eckhoff, et al. The effect of malaria control on plasmodium falciparum in africa between 2000 and 2015. *Nature*, 526(7572):207–211, 2015.

Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yuguang Wang, Pietro Lio, Guido F Montufar, and Michael Bronstein. Weisfeiler and lehman go cellular: Cw networks. *Advances in Neural Information Processing Systems*, 34:2625–2640, 2021a.

Cristian Bodnar, Fabrizio Frasca, Yuguang Wang, Nina Otter, Guido F Montufar, Pietro Lio, and Michael Bronstein. Weisfeiler and lehman go topological: Message passing simplicial networks. In *International Conference on Machine Learning*, pp. 1026–1037. PMLR, 2021b.

Navid Borhani, Eirini Kakkava, Christophe Moser, and Demetri Psaltis. Learning to see through multimode fibers. *Optica*, 5(8):960–966, 2018.

Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10843–10852, 2019.

Giorgos Bouritsas, Fabrizio Frasca, Stefanos P Zafeiriou, and Michael Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3D hand pose estimation from monocular RGB images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 666–682, 2018.

Peter J Cameron. *Introduction to algebra*. OUP Oxford, 2007.

Beniamino Caputo, Francesca R Dani, Gill L Horne, Vincenzo Petrarca, Stefano Turillazzi, Mario Coluzzi, Angela A Priestman, and Alessandra della Torre. Identification and composition of cuticular hydrocarbons of the major afrotropical malaria vector anopheles gambiae ss (diptera: Culicidae): analysis of sexual dimorphism and age-related changes. *Journal of Mass Spectrometry*, 40(12):1595–1604, 2005.

Piergiorgio Caramazza, Oisín Moran, Roderick Murray-Smith, and Daniele Faccio. Transmission of natural scene images through a multimode fibre. *Nature communications*, 10 (1):1–6, 2019.

Kimberly M Carlson, Lisa M Curran, Dessy Ratnasari, Alice M Pittman, Britaldo S Soares-Filho, Gregory P Asner, Simon N Trigg, David A Gaveau, Deborah Lawrence, and Hermann O Rodrigues. Committed carbon emissions, deforestation, and community land conversion from oil palm plantation expansion in west kalimantan, indonesia. *Proceedings of the National Academy of Sciences*, 109(19):7559–7564, 2012.

Kimberly M. Carlson, Lisa M. Curran, Gregory P. Asner, Alice McDonald Pittman, Simon N. Trigg, and J. Marion Adeney. Carbon Emissions from Forest Conversion by Kalimantan Oil Palm Plantations. *Nature Climate Change*, 3(3):283–287, 2013.

Joel Carpenter, Benjamin J Eggleton, and Jochen Schröder. Maximally efficient imaging through multimode fiber. In *CLEO: Science and Innovations*, pp. STh1H–3. Optical Society of America, 2014.

Gabriele Cesa. *E (2)-Equivariant Steerable CNNs*. PhD thesis, Master's thesis, Universiteit van Amsterdam, 2020.

Theocharis Chatzis, Andreas Stergioulas, Dimitrios Konstantinidis, Kosmas Dimitropoulos, and Petros Daras. A comprehensive study on deep learning-based 3d hand pose estimation methods. *Applied Sciences*, 10(19):6850, 2020.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *International Conference on Machine Learning*, pp. 1725–1735. PMLR, 2020.

Xingyu Chen, Yufeng Liu, Chongyang Ma, Jianlong Chang, Huayan Wang, Tian Chen, Xiaoyan Guo, Pengfei Wan, and Wen Zheng. Camera-space hand mesh recovery via semantic aggregation and adaptive 2d-1d registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13274–13283, 2021.

Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *European Conference on Computer Vision*, pp. 769–787. Springer, 2020.

Youngwoon Choi, Changhyeong Yoon, Moonseok Kim, Taeseok Daniel Yang, Christopher Fang-Yen, Ramachandra R Dasari, Kyoung Jin Lee, and Wonshik Choi. Scanner-free and wide-field endoscopic imaging by using a single multimode optical fiber. *Physical review letters*, 109(20):203901, 2012.

Thomas S Churcher, Natalie Lissenden, Jamie T Griffin, Eve Worrall, and Hilary Ranson. The impact of pyrethroid resistance on the efficacy and effectiveness of bednets for malaria control in africa. *Elife*, 5:e16090, 2016.

Tomáš Čižmár and Kishan Dholakia. Shaping the light transmission through a multimode optical fibre: complex transformation analysis and applications in biophotonics. *Optics express*, 19(20):18871–18884, 2011.

Tomáš Čižmár and Kishan Dholakia. Exploiting multimode waveguides for pure fibre-based imaging. *Nature communications*, 3(1):1–9, 2012.

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

Taco Cohen et al. *Equivariant convolutional networks*. PhD thesis, Doctoral dissertation, Universiteit van Amsterdam, 2021.

Taco S Cohen and Max Welling. Steerable CNNs. *International Conference on Learning Representations (ICLR)*, 2017.

Taco S. Cohen, Mario Geiger, and Maurice Weiler. Intertwiners between induced representations (with applications to the theory of equivariant neural networks). *arXiv preprint arXiv:1803.10743*, 2018.

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

Anna Cohuet, Caroline Harris, Vincent Robert, and Didier Fontenille. Evolutionary forces on anopheles: what makes a malaria vector? *Trends in parasitology*, 26(3):130–136, 2010.

Peter E Cook, Leon E Hugo, Inaki Iturbe-Ormaetxe, Craig R Williams, Stephen F Chenoweth, Scott A Ritchie, Peter A Ryan, Brian H Kay, Mark W Blows, and Scott L O'Neill. Predicting the age of mosquitoes using transcriptional profiles. *Nature Protocols*, 2(11):2796–2806, 2007.

Leonardo Cotta, Christopher Morris, and Bruno Ribeiro. Reconstruction for powerful graph representations. *Advances in Neural Information Processing Systems*, 34, 2021.

Philip G. Curtis, Christy M. Slay, Nancy L. Harris, Alexandra Tyukavina, and Matthew C. Hansen. Classifying Drivers of Global Forest Loss. *Science*, 361(6407):1108–1111, 2018.

Pim de Haan, Taco S Cohen, and Max Welling. Natural graph networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 3636–3646. Curran Associates, Inc., 2020.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29:3844–3852, 2016.

Hui Deng, Tong Zhang, Yuchao Dai, Jiawei Shi, Yiran Zhong, and Hongdong Li. Deep non-rigid structure-from-motion: A sequence-to-sequence translation perspective. *arXiv preprint arXiv:2204.04730*, 2022.

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

Adrià Descals, Zoltan Szantoi, Erik Meijaard, Harsono Sutikno, Guruh Rindanata, and Serge Wich. Oil Palm (Elaeis Guineensis) Mapping with Details: Smallholder Versus Industrial Plantations and Their Extent in Riau, Sumatra. *Remote Sensing*, 11(21):2590, 2019.

Tatiana Sergeevna Detinova, Douglas S Bertram, World Health Organization, et al. *Age-grouping methods in Diptera of medical importance, with special reference to some vectors of malaria*. World Health Organization, 1962.

David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems*, 28:2224–2232, 2015.

Pengfei Fan, Tianrui Zhao, and Lei Su. Deep learning the high variability and randomness inside multimode fibers. *Optics express*, 27(15):20241–20258, 2019.

Yutong Feng, Yifan Feng, Haoxuan You, Xibin Zhao, and Yue Gao. MeshNet: mesh neural network for 3D shape representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 8279–8286, 2019.

Heather M Ferguson, Anna Dornhaus, Arlyne Beeche, Christian Borgemeister, Michael Gottlieb, Mir S Mulla, John E Gimnig, Durland Fish, and Gerry F Killeen. Ecology: a prerequisite for malaria elimination and eradication. *PLoS medicine*, 7(8):e1000303, 2010.

Marc Finzi, Max Welling, and Andrew Gordon Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In *International Conference on Machine Learning*, pp. 3318–3328. PMLR, 2021.

Jonathan A. Foley, Ruth DeFries, Gregory P. Asner, Carol Barford, Gordon Bonan, Stephen R. Carpenter, F. Stuart Chapin, Michael T. Coe, Gretchen C. Daily, Holly K. Gibbs, et al. Global Consequences of Land Use. *science*, 309(5734):570–574, 2005.

Daniel Franzen and Michael Wand. General nonlinearities in SO(2)-Equivariant CNNs. *Advances in Neural Information Processing Systems*, 34, 2021.

Fabian B. Fuchs, Daniel E. Worrall, Volker Fischer, and Max Welling. SE(3)-Transformers: 3D Roto-Translation Equivariant Attention Networks. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2020.

William Fulton and Joe Harris. *Representation theory: a first course*, volume 129. Springer Science & Business Media, 2013.

Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3D hand shape and pose estimation from a single RGB image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10833–10842, 2019.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017a.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural Message Passing for Quantum Chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1263–1272, 2017b.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pp. 729–734. IEEE, 2005.

Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. MeshCNN: a network with an edge. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019.

Rana Hanocka, Gal Metzer, Raja Giryes, and Daniel Cohen-Or. Point2mesh: A self-prior for deformable meshes. *ACM Trans. Graph.*, 39(4), 2020. ISSN 0730-0301. doi: 10.1145/3386569.3392415.

Jason Hartford, Devon Graham, Kevin Leyton-Brown, and Siamak Ravanbakhsh. Deep models of interactions across sets. In *International Conference on Machine Learning*, pp. 1909–1918. PMLR, 2018.

Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11807–11816, 2019.

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

Amir Hertz, Rana Hanocka, Raja Giryes, and Daniel Cohen-Or. PointGMM: A neural GMM network for point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12054–12063, 2020.

Matthew G. Hethcoat, David P. Edwards, Joao MB Carreiras, Robert G. Bryant, Filipe M. Franca, and Shaun Quegan. A Machine Learning Approach to Map Tropical Selective Logging. *Remote sensing of environment*, 221:569–582, 2019.

Leon Eklund Hugo, S Quick-Miles, BH Kay, and PA Ryan. Evaluations of mosquito age grading techniques based on morphological changes. *Journal of medical entomology*, 45 (3):353–369, 2014.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

Jeremy Irvin, Hao Sheng, Neel Ramachandran, Sonja Johnson-Yu, Sharon Zhou, Kyle Story, Rose Rustowicz, Cooper Elsworth, Kemen Austin, and Andrew Y. Ng. Forestnet: Classifying Drivers of Deforestation in Indonesia Using Deep Learning on Satellite Imagery. *arXiv preprint arXiv:2011.05479*, 2020.

Krzysztof Janowicz, Song Gao, Grant McKenzie, Yingjie Hu, and Budhendra Bhaduri. GeoAI: Spatially Explicit Artificial Intelligence Techniques for Geographic Knowledge Discovery and Beyond, 2020.

Mario González Jiménez, Simon A Babayan, Pegah Khazaeli, Margaret Doyle, Finlay Walton, Elliott Reedy, Thomas Glew, Mafalda Viana, Lisa Ranford-Cartwright, Abdoulaye Niang, et al. Prediction of mosquito species and population age structure using mid-infrared spectroscopy and supervised machine learning. *Wellcome open research*, 4, 2019.

Brian J Johnson, Leon E Hugo, Thomas S Churcher, Oselyne TW Ong, and Gregor J Devine. Mosquito age grading and vector-control programmes. *Trends in Parasitology*, 36(1):39–51, 2020.

Anuj Karpatne, Imme Ebert-Uphoff, Sai Ravela, Hassan Ali Babaie, and Vipin Kumar. Machine Learning for the Geosciences: Challenges and Opportunities. *IEEE Transactions on Knowledge and Data Engineering*, 31(8):1544–1554, 2018.

Nicolas Keriven and Gabriel Peyré. Universal invariant and equivariant graph neural networks. *Advances in Neural Information Processing Systems*, 32:7092–7101, 2019.

Aazam Khoshmanesh, Dale Christensen, David Perez-Guaita, Inaki Iturbe-Ormaetxe, Scott L O′Neill, Don McNaughton, and Bayden R Wood. Screening of wolbachia endosymbiont infection in aedes aegypti mosquitoes using attenuated total reflection mid-infrared spectroscopy. *Analytical chemistry*, 89(10):5285–5293, 2017.

Felix Klein. *Vergleichende Betrachtungen über neuere geometrische Forschungen: Programm zum Eintritt in die philosophische Facultät und den Senat der k. Friedrich-Alexanders-Universität zu Erlangen*. Deichert, 1872.

Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123*, 2020.

Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: sampling configurations for multi-body systems with symmetric energies. *arXiv preprint arXiv:1910.00753*, 2019.

Risi Kondor. N-body networks: a covariant hierarchical neural network architecture for learning atomic potentials. *arXiv preprint arXiv:1803.01588*, 2018.

Risi Kondor, Hy Truong Son, Horace Pan, Brandon Anderson, and Shubhendu Trivedi. Covariant compositional networks for learning graphs. *arXiv preprint arXiv:1801.02144*, 2018.

Chen Kong and Simon Lucey. Deep non-rigid structure from motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1558–1567, 2019.

Benjamin J Krajacich, Jacob I Meyers, Haoues Alout, Roch K Dabiré, Floyd E Dowell, and Brian D Foy. Analysis of near infrared spectra for age-grading of wild populations of anopheles gambiae. *Parasites & Vectors*, 10(1):1–13, 2017.

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

Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

Dominik Kulon, Haoyang Wang, Riza Alp Güler, Michael M. Bronstein, and Stefanos Zafeiriou. Single image 3D hand reconstruction with mesh convolutions. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2019.

Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4990–5000, 2020.

Ben Lambert, Maggy T Sikulu-Lord, Vale S Mayagaya, Greg Devine, Floyd Dowell, and Thomas S Churcher. Monitoring the age of mosquito populations using near-infrared spectroscopy. *Scientific reports*, 8(1):1–9, 2018.

Leon Lang and Maurice Weiler. A Wigner-Eckart theorem for group equivariant convolution kernels. *International Conference on Learning Representations (ICLR)*, 2020.

Guillaume Leclerc, Andrew Ilyas, Logan Engstrom, Sung Min Park, Hadi Salman, and Aleksander Madry. ffcv. `https://github.com/libffcv/ffcv/`, 2022. commit e97289f.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1 (4):541–551, 1989a. doi: 10.1162/neco.1989.1.4.541.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989b.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Ivo T Leite, Sergey Turtaev, Dirk E Boonzajer Flaes, and Tomáš Čižmár. Observing distant objects with a multimode fiber-based holographic endoscope. *APL Photonics*, 6(3): 036112, 2021.

Timothy M. Lenton. Early Warning of Climate Tipping Points. *Nature climate change*, 1 (4):201–209, 2011.

Mengcheng Li, Liang An, Hongwen Zhang, Lianpeng Wu, Feng Chen, Tao Yu, and Yebin Liu. Interacting attention graph for single image two-hand reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2761–2770, 2022.

Shile Li and Dongheui Lee. Point-to-pose voting based hand pose estimation using residual permutation equivariant layer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11927–11936, 2019.

Shuhui Li, Charles Saunders, Daniel J Lum, John Murray-Bruce, Vivek K Goyal, Tomáš Čižmár, and David B Phillips. Compressively sampling the optical transmission matrix of a multimode fibre. *Light: Science & Applications*, 10(1):1–15, 2021.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

George Macdonald. Epidemiological basis of malaria control. *Bulletin of the World Health Organization*, 15(3-5):613, 1956.

George Macdonald et al. Symposium on insecticides. ii. the objectives of residual insecticide campaigns. *Transactions of the Royal Society of Tropical Medicine and Hygiene*, 46(3), 1952.

Reza Nasiri Mahalati, Ruo Yu Gu, and Joseph M Kahn. Resolution limits for imaging through multi-mode fiber. *Optics express*, 21(2):1656–1668, 2013.

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

Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. In H. Wallach and H. Larochelle and A. Beygelzimer and F. d'Alché-Buc and E. Fox and R. Garnett (ed.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Valeliana S Mayagaya, Kristin Michel, Mark Q Benedict, Gerry F Killeen, Robert A Wirtz, Heather M Ferguson, and Floyd E Dowell. Non-destructive determination of age and species of anopheles gambiae sl using near-infrared spectroscopy. *The American journal of tropical medicine and hygiene*, 81(4):622–630, 2009.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Brendan D McKay and Adolfo Piperno. Practical graph isomorphism, ii. *Journal of symbolic computation*, 60:94–112, 2014.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, pp. 1045–1048. Makuhari, 2010.

Fedor Mitschke. *Fiber optics*. Springer, 2016.

Joshua Mitton, Hans M. Senn, Klaas Wynne, and Roderick Murray-Smith. A Graph VAE and Graph Transformer approach to generating molecular graphs. *ICML Graph Representation Learning and Beyond (GRL+) Workshop*, 2021.

Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *European Conference on Computer Vision*, pp. 752–768. Springer, 2020.

Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2.6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *European Conference on Computer Vision*, pp. 548–564. Springer, 2020.

Oisín Moran, Piergiorgio Caramazza, Daniele Faccio, and Roderick Murray-Smith. Deep, complex, invertible networks for inversion of transmission effects in multimode optical fibres. *Advances in Neural Information Processing Systems*, 31, 2018.

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

Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *ICML Graph Representation Learning and Beyond (GRL+) Workshop*, 2020a.

Christopher Morris, Gaurav Rattan, and Petra Mutzel. Weisfeiler and leman go sparse: Towards scalable higher-order graph embeddings. *Advances in Neural Information Processing Systems*, 33:21824–21840, 2020b.

Vinod Nair and Geoffrey E Hinton. Rectified Linear Units improve Restricted Boltzmann machines. In *ICML*, 2010.

Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pp. 2014–2023, 2016.

Emmy Noether. Invariant variation problems. *Transport theory and statistical physics*, 1(3): 186–207, 1971.

David Novotny, Nikhila Ravi, Benjamin Graham, Natalia Neverova, and Andrea Vedaldi. C3DPO: Canonical 3D pose networks for non-rigid structure from motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7688–7697, 2019.

Ioannis N Papadopoulos, Salma Farahi, Christophe Moser, and Demetri Psaltis. Focusing and scanning light through a multimode optical fiber using digital phase conjugation. *Optics express*, 20(10):10583–10590, 2012.

Pál András Papp, Karolis Martinkus, Lukas Faber, and Roger Wattenhofer. DropGNN: random dropouts increase the expressiveness of graph neural networks. *Advances in Neural Information Processing Systems*, 34, 2021.

Sungheon Park, Minsik Lee, and Nojun Kwak. Procrustean regression networks: Learning 3d structure of non-rigid objects from 2D annotations. In *European Conference on Computer Vision*, pp. 1–18. Springer, 2020.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Darius Phiri, Justin Morgenroth, and Cong Xu. Long-Term Land Cover Change in Zambia: An Assessment of Driving Factors. *Science of The Total Environment*, 697:134206, 2019.

Martin Plöschner, Tomáš Tyc, and Tomáš Čižmár. Seeing through chaos in multimode fibres. *Nature Photonics*, 9(8):529–535, 2015.

Ate Poortinga, Karis Tenneson, Aurélie Shapiro, Quyen Nquyen, Khun San Aung, Farrukh Chishtie, and David Saah. Mapping Plantations in Myanmar by Fusing Landsat-8, Sentinel-2 and Sentinel-1 Data along with Systematic Error Quantification. *Remote Sensing*, 11(7):831, 2019.

Babak Rahmani, Damien Loterie, Georgia Konstantinou, Demetri Psaltis, and Christophe Moser. Multimode optical fiber transmission with a deep learning network. *Light: Science & Applications*, 7(1):1–11, 2018.

Gaurav Rattan and Tim Seppelt. Weisfeiler–leman, graph spectra, and random walks. *arXiv preprint arXiv:2103.02972*, 2021.

Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. Equivariance through parameter-sharing. In *International Conference on Machine Learning*, pp. 2892–2901. PMLR, 2017.

Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (ToG)*, 36(6):1–17, 2017.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *International Conference on Learning Representations*, 2020.

Yu Rong, Jingbo Wang, Ziwei Liu, and Chen Change Loy. Monocular 3D reconstruction of interacting hands via collision-aware factorized refinements. In *2021 International Conference on 3D Vision (3DV)*, pp. 432–441. IEEE, 2021.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

David P. Roy, Michael A. Wulder, Thomas R. Loveland, Curtis E. Woodcock, Richard G. Allen, Martha C. Anderson, Dennis Helder, James R. Irons, David M. Johnson, Robert Kennedy, et al. Landsat-8: Science and Product Vision for Terrestrial Global Change Research. *Remote sensing of Environment*, 145:154–172, 2014.

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

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

Y Schlein. Age grouping of anopheline malaria vectors (diptera: Culicidae) by the cuticular growth lines. *Journal of Medical Entomology*, 16(6):502–506, 1979.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Maggy T Sikulu, James Monkman, Keyur A Dave, Marcus L Hastie, Patricia E Dale, Roger L Kitching, Gerry F Killeen, Brian H Kay, Jeffry J Gorman, and Leon E Hugo. Mass spectrometry identification of age-associated proteins from the malaria mosquitoes anopheles gambiae ss and anopheles stephensi. *Data in brief*, 4:461–467, 2015.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3693–3702, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2015.

Doreen J Siria, Roger Sanou, Joshua Mitton, Emmanuel P Mwanga, Abdoulaye Niang, Issiaka Sare, Paul CD Johnson, Geraldine M Foster, Adrien MG Belem, Klaas Wynne, et al. Rapid age-grading and species identification of natural mosquitoes for malaria surveillance. *Nature communications*, 13(1):1–9, 2022.

Damir Sorak, Lars Herberholz, Sylvia Iwascek, Sedakat Altinpinar, Frank Pfeifer, and Heinz W Siesler. New developments and applications of handheld raman, mid-infrared, and near-infrared spectrometers. *Applied Spectroscopy Reviews*, 47(2):83–115, 2012.

Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 89–98, 2018.

Nicolino Stasio. Multimode fiber optical imaging using wavefront control. Technical report, EPFL, 2017.

Jean-Daniel Sylvain, Guillaume Drolet, and Nicolas Brown. Mapping Dead Forest Cover Using a Deep Convolutional Neural Network and Digital Aerial Photography. *ISPRS Journal of Photogrammetry and Remote Sensing*, 156:14–26, 2019.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

Erik Thiede, Wenda Zhou, and Risi Kondor. Autobahn: Automorphism-based graph neural nets. *Advances in Neural Information Processing Systems*, 34:29922–29934, 2021.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3D point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 210–218. Springer, 2018.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *International Conference on Learning Representations*, 2018.

Charles Verpoorter, Tiit Kutser, and Lars Tranvik. Automated Mapping of Water Bodies Using Landsat Multispectral Data. *Limnology and Oceanography: Methods*, 10(12): 1037–1050, 2012.

Charles Verpoorter, Tiit Kutser, David A. Seekell, and Lars J. Tranvik. A Global Inventory of Lakes Based on High-Resolution Satellite Imagery. *Geophysical Research Letters*, 41 (18):6396–6402, 2014.

Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8684–8694, 2020.

Ronald W Waynant, Ilko K Ilev, and Israel Gannot. Mid–infrared laser applications in medicine and biology. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 359(1780):635–644, 2001.

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

Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco S Cohen. 3D steerable CNNs: Learning rotationally equivariant features in volumetric data. In *Advances in Neural Information Processing Systems*, pp. 10381–10392, 2018.

Boris Weisfeiler and Andrei Leman. The reduction of a graph to canonical form and the algebra which appears therein. *NTI, Series*, 2(9):12–16, 1968.

Max Welling and Thomas N Kipf. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*, 2016.

M.K.F. Wong. Unitary representations of $SO(n, 1)$. *Journal of Mathematical Physics*, 15(1): 25–30, 1974.

Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5028–5037, 2017.

Wenxuan Wu, Zhongang Qi, and Li Fuxin. PointConv: Deep convolutional networks on 3D point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9621–9630, 2019.

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

Fengli Xu, Quanming Yao, Pan Hui, and Yong Li. Automorphic equivalence-aware graph neural network. *Advances in Neural Information Processing Systems*, 34:15138–15150, 2021.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

Pinar Yanardag and SVN Vishwanathan. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1365–1374, 2015.

Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in neural information processing systems*, pp. 4800–4810, 2018.

Baowen Zhang, Yangang Wang, Xiaoming Deng, Yinda Zhang, Ping Tan, Cuixia Ma, and Hongan Wang. Interacting two-hand 3D pose and shape reconstruction from single color image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11354–11363, 2021.

Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular rgb image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2354–2364, 2019.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16259–16268, 2021.

Lingxiao Zhao, Wei Jin, Leman Akoglu, and Neil Shah. From stars to subgraphs: Uplifting any GNN with local structure awareness. *International Conference on Learning Representations (ICLR)*, 2022.

Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. Monocular real-time hand shape and motion capture using multi-modal data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5346–5355, 2020.

Christian Zimmermann and Thomas Brox. Learning to estimate 3D hand pose from single RGB images. In *Proceedings of the IEEE international conference on computer vision*, pp. 4903–4911, 2017.

Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single RGB images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 813–822, 2019.

A

Code Appendix

We provide a static version of the code and data used for each experiment and to produce each of the figures in the thesis here: http://dx.doi.org/10.5525/gla.researchdata.1349

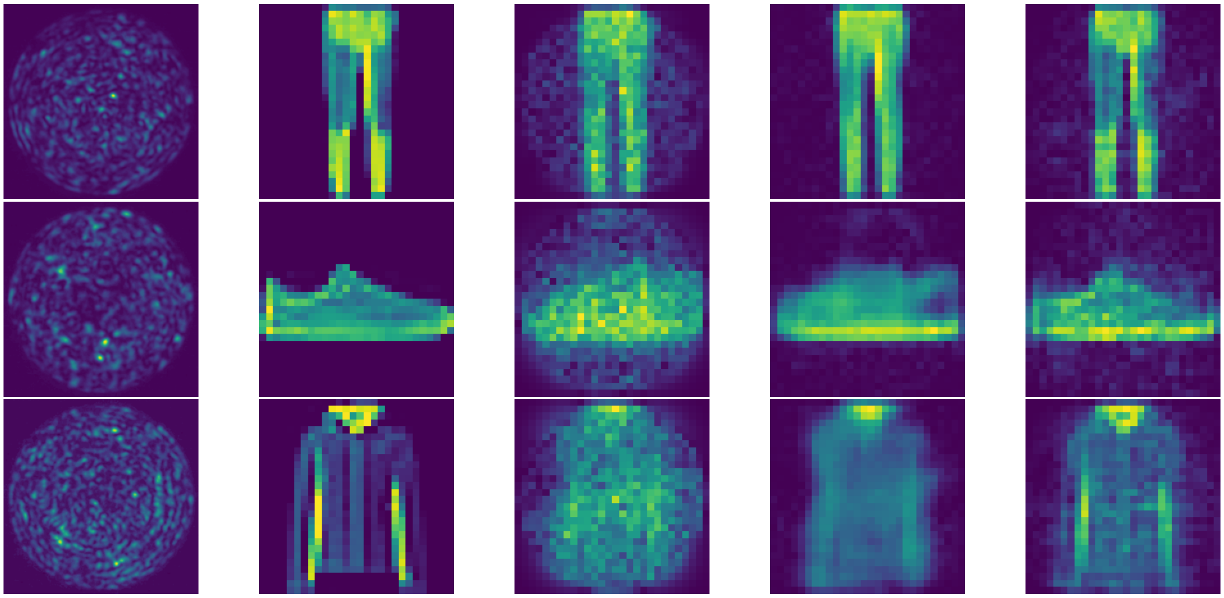
In addition to the static version maintained public github repositories can be found here: https://github.com/JoshuaMitton

B

MMF Appendix

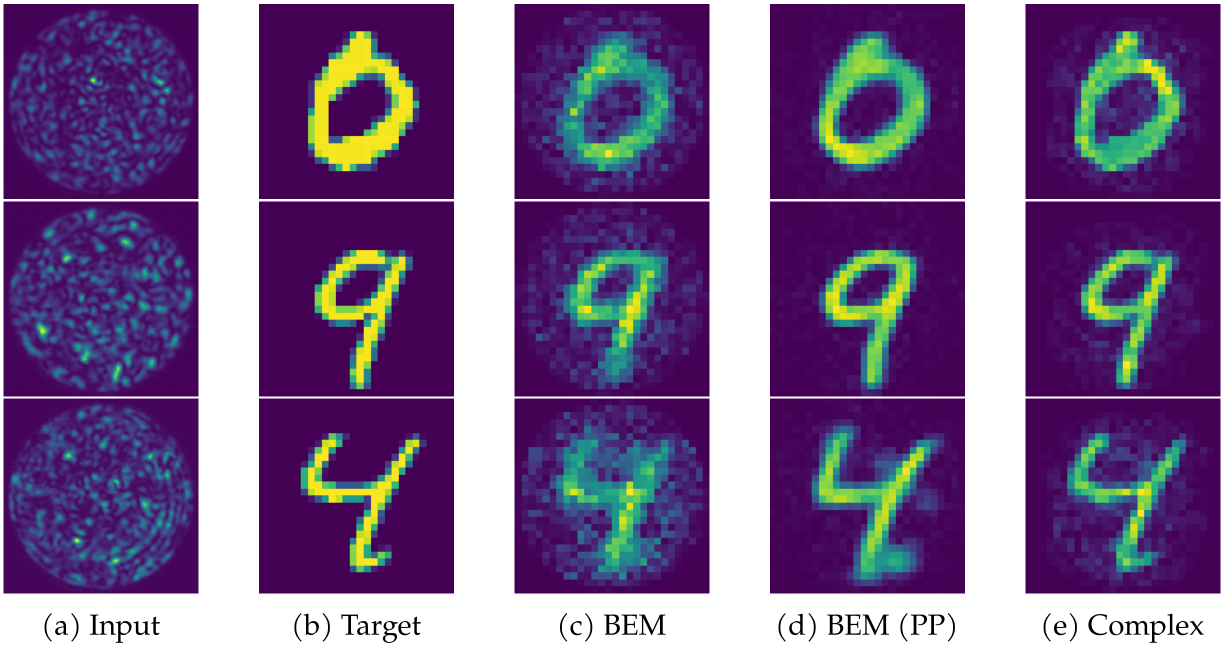
B.1 Lab Based MMF – Inversions

This section provides further visualisations for the task of inverting the transmission effects of a fibre in a lab based context. Further examples of the reconstructions are provided for both MNIST and fMNIST datasets in Figures B.1 and B.2.



(a) Input (b) Target (c) BEM (d) BEM (PP) (e) Complex

Figure B.1: Comparison of predicted images from inverting transmission effects of a MMF for fMNIST data. (a) Input speckled image, (b) the target original image to reconstruct, (c) Output of the Bessel equivariant model, (d) Output of the combination of Bessel equivariant and post-processing model, and (e) the output of the complex-valued linear model.



(a) Input (b) Target (c) BEM (d) BEM (PP) (e) Complex

Figure B.2: Comparison of predicted images from inverting transmission effects of a MMF for MNIST data. (a) Input speckled image, (b) the target original image to reconstruct, (c) Output of the Bessel equivariant model, (d) Output of the combination of Bessel equivariant and post-processing model, and (e) the output of the complex-valued linear model.

B.2 Lab Based MMF – Accounting for Losses in a Lab Based System

This section provides further visualisations for the task of inverting the transmission effects of a fibre in a lab based context, where the diagonal constraint of the complex weight matrix is relaxed. Further examples of the reconstructions are provided for both MNIST and fMNIST datasets in Figures B.3 and B.4.

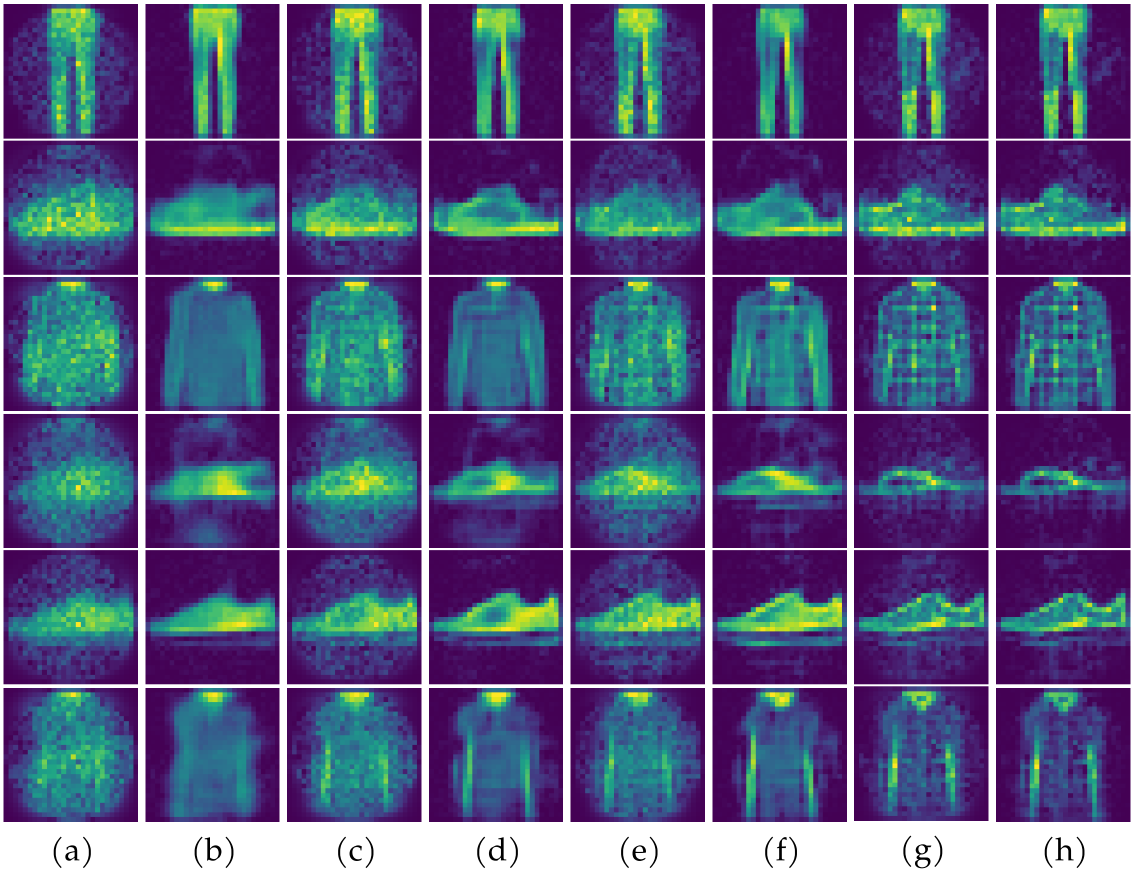


Figure B.3: Comparison of predicted images from inverting transmission effects of a MMF with varying relaxations on the assumption the fibre has a diagonal fibre propagation matrix for fMNIST data. (a, c, e, g) Reconstructions using the Bessel equivariant model only. (b, d, f, h) Reconstructions using the Bessel equivariant model and post-processing model. (a, b) Reconstructions when the complex weight matrix is diagonal. (c, d) Reconstructions when the complex weight matrix has five block diagonal structure. (e, f) Reconstructions when the complex weight matrix has ten block diagonal structure. (g, h) Reconstructions when the complex weight matrix is full.

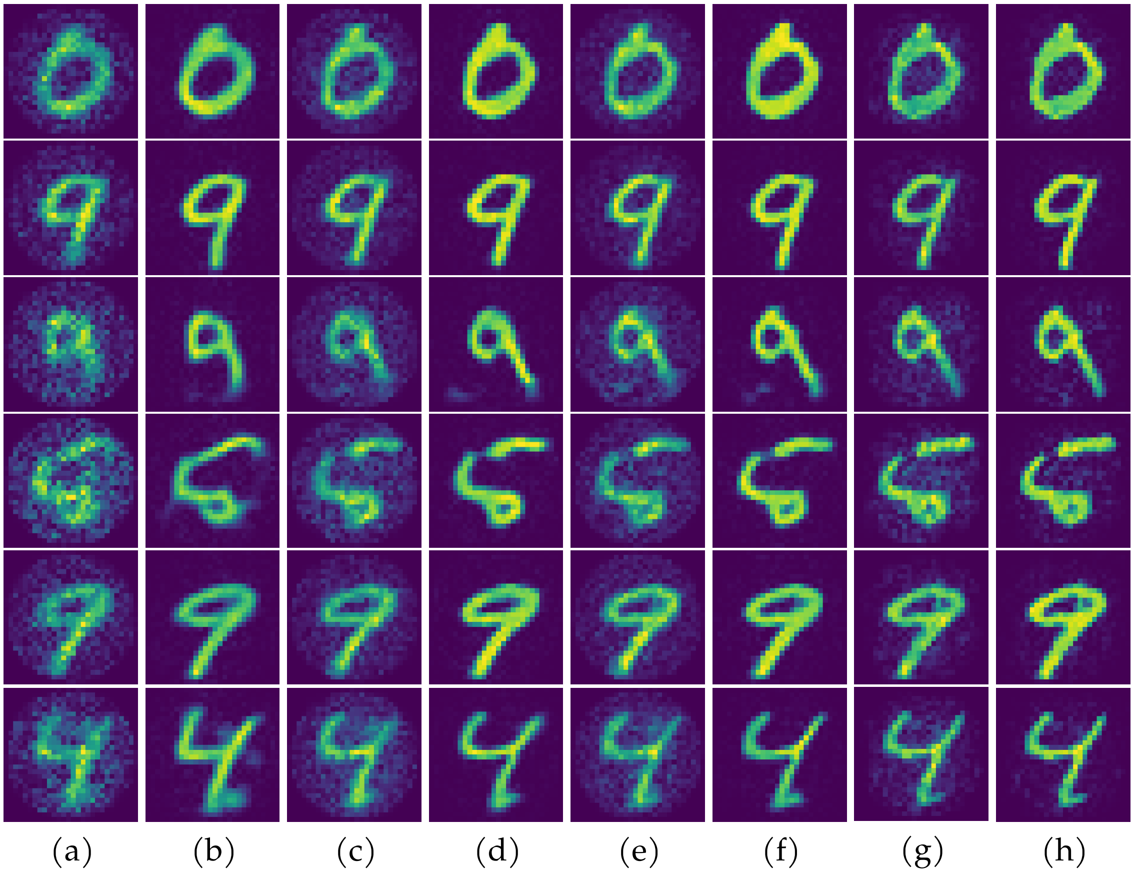


Figure B.4: Comparison of predicted images from inverting transmission effects of a MMF with varying relaxations on the assumption the fibre has a diagonal fibre propagation matrix for MNIST data. (a, c, e, g) Reconstructions using the Bessel equivariant model only. (b, d, f, h) Reconstructions using the Bessel equivariant model and post-processing model. (a, b) Reconstructions when the complex weight matrix is diagonal. (c, d) Reconstructions when the complex weight matrix has five block diagonal structure. (e, f) Reconstructions when the complex weight matrix has ten block diagonal structure. (g, h) Reconstructions when the complex weight matrix is full.

B.3 Theoretical TM - Inversions

This section provides further visualisations for the task of inverting the transmission effects of a theoretical fibre for fMNIST data and the corresponding speckled patterns. Further examples of the reconstructions are provided in Figure B.5.

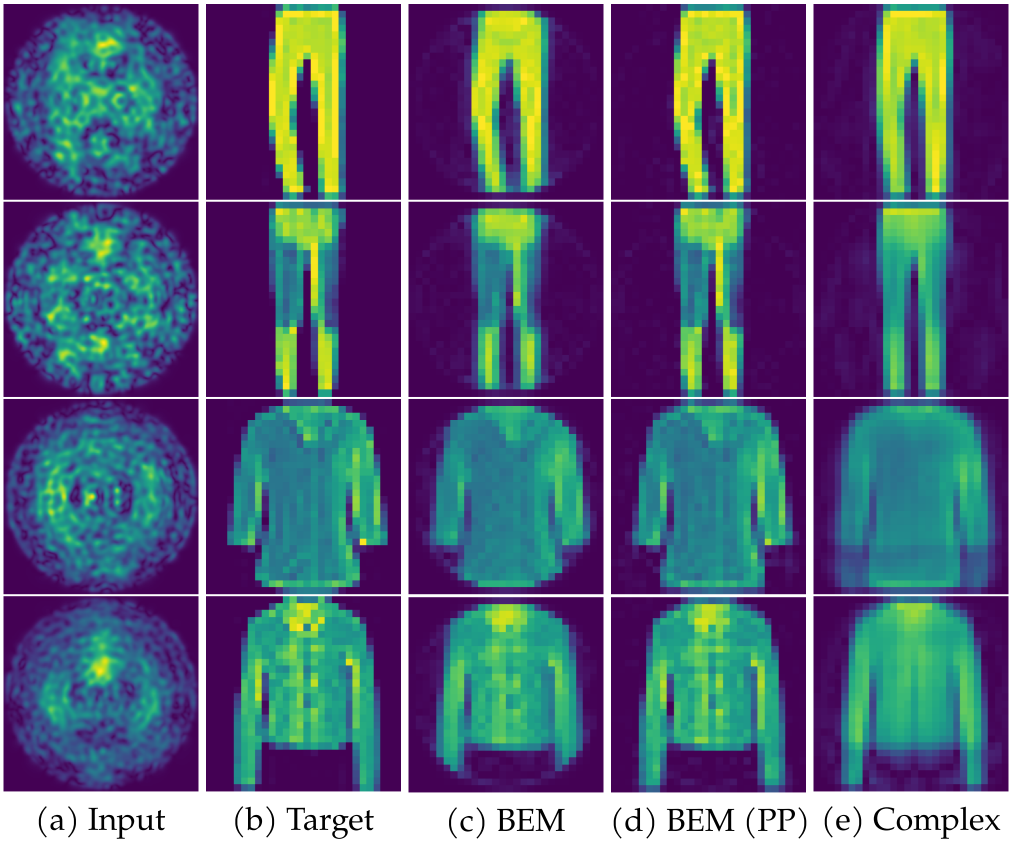


Figure B.5: Comparison of predicted images from inverting transmission effects of a theoretical MMF with fMNIST data. (a) Input speckled image, (b) Target original image to reconstruct, (c) Output of the Bessel equivariant model, (d) Output of the Bessel equivariant and post-processing model, and (e) Output of the complex valued linear model.

B.4 Theoretical TM – Generalising To Out-Of-Distribution Data

This section provides further visualisations for the task of inverting the transmission effects of a theoretical fibre. Here predictions are made on the out-of-distribution MNIST data and the corresponding speckled patterns, which are not similar to the examples seen during training. Further examples of the reconstructions are provided in Figure B.6.

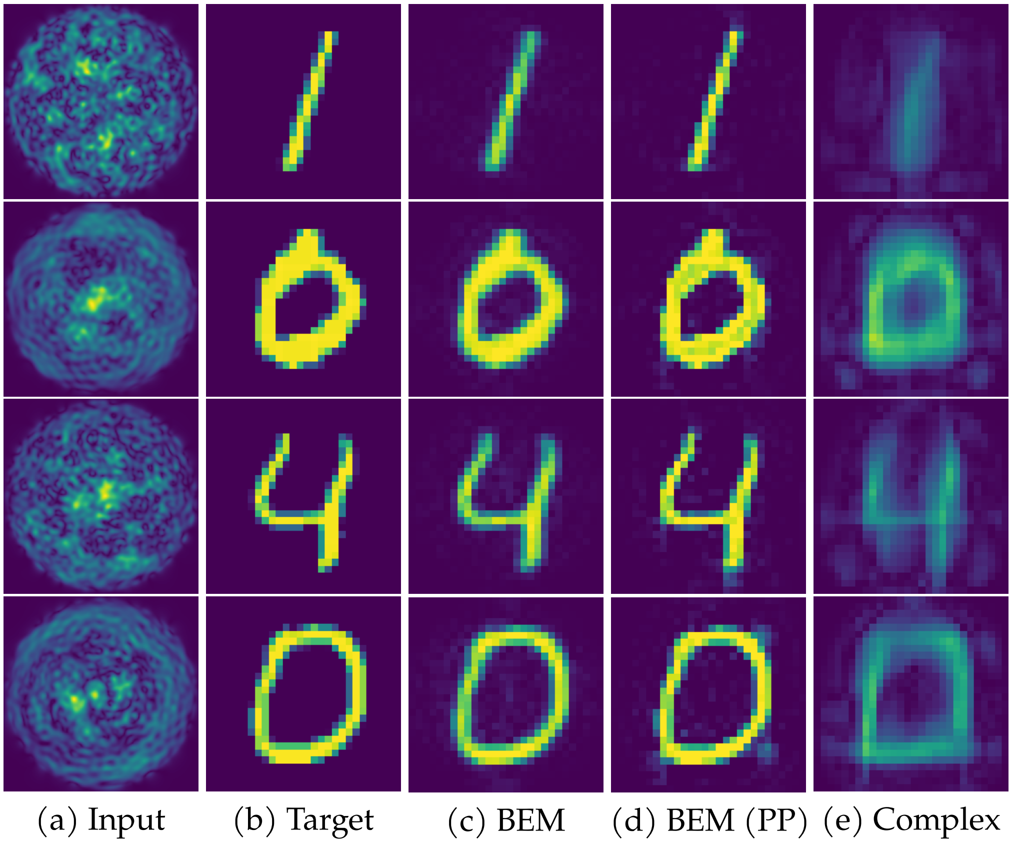


Figure B.6: Comparison of predicted images from inverting transmission effects of a MMF for models trained with fMNIST data and tested on MNIST data. (a) Input speckled images, (b) Target original images, (c) Output of the Bessel equivariant model, (d) Output of Bessel equivariant and post-processing model, and (e) Output of the complex valued linear model.

B.5 Theoretical TM – Under Parameterising the Bases Set

This section provides further visualisations for the task of inverting the transmission effects of a theoretical fibre when the basis set within our Bessel equivariant model is under parameterised in Figures B.7, Figures B.8, and Figures B.9.

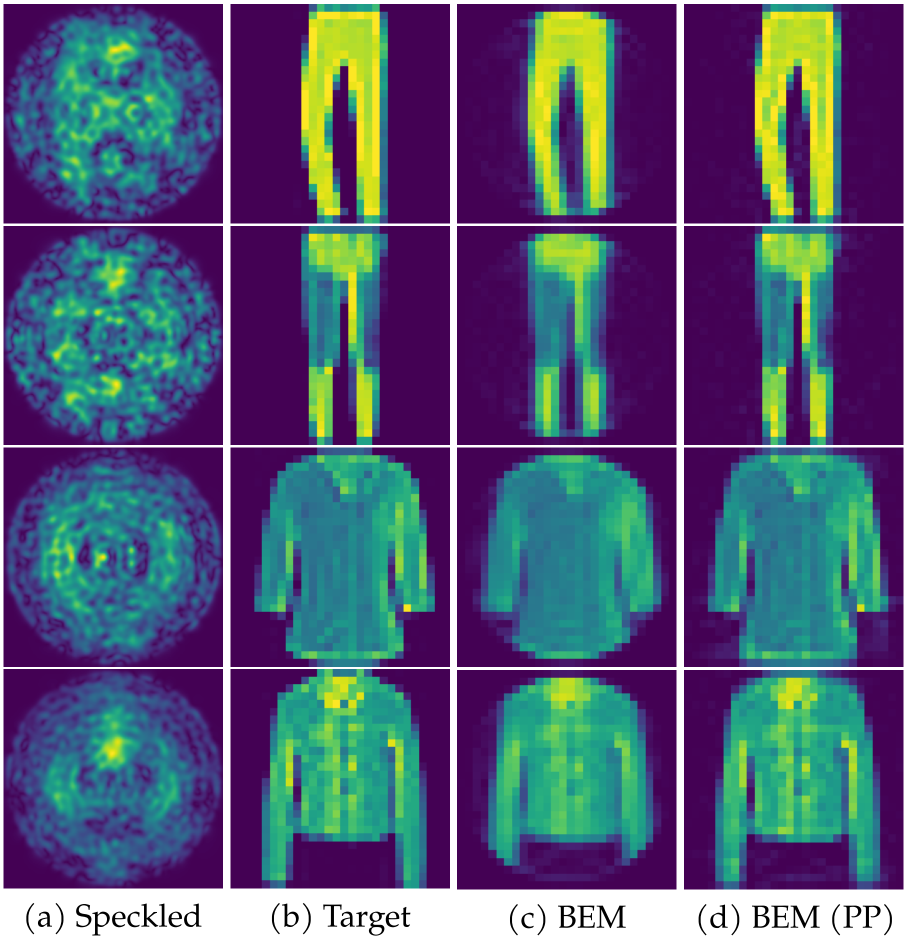
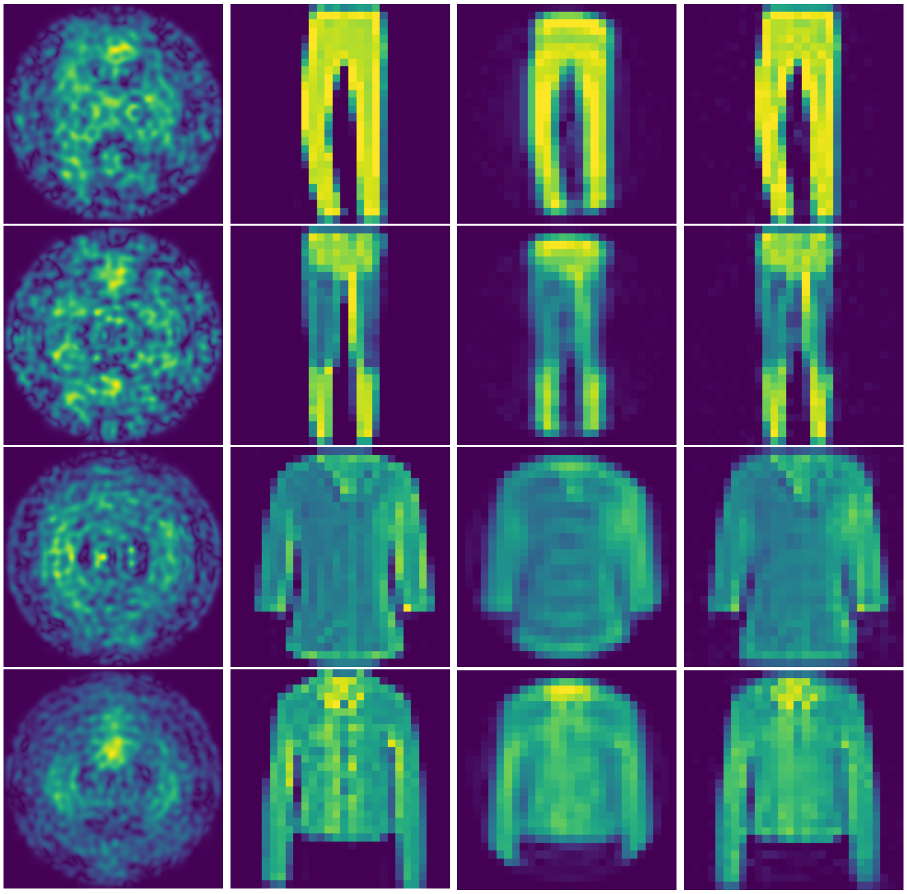
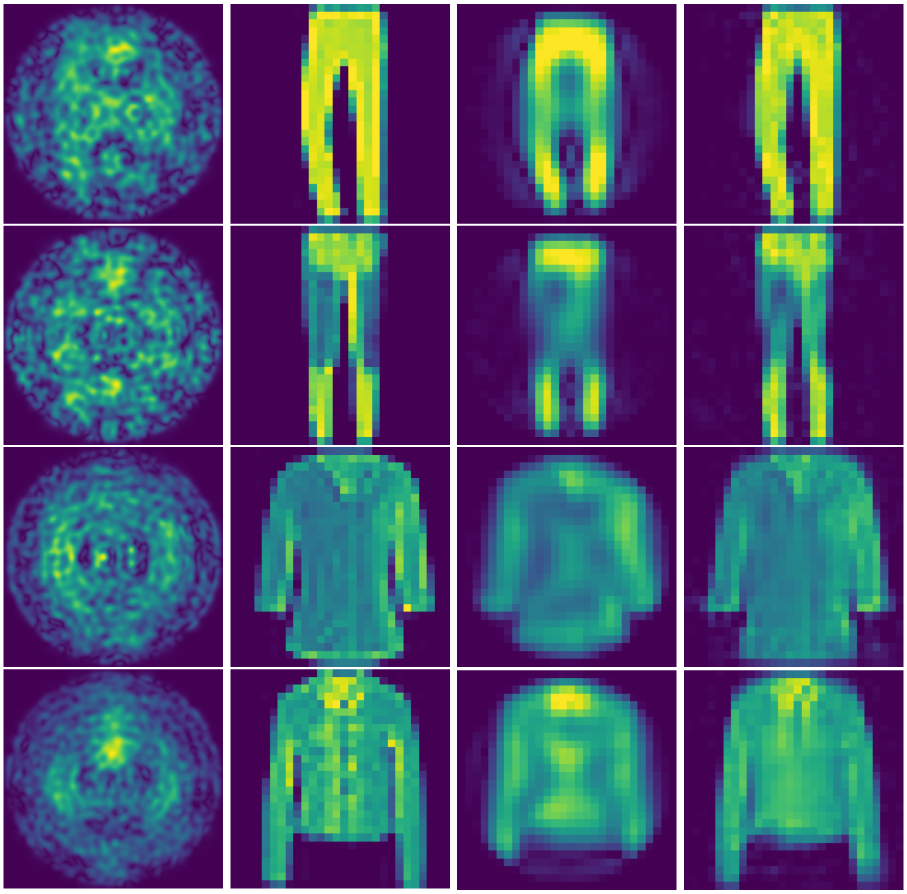


Figure B.7: Comparison of predicted images from inverting transmission effects of a MMF, when reducing the number of radial frequencies present in the Bessel function bases from 21 to 14. This represents a reduction in the number of bases from the original value of 1061 to 930. (a) The input speckled image. (b) The target image. (c) The output of the Bessel equivariant model. (d) The output of the Bessel equivariant model and post-processing model. The data used was fMNIST with speckled patterns created with a theoretical TM.



(a) Speckled (b) Target (c) BEM (d) BEM (PP)

Figure B.8: Comparison of predicted images from inverting transmission effects of a MMF, when reducing the number of radial frequencies present in the Bessel function bases from 21 to 7. This represents a reduction in the number of bases from the original value of 1061 to 567. (a) The input speckled image. (b) The target image. (c) The output of the Bessel equivariant model. (d) The output of the Bessel equivariant model and post-processing model. The data used was fMNIST with speckled patterns created with a theoretical TM.

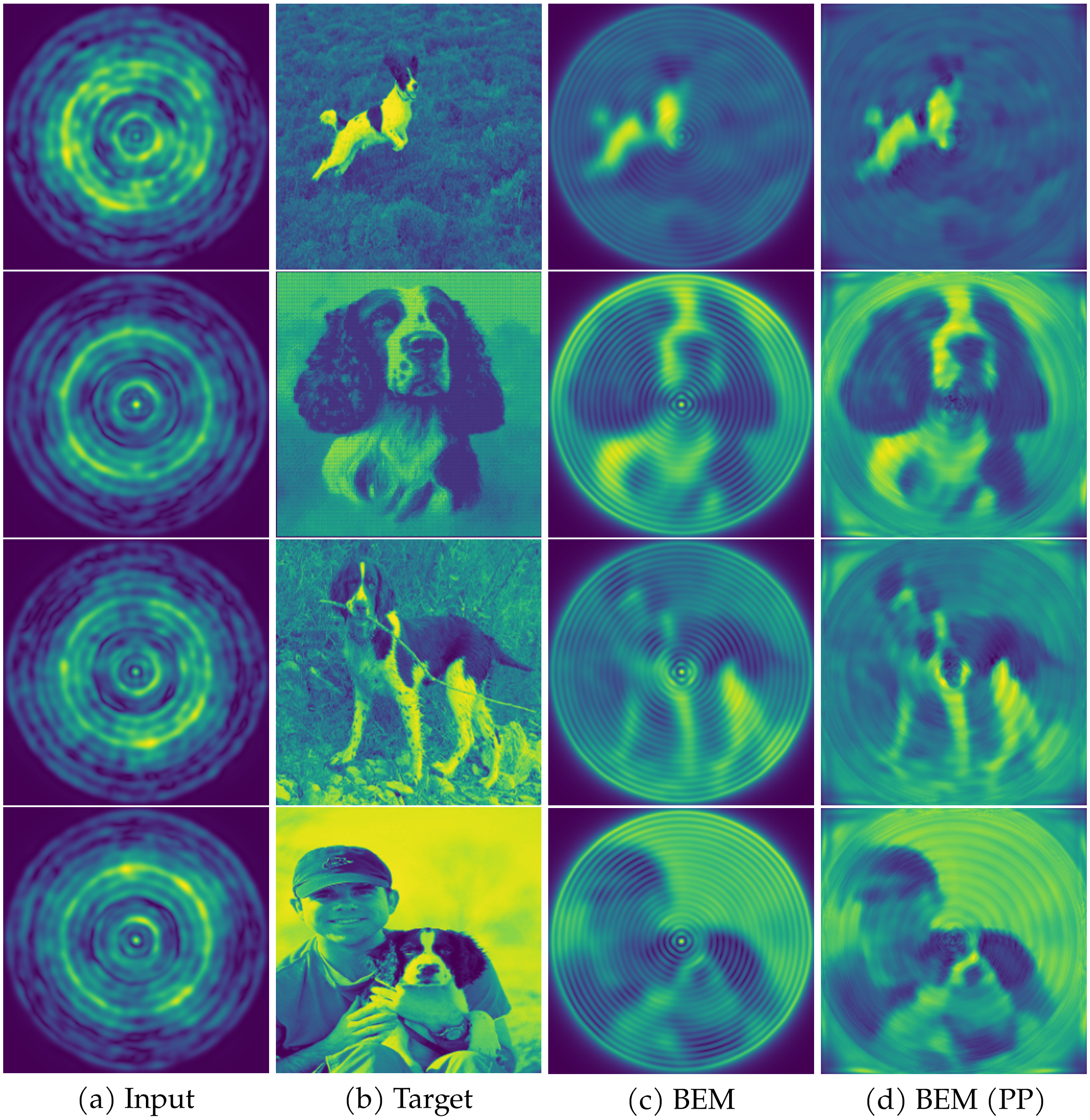


(a) Speckled (b) Target (c) BEM (d) BEM (PP)

Figure B.9: Comparison of predicted images from inverting transmission effects of a MMF, when reducing the number of radial frequencies present in the Bessel function bases from 21 to 4. This represents a reduction in the number of bases from the original value of 1061 to 322. (a) The input speckled image. (b) The target image. (c) The output of the Bessel equivariant model. (d) The output of the Bessel equivariant model and post-processing model. The data used was fMNIST with speckled patterns created with a theoretical TM.

B.6 Theoretical TM – Scaling to Larger Images – ImageNet

This section provides further visualisations for the task of inverting speckled images created with a theoretical fibre to imagenet images in Figure B.10



(a) Input (b) Target (c) BEM (d) BEM (PP)

Figure B.10: Comparison of predicted images from inverting transmission effects of a MMF using high resolution ImageNet data. (a) Input speckled image, (b) Target original image to reconstruct, (c) Output of Bessel equivariant model (d) Output of Bessel equivariant and post-processing model.

C

Hands Appendix

C.1 Additional Real-World Reconstructions

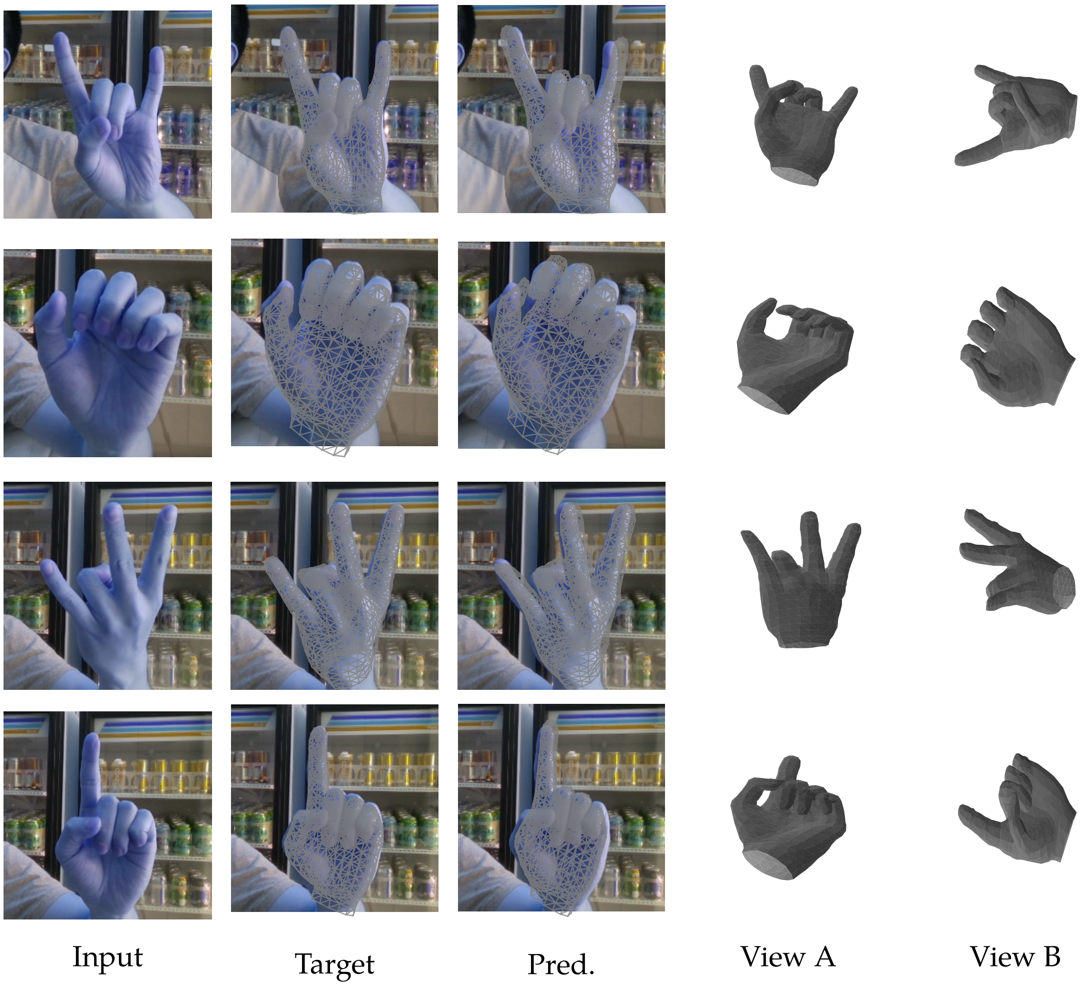


Figure C.1: Qualitative mesh reconstruction results on the testing dataset from the real-world dataset of Ge et al. (2019) comparing our model's predictions to the target meshes.

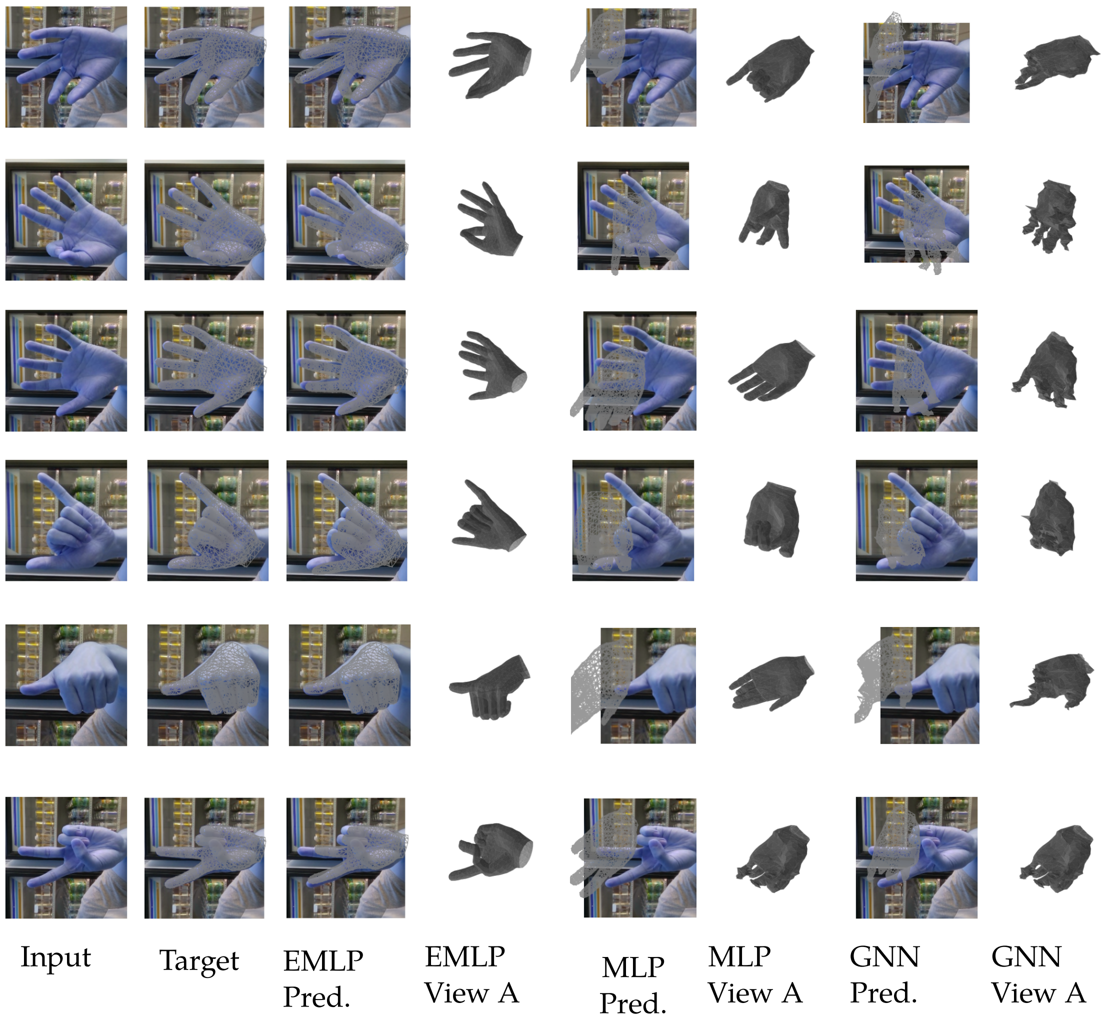


Figure C.2: Qualitative mesh reconstruction results on rotated testing hand images from the testing dataset for the EMLP, MLP, and GNN models.

C.2 Additional FreiHand Reconstructions

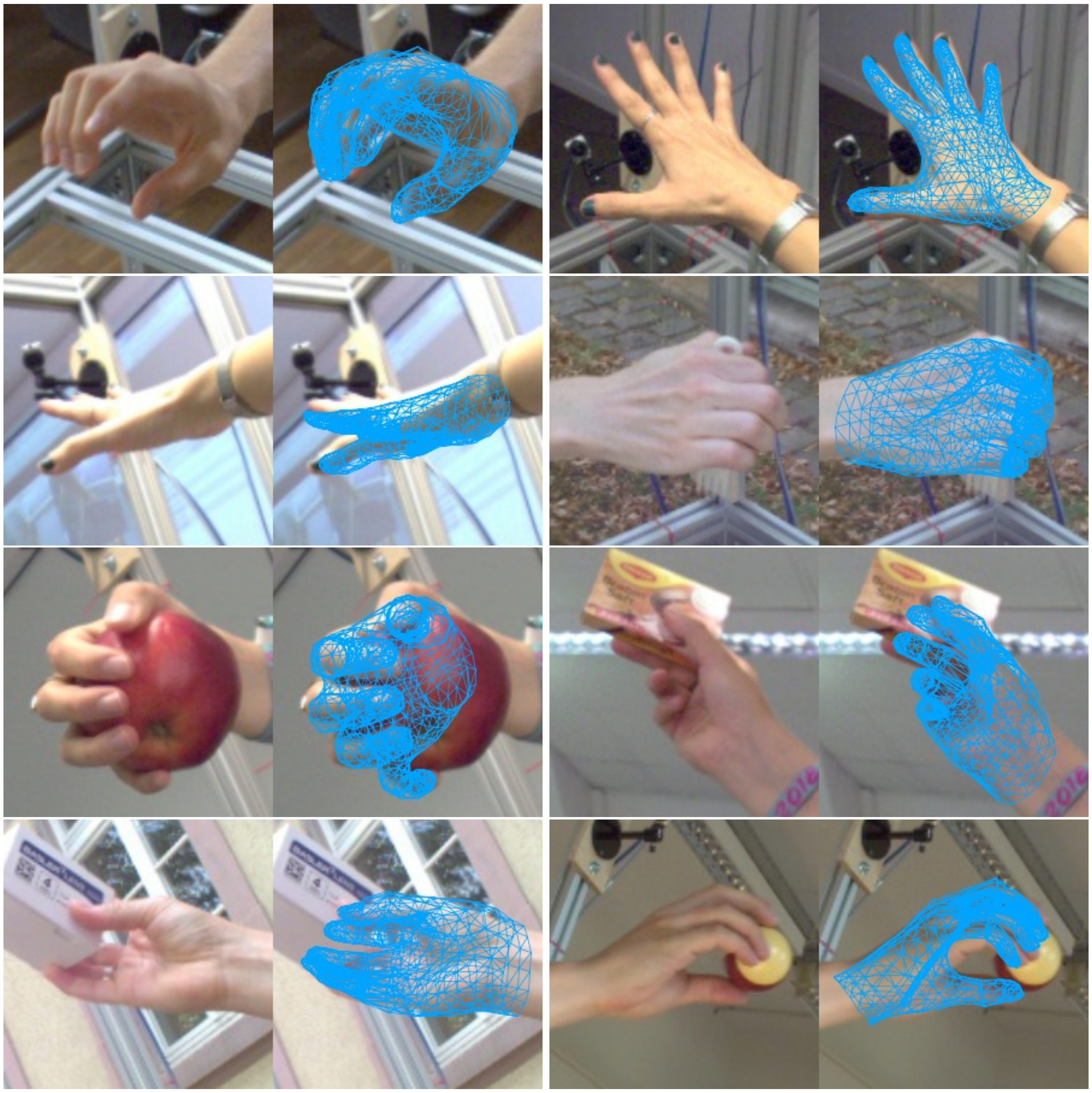


Figure C.3: Qualitative mesh reconstruction results on the testing dataset from the FreiHand dataset of Zimmermann et al. (2019).

D

GNN Appendix

D.1 Further Results Discussion

In addition to the comparison across datasets in Table 6.5 Figures D.1, D.2, D.3, D.4, D.5, D.6, and D.7 show the test accuracy distribution of the SPEN method and compares to other methods from Table 6.5. This shows for the smaller datasets that the spread of test accuracy's is larger leading to our method and others presenting large standard deviations over the results. Comparing the SPEN results to the other methods here highlights that the SPEN result is competitive across a range of datasets. For the NCI1 and NCI109 datasets the distribution of results of our method highlights the strong performance of the SPEN method. For IMDBB and IMDBM the distribution of results for the SPEN method also highlights that it is competitive on these datasets.

We also present analysis of the statistical significance of test accuracies from each method across the seven datasets considered. Here we make use of the Welch's ANOVA method for comparing the means of multiple scores with different variances. We consider the null hypothesis that the means are equal and to reject this null hypothesis the p value is required to be below 0.05. Tables D.1 and D.2 show that for leading methods SPEN, CIN, GSN, CCN, and DSS the p values between each of these methods is greater than 0.05 and therefore we cannot reject the null hypothesis, and thus conclude that the means are not significantly different. Despite this our SPEN method does produce statistically significantly better results than some benchmark results. Table D.3 compares the statistical significance of results on the Proteins dataset, which is one where our method appears to perform more poorly ranking 15th for mean values. Here we show that when comparing the leading results of PPGN, CIN, IGN, DSS, GSN, and SIN there is no statistical significance between our mean accuracy and theirs. Therefore we can conclude that there is no significant difference between the means and our method is comparable with SOTA results. Tables D.4, D.5, and D.6 also show that for leading methods the p values between each of these methods is greater than 0.05 and therefore we cannot reject the null hypothesis, and thus conclude that the means are not significantly different. Despite this our SPEN method does produce statistically significantly better results than some benchmark results. Finally, Table D.7 does show statistically significant results between our method and leading methods and therefore our method is underperforming on this dataset. This is something we seeks to explore and improve in the future. Overall the analysis of the statistical significance of results by our SPEN method and other benchmark results highlights that the mean accuracy's from recent leading methods are not significantly different. This is the same across each recent method claiming SOTA on some benchmark datasets and not an exclusive result to our method. This highlights that our method is competitive with SOTA methods across a range of benchmarks.

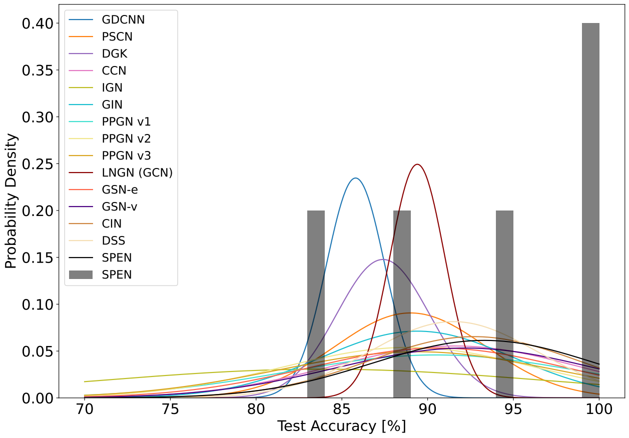


Figure D.1: Comparison between our SPEN method and other methods on the MUTAG dataset. Results for the SPEN method are also presented as a histogram of the 10-fold runs. Each other method is given as a Gaussian distribution with mean and standard deviation as is presented in Table 6.5.

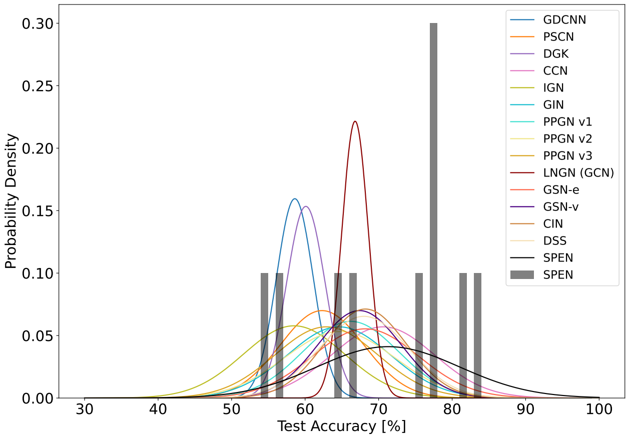


Figure D.2: Comparison between our SPEN method and other methods on the PTC dataset. Results for the SPEN method are also presented as a histogram of the 10-fold runs. Each other method is given as a Gaussian distribution with mean and standard deviation as is presented in Table 6.5.

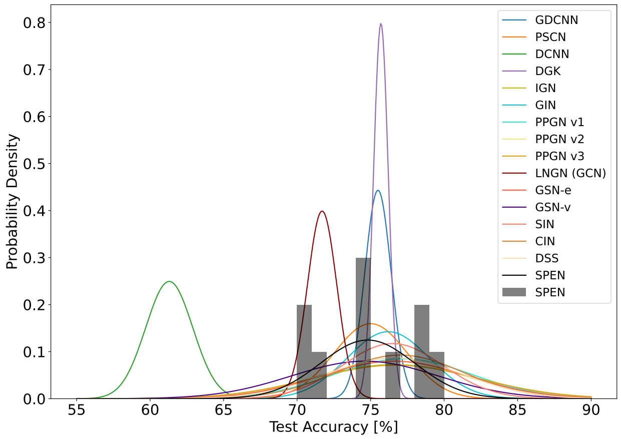


Figure D.3: Comparison between our SPEN method and other methods on the PROTEINS dataset. Results for the SPEN method are also presented as a histogram of the 10-fold runs. Each other method is given as a Gaussian distribution with mean and standard deviation as is presented in Table 6.5.

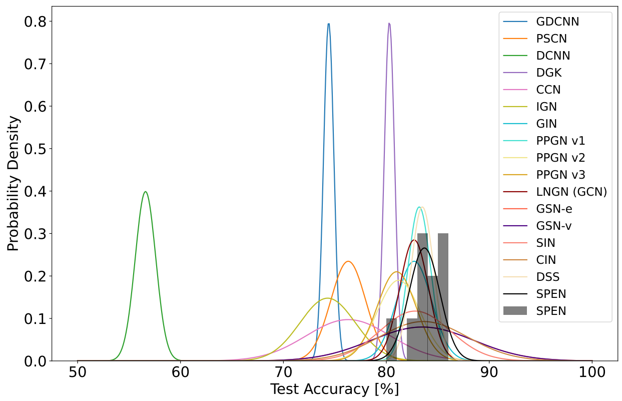


Figure D.4: Comparison between our SPEN method and other methods on the NCI1 dataset. Results for the SPEN method are also presented as a histogram of the 10-fold runs. Each other method is given as a Gaussian distribution with mean and standard deviation as is presented in Table 6.5.

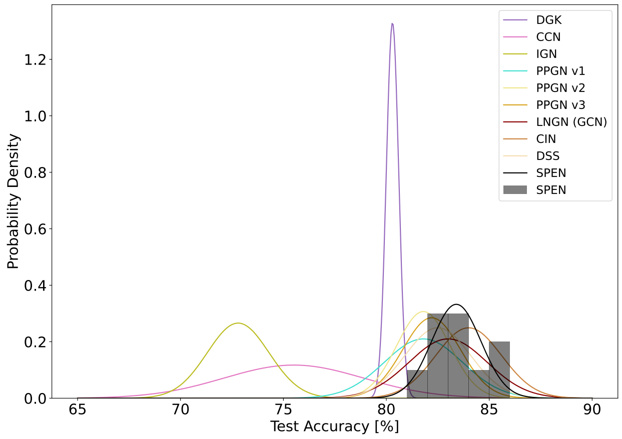


Figure D.5: Comparison between our SPEN method and other methods on the NCI109 dataset. Results for the SPEN method are also presented as a histogram of the 10-fold runs. Each other method is given as a Gaussian distribution with mean and standard deviation as is presented in Table 6.5.

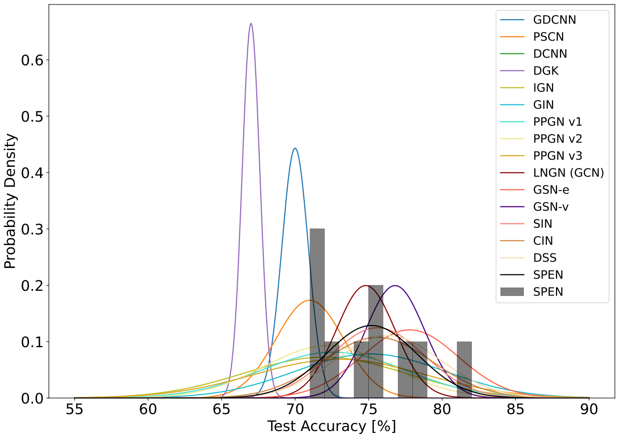


Figure D.6: Comparison between our SPEN method and other methods on the IMDB-B dataset. Results for the SPEN method are also presented as a histogram of the 10-fold runs. Each other method is given as a Gaussian distribution with mean and standard deviation as is presented in Table 6.5.

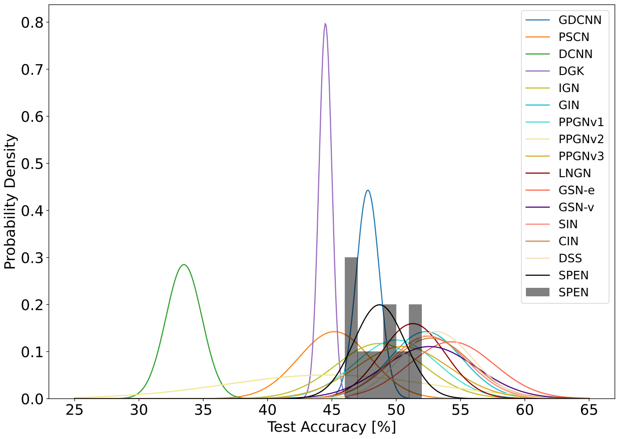


Figure D.7: Comparison between our SPEN method and other methods on the IMDB-M dataset. Results for the SPEN method are also presented as a histogram of the 10-fold runs. Each other method is given as a Gaussian distribution with mean and standard deviation as is presented in Table 6.5.

Table D.1: Statistical significant analysis of results on the MUTAG dataset given as p-values. The null hypothesis is that each model produces the same accuracy. A p-value of less than 0.05 is required to reject this null hypothesis and thus conclude that two models produce different accuracy's.

	GDCNN	PSCN	DGK	CCN	IGN	GIN	PPGN v1	PPGN v2	PPGN v3	LNGN	GSN-e	GSN-v	CIN	DSS	SPEN
GDCNN		0.05	0.13	0.03	0.66	0.08	0.13	0.23	0.20	0.00	0.08	0.03	0.01	0.01	0.01
PSCN	0.05		0.34	0.35	0.26	0.86	0.63	0.97	0.89	0.79	0.57	0.26	0.14	0.25	0.10
DGK	0.13	0.34		0.11	0.42	0.33	0.31	0.56	0.47	0.06	0.23	0.08	0.03	0.04	0.02
CCN	0.03	0.35	0.11		0.12	0.46	0.76	0.42	0.53	0.37	0.76	0.86	0.72	0.97	0.59
IGN	0.66	0.26	0.42	0.12		0.24	0.20	0.31	0.27	0.22	0.18	0.10	0.08	0.11	0.06
GIN	0.08	0.86	0.33	0.46	0.24		0.74	0.87	1.00	1.00	0.69	0.36	0.22	0.38	0.17
PPGN v1	0.13	0.63	0.31	0.76	0.20	0.74		0.66	0.77	0.70	0.98	0.65	0.52	0.76	0.43
PPGN v2	0.23	0.97	0.56	0.42	0.31	0.87	0.66		0.89	0.84	0.62	0.34	0.23	0.37	0.18
PPGN v3	0.20	0.89	0.47	0.53	0.27	1.00	0.77	0.89		1.00	0.74	0.43	0.32	0.49	0.25
LNGN	0.00	0.79	0.06	0.37	0.22	1.00	0.70	0.84	1.00		0.63	0.28	0.13	0.22	0.09
GSN-e	0.08	0.57	0.23	0.76	0.18	0.69	0.98	0.62	0.74	0.63		0.64	0.50	0.75	0.40
GSN-v	0.03	0.26	0.08	0.86	0.10	0.36	0.65	0.34	0.43	0.28	0.64		0.87	0.81	0.73
CIN	0.01	0.14	0.03	0.72	0.08	0.22	0.52	0.23	0.32	0.13	0.50	0.87		0.63	0.83
DSS	0.01	0.25	0.04	0.97	0.11	0.38	0.76	0.37	0.49	0.22	0.75	0.81	0.63		0.49
SPEN	0.01	0.10	0.02	0.59	0.06	0.17	0.43	0.18	0.25	0.09	0.40	0.73	0.83	0.49	

Table D.2: Statistical significant analysis of results on the PTC dataset given as p-values. The null hypothesis is that each model produces the same accuracy. A p-value of less than 0.05 is required to reject this null hypothesis and thus conclude that two models produce different accuracy's.

	GDCNN	PSCN	DGK	CCN	IGN	GIN	PPGN v1	PPGN v2	PPGN v3	LNGN	GSN-e	GSN-v	CIN	DSS	SPEN
GDCNN		0.08	0.21	0.00	0.97	0.03	0.01	0.03	0.09	0.00	0.00	0.00	0.00	0.00	0.00
PSCN	0.08		0.29	0.01	0.20	0.43	0.17	0.43	0.84	0.04	0.06	0.06	0.03	0.04	0.02
DGK	0.21	0.29		0.00	0.51	0.08	0.02	0.09	0.26	0.00	0.01	0.00	0.00	0.00	0.01
CCN	0.00	0.01	0.00		0.00	0.07	0.16	0.09	0.02	0.13	0.46	0.28	0.41	0.39	0.86
IGN	0.97	0.20	0.51	0.00		0.07	0.02	0.07	0.17	0.00	0.01	0.01	0.00	0.00	0.00
GIN	0.03	0.43	0.08	0.07	0.07		0.60	0.98	0.59	0.36	0.27	0.34	0.22	0.26	0.10
PPGN v1	0.01	0.17	0.02	0.16	0.02	0.60		0.64	0.29	0.78	0.52	0.67	0.47	0.53	0.19
PPGN v2	0.03	0.43	0.09	0.09	0.07	0.98	0.64		0.59	0.41	0.30	0.38	0.25	0.30	0.11
PPGN v3	0.09	0.84	0.26	0.02	0.17	0.59	0.29	0.59		0.12	0.11	0.13	0.08	0.10	0.04
LNGN	0.00	0.04	0.00	0.13	0.00	0.36	0.78	0.41	0.12		0.56	0.76	0.47	0.56	0.18
GSN-e	0.00	0.06	0.01	0.46	0.01	0.27	0.52	0.30	0.11	0.56		0.79	1.00	0.95	0.43
GSN-v	0.00	0.06	0.00	0.28	0.01	0.34	0.67	0.38	0.13	0.76	0.79		0.76	0.82	0.29
CIN	0.00	0.03	0.00	0.41	0.00	0.22	0.47	0.25	0.08	0.47	1.00	0.76		0.94	0.40
DSS	0.00	0.04	0.00	0.39	0.00	0.26	0.53	0.30	0.10	0.56	0.95	0.82	0.94		0.38
SPEN	0.00	0.02	0.01	0.86	0.00	0.10	0.19	0.11	0.04	0.18	0.43	0.29	0.40	0.38	

Table D.3: Statistical significant analysis of results on the PROTEINS dataset given as p-values. The null hypothesis is that each model produces the same accuracy. A p-value of less than 0.05 is required to reject this null hypothesis and thus conclude that two models produce different accuracy's.

	GDCNN	PSCN	DCNN	DGK	IGN	GIN	PPGN v1	PPGN v2	PPGN v3	LNGN	GSN-e	GSN-v	SIN	CIN	DSS	SPEN
GDCNN		0.56	0.00	0.55	0.55	0.47	0.29	0.59	0.52	0.00	0.51	0.59	0.39	0.31	0.48	0.52
PSCN	0.56		0.00	0.41	0.42	0.33	0.21	0.44	0.40	0.00	0.38	0.82	0.28	0.22	0.35	0.88
DCNN	0.00	0.00		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DGK	0.55	0.41	0.00		0.62	0.59	0.34	0.67	0.59	0.00	0.58	0.51	0.48	0.37	0.55	0.40
IGN	0.55	0.42	0.00	0.62		0.84	0.80	0.93	0.97	0.02	1.00	0.41	0.96	0.86	1.00	0.39
GIN	0.47	0.33	0.00	0.59	0.84		0.57	0.91	0.80	0.00	0.83	0.39	0.83	0.63	0.82	0.31
PPGN v1	0.29	0.21	0.00	0.34	0.80	0.57		0.72	0.83	0.00	0.79	0.25	0.71	0.92	0.78	0.20
PPGN v2	0.59	0.44	0.00	0.67	0.93	0.91	0.72		0.90	0.02	0.93	0.43	0.96	0.78	0.93	0.41
PPGN v3	0.52	0.40	0.00	0.59	0.97	0.80	0.83	0.90		0.02	0.97	0.39	0.92	0.89	0.97	0.37
LNGN	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.02	0.02		0.01	0.10	0.00	0.00	0.01	0.01
GSN-e	0.51	0.38	0.00	0.58	1.00	0.83	0.79	0.93	0.97	0.01		0.38	0.96	0.85	1.00	0.35
GSN-v	0.59	0.82	0.00	0.51	0.41	0.39	0.25	0.43	0.39	0.10	0.38		0.34	0.27	0.36	0.92
SIN	0.39	0.28	0.00	0.48	0.96	0.83	0.71	0.96	0.92	0.00	0.96	0.34		0.78	0.96	0.26
CIN	0.31	0.22	0.00	0.37	0.86	0.63	0.92	0.78	0.89	0.00	0.85	0.27	0.78		0.84	0.21
DSS	0.48	0.35	0.00	0.55	1.00	0.82	0.78	0.93	0.97	0.01	1.00	0.36	0.96	0.84		0.32
SPEN	0.52	0.88	0.00	0.40	0.39	0.31	0.20	0.41	0.37	0.01	0.35	0.92	0.26	0.21	0.32	

Table D.4: Statistical significant analysis of results on the NCI1 dataset given as p-values. The null hypothesis is that each model produces the same accuracy. A p-value of less than 0.05 is required to reject this null hypothesis and thus conclude that two models produce different accuracy's.

	GDCNN	PSCN	DCNN	DGK	CCN	IGN	GIN	PPGN v1	PPGN v2	PPGN v3	LNGN	GSN-e	GSN-v	SIN	CIN	DSS	SPEN
GDCNN		0.01	0.00	0.00	0.18	0.91	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
PSCN	0.01		0.00	0.00	1.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DCNN	0.00	0.00		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DGK	0.00	0.00	0.00		0.01	0.00	0.00	0.00	0.22	0.29	0.00	0.07	0.07	0.05	0.04	0.00	0.00
CCN	0.18	1.00	0.00	0.01		0.22	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
IGN	0.91	0.07	0.00	0.00	0.22		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GIN	0.00	0.00	0.00	0.00	0.00	0.00		0.45	0.10	0.05	1.00	0.64	0.64	0.93	0.55	0.23	0.18
PPGN v1	0.00	0.00	0.00	0.00	0.00	0.00	0.45		0.02	0.01	0.39	0.86	0.86	0.73	0.78	0.55	0.41
PPGN v2	0.00	0.00	0.00	0.22	0.00	0.00	0.10	0.02		0.83	0.08	0.20	0.20	0.22	0.14	0.01	0.01
PPGN v3	0.00	0.00	0.00	0.29	0.01	0.00	0.05	0.01	0.83		0.04	0.17	0.17	0.17	0.11	0.00	0.00
LNGN	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.39	0.08	0.04		0.64	0.64	0.93	0.54	0.17	0.14
GSN-e	0.00	0.00	0.00	0.07	0.00	0.00	0.64	0.86	0.20	0.17	0.64		1.00	0.72	0.96	1.00	0.91
GSN-v	0.00	0.00	0.00	0.07	0.00	0.00	0.64	0.86	0.20	0.17	0.64	1.00		0.72	0.96	1.00	0.91
SIN	0.00	0.00	0.00	0.05	0.00	0.00	0.93	0.73	0.22	0.17	0.93	0.72	0.72		0.65	0.55	0.46
CIN	0.00	0.00	0.00	0.04	0.00	0.00	0.55	0.78	0.14	0.11	0.54	0.96	0.96	0.65		0.94	0.95
DSS	0.00	0.00	0.00	0.00	0.00	0.00	0.23	0.55	0.01	0.00	0.17	1.00	1.00	0.55	0.94		0.74
SPEN	0.00	0.00	0.00	0.00	0.00	0.00	0.18	0.41	0.01	0.00	0.14	0.91	0.91	0.46	0.95	0.74	

Table D.5: Statistical significant analysis of results on the NCI109 dataset given as p-values. The null hypothesis is that each model produces the same accuracy. A p-value of less than 0.05 is required to reject this null hypothesis and thus conclude that two models produce different accuracy's.

	DGK	CCN	IGN	PPGN v1	PPGN v2	PPGN v3	LNGN	CIN	DSS	SPEN
DGK		0.00	0.00	0.03	0.01	0.00	0.00	0.00	0.00	0.00
CCN	0.00		0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00
IGN	0.00	0.04		0.00	0.00	0.00	0.00	0.00	0.00	0.00
PPGN v1	0.03	0.00	0.00		1.00	0.60	0.17	0.01	0.38	0.04
PPGN v2	0.01	0.00	0.00	1.00		0.52	0.12	0.00	0.30	0.01
PPGN v3	0.00	0.00	0.00	0.60	0.52		0.30	0.02	0.66	0.05
LNGN	0.00	0.00	0.00	0.17	0.12	0.30		0.22	0.53	0.58
CIN	0.00	0.00	0.00	0.01	0.00	0.02	0.22		0.05	0.36
DSS	0.00	0.00	0.00	0.38	0.30	0.66	0.53	0.05		0.17
SPEN	0.00	0.00	0.00	0.04	0.01	0.05	0.58	0.36	0.17	

Table D.6: Statistical significant analysis of results on the IMDB-B dataset given as p-values. The null hypothesis is that each model produces the same accuracy. A p-value of less than 0.05 is required to reject this null hypothesis and thus conclude that two models produce different accuracy's.

	GDCNN	PSCN	DCNN	DGK	IGN	GIN	PPGN v1	PPGN v2	PPGN v3	LNGN	GSN-e	GSN-v	SIN	CIN	DSS	SPEN
GDCNN		0.23	0.00	0.00	0.28	0.01	0.13	0.15	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00
PSCN	0.23		0.00	0.00	0.61	0.04	0.37	0.45	0.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DCNN	0.00	0.00		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DGK	0.00	0.00	0.00		0.02	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
IGN	0.28	0.61	0.00	0.02		0.21	0.80	0.93	0.70	0.16	0.01	0.02	0.09	0.11	0.06	0.13
GIN	0.01	0.04	0.00	0.00	0.21		0.28	0.19	0.40	0.87	0.18	0.35	0.80	0.80	0.55	0.96
PPGN v1	0.13	0.37	0.00	0.01	0.80	0.28		0.85	0.87	0.21	0.01	0.03	0.13	0.14	0.07	0.18
PPGN v2	0.15	0.45	0.00	0.00	0.93	0.19	0.85		0.73	0.11	0.00	0.01	0.06	0.07	0.03	0.09
PPGN v3	0.14	0.33	0.00	0.01	0.70	0.40	0.87	0.73		0.37	0.04	0.08	0.23	0.25	0.15	0.31
LNGN	0.00	0.00	0.00	0.00	0.16	0.87	0.21	0.11	0.37		0.03	0.04	0.51	0.56	0.27	0.74
GSN-e	0.00	0.00	0.00	0.00	0.01	0.18	0.01	0.00	0.04	0.03		0.43	0.15	0.18	0.34	0.09
GSN-v	0.00	0.00	0.00	0.00	0.02	0.35	0.03	0.01	0.08	0.04	0.43		0.33	0.38	0.71	0.19
SIN	0.00	0.00	0.00	0.00	0.09	0.80	0.13	0.06	0.23	0.51	0.15	0.33		1.00	0.65	0.78
CIN	0.00	0.00	0.00	0.00	0.11	0.80	0.14	0.07	0.25	0.56	0.18	0.38	1.00		0.67	0.80
DSS	0.00	0.00	0.00	0.00	0.06	0.55	0.07	0.03	0.15	0.27	0.34	0.71	0.65	0.67		0.47
SPEN	0.00	0.00	0.00	0.00	0.13	0.96	0.18	0.09	0.31	0.74	0.09	0.19	0.78	0.80	0.47	

Table D.7: Statistical significant analysis of results on the IMDB-M dataset given as p-values. The null hypothesis is that each model produces the same accuracy. A p-value of less than 0.05 is required to reject this null hypothesis and thus conclude that two models produce different accuracy's.

	GDCNN	PSCN	DCNN	DGK	IGN	GIN	PPGNv1	PPGNv2	PPGNv3	LNGN	GSN-e	GSN-v	SIN	CIN	DSS	SPEN
GDCNN		0.02	0.00	0.00	0.44	0.00	0.06	0.25	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.22
PSCN	0.02		0.00	0.46	0.02	0.00	0.00	0.85	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
DCNN	0.00	0.00		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DGK	0.00	0.46	0.00		0.00	0.00	0.00	0.94	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
IGN	0.44	0.02	0.00	0.00		0.02	0.39	0.17	0.27	0.07	0.00	0.02	0.02	0.01	0.01	1.00
GIN	0.00	0.00	0.00	0.00	0.02		0.10	0.02	0.23	0.41	0.16	0.84	0.88	0.77	0.53	0.00
PPGNv1	0.06	0.00	0.00	0.00	0.39	0.10		0.07	0.75	0.33	0.01	0.11	0.09	0.07	0.03	0.29
PPGNv2	0.25	0.85	0.00	0.94	0.17	0.02	0.07		0.06	0.03	0.00	0.01	0.01	0.01	0.01	0.15
PPGNv3	0.04	0.00	0.00	0.00	0.27	0.23	0.75	0.06		0.57	0.02	0.21	0.19	0.16	0.09	0.19
LNGN	0.00	0.00	0.00	0.00	0.07	0.41	0.33	0.03	0.57		0.04	0.36	0.34	0.28	0.15	0.02
GSN-e	0.00	0.00	0.00	0.00	0.00	0.16	0.01	0.00	0.02	0.04		0.29	0.22	0.28	0.39	0.00
GSN-v	0.00	0.00	0.00	0.00	0.02	0.84	0.11	0.01	0.21	0.36	0.29		0.95	0.95	0.73	0.01
SIN	0.00	0.00	0.00	0.00	0.02	0.88	0.09	0.01	0.19	0.34	0.22	0.95		0.89	0.65	0.00
CIN	0.00	0.00	0.00	0.00	0.01	0.77	0.07	0.01	0.16	0.28	0.28	0.95	0.89		0.77	0.00
DSS	0.00	0.00	0.00	0.00	0.01	0.53	0.03	0.01	0.09	0.15	0.39	0.73	0.65	0.77		0.00
SPEN	0.22	0.01	0.00	0.00	1.00	0.00	0.29	0.15	0.19	0.02	0.00	0.01	0.00	0.00	0.00	