Developing Machine Learning Architectures Purpose-Built for Condensed Matter Physics

George Bird george.bird@student.manchester.ac.uk
Supervisor: Dr. Mohammad Saeed Bahramy

Department of Physics and Astronomy
University of Manchester

March 2023

Abstract

Machine learning has already been shown to be an effective tool in modelling the properties of crystalline systems. However, most approaches to date can be considered as a repurposing of computer vision architectures. We posit that a form of neural network which is purpose-built for condensed matter physics, by considering and leveraging important symmetries, maybe even more effective than these prior approaches.

To achieve this, a ground-up review of deep learning methods is undertaken, with a substantial emphasis on the symmetry behaviours of common functions. Features which are advantageous to condensed matter modelling are highlighted. In this, several deficits of existing models are defined, with proven resolutions.

The project concludes with a machine learning model which can be applied universally to all crystalline systems and is able to predict a wide repertoire of properties, by making use of the reciprocal-space. Models which analyse the orbital overlaps of the crystal Bismuth-Telluride-Iodide are also conducted, and shown to have some success in predicting emergent quantum states such as topological insulators. Some of the proposed models, and newly identified problems, may also have a wider impact on other applications of machine learning.

Contents

1 Introduction

1.1 Project Overview and Prior Work

Both machine learning and condensed matter physics are considered among the most exciting and fruitful avenues of modern scientific research. In this thesis, the considerable overlap between these domains will be made clear. A novel interdisciplinary approach to developing new materials, defined by emergent quantum behaviours, will be demonstrated through custom-designed machine learning architectures. The term "architecture" represents the wiring pattern and overall structure of the network, independent of its training to perform a particular task. The grand aim is to develop strong computational models which can quickly predict the properties of untested materials derived from the understanding of other crystalline systems. The goal is to streamline the search for promising materials, which can then be verified experimentally. Moreover, a purpose-built machine learning approach to evaluating these systems have the unique potential to embed desirable properties into current crystal configurations [1], enabling a host of new applications in spintronics [2,3], quantum computing [3], energy storage [4], and a greater understanding of quantum field theories through exotic quasiparticles [5,6].

In the previous semester [7], analytic techniques were developed for the crystal Bismuth-Telluride-Iodide (BiTeI) to determine its state under particular mechanical distortions. This crystal was chosen as it exhibits a range of well-understood emergent quantum phenomena under various conditions, including trivial and topological insulating states alongside Weyl and Dirac semimetallic states. The Weyl-semimetal and topological insulator states result from strong spin-orbit couplings which break spatial-inversion symmetry, through the Rashba effect [8]. This causes spin-splitting of the otherwise degenerate electronic band structure, allowing an intermediate Weyl semimetal state to exist separating the trivial and topological insulating states in the phase space. A rearrangement of the band structure, known as band inversion, occurs in the Weyl-semimetal state enabling the $\mathbb{Z}_2$ topological invariant to change which distinguishes the trivial and topological insulating states. It was demonstrated that varying hydrostatic pressure is sufficient for BiTeI to display these emergent quantum states. A novel algorithm was then developed to determine the corresponding state from interpolated hoppings between neighbouring atoms in the lattice.

Consequently, BiTeI offered an ideal system to benchmark and compare various computational approaches to understanding and predicting the presence of these material properties. Hence, enabling a particular model's effectiveness to be determined on a well-understood system before applying it to new systems. Additionally, the pairing of atomic orbital hoppings to a resultant state enables the use of supervised machine-learning techniques. The prior work focussed on how this dataset could be produced efficiently, whereas this thesis will explore the development of novel machine-learning architectures. The intent is to develop purpose-built models which leverage crystal properties for improved accuracy in classifying states of BiTeI. Despite an analytical algorithm already existing, the aim is to generalise these architectures to allow knowledge gained on one crystalline system to be applied to another, alongside offering alternative insights into the nature and emergence of the states.

1.2 The Potential of Machine Learning for Condensed Matter Applications

Neural networks, or deep-learning, is a subset of machine learning algorithms which are widely applicable to many circumstances. Particularly, they are advantageous for problems with a large number of degrees of freedom which interact non-linearly and cannot be easily visualised or interpreted by simpler methods or humans. Predicting emergent quantum phenomena in crystalline systems falls into this category, making deep-learning a promising tool for understanding these phenomena. Moreover, if beneficial, the operations may become hermitian or unitary allowing them to closely model the quantum dynamics within the material. The exact band-structure make-up of the various material states may also be highly variable, and deep-learning's adeptness at generalising classifications is also strongly applicable.

Deep-learning models consist of stacked and optimisable linear algebra operations sandwiched between non-linear activation functions. The particular linear algebra operations and activation functions used are defined by its architecture, with the particular structure having a large impact on the success of the model on its respective task. These operations include tunable parameters (ϱ) which are optimised using a process called backpropagation and gradient-descent, the latter shown in *Eqn.* 1, to achieve increasingly better performance. Free parameters, or activations, represent the information to be processed by the model and are an encoded representation of important aspects of the crystalline system.

$$\varrho_{i+1} = \varrho_i - \eta \frac{\partial C}{\partial \varrho_i} \tag{1}$$

Where C is an error function to be minimised, known as cost, with $0 < \eta \ll 1$ being a small step, known as the learning rate.

Machine learning has already had significant in-roads in analogous search domains, such as drug discovery [9], and numerous adoptions of the technology have already been demonstrated for condensed matter systems [10–15]. However, the latter primarily feature adapted architectures from computer vision [10–13] and natural language processing [15]. These

models are not structured to exploit the features of condensed matter systems but rather their respective domains, which may result in poorer predictive performance when they are applied to crystalline systems. Often these crystalline systems are reformulated to superficially appear as a computer vision or natural-language problem, yet do not share the respective relations and symmetries that these models leverage. Additionally, the reformulation process may obfuscate important features of the system. Therefore, it is preferable to construct a custom machine-learning model which accepts the condensed matter parameters in their natural form alongside using an architecture designed to accelerate learning and boost accuracy by utilising the characteristics of the materials. This ground-up approach appears absent in the literature, so offers an exciting avenue to create more potent computer models.

A particular emphasis on generating a large, varied dataset, using the previously defined techniques, is essential such that these models learn the underlying physics as opposed to shortcutting the problems through simpler means. To this end, adding constrained physical noise to the dataset, and how it in turn affects the state classification, is explored in this project. This noise encourages the model to correctly approximate the physical behaviour, by penalising the unphysical shortcut approaches since they do not generalise well. It is feasible that the model will still find unexpected shortcuts, but analysis of these may instead reveal hidden physical insights. Further, only a model which has learnt the underlying physics has the capacity to reverse engineer desirable properties into a material in a physical manner.

Two primary routes of constructing condensed matter neural networks will be explored: one operating on the atomic hoppings derived from density functional theory, shown in *Eqn.* 2, and the other utilising the symmetries of the Brillouin zone, shown in *Eqn.* 3. To guarantee the symmetry leveraging characteristics of the models, a series of tests will be performed on common machine learning datasets, MNIST [16] and CIFAR10 [17], which are perturbed in equivalent symmetric ways to the BiTeI crystal. Thus, performance in these more-illustrative domains maps directly to the condensed matter applications and enables wider applicability of the architectures.

$$\hat{H}_{ij\vec{A}} = \left\langle \psi_i \left| \hat{H} \right| \psi_j \right\rangle = \int \mathrm{d}\vec{x} \psi_i^* \left(\vec{x}\right) \hat{H} \left(\vec{x}\right) \psi_j \left(\vec{x} - \sum_k^d A_k \vec{r_k} \right) \tag{2}$$

With $\hat{H}_{ijA}$ being the complex wavefunction overlap termed "hoppings", determined using density functional theory. Whilst $\vec{r_k}$ are the lattice vectors for a d dimensional crystal with A_k giving a particular number of cell offsets which the orbital overlap, between wavefunctions ψ, is evaluated for. For the BiTeI system studied, A takes 1155 unique combinations, with 18 orbitals considered for all permutations of Bismuth ($6P_{\{x,y,z\}\{\uparrow,\downarrow\}}$), Tellurium ($5P_{\{x,y,z\}\{\uparrow,\downarrow\}}$) and Iodide ($5P_{\{x,y,z\}\{\uparrow,\downarrow\}}$). All of these hopping terms are used to evaluate the material state, however, as later discussed, only a truncated number is provided to the network to mitigate the network shortcutting the problem.

$$\hat{H}_{ij}\left(\vec{k}\right) = \mathcal{F}_{\vec{k}}\left[\hat{H}_{ij\vec{A}}\right] = \sum_{\forall \vec{A}} e^{i\vec{k}\cdot\left(\sum_k^d A_k \vec{r_k}\right)} \hat{H}_{ij\vec{A}} \tag{3}$$

The reciprocal space, or crystal-momentum ($\vec{k}$) space, is a Fourier transform of the hoppings and allows for a model universality to crystalline systems. Unlike hoppings, which consist of an arbitrarily chosen set of orbital overlaps, the electronic behaviour of a crystal can be interpreted through a bounded unique region known as the Brillouin zone, which is then tessellated to form the full reciprocal space. The hoppings can be thought to add corrections to this space, where more distant neighbours (large $\left|\vec{A}\right|$) typically result in smaller corrections. Therefore, the Brillouin zone generalises across all crystals and characterises their state and a sampling of this zone can be provided to machine learning models. This shared feature allows a model to transfer knowledge between crystals, for greater insight. Therefore, it is preferable for a model to operate on the Brillouin zone.

2 Background and Methodology

2.1 Symmetries of Brillouin Zones

Crystalline systems can exhibit many characteristic symmetries in their respective Brillouin zones. Primarily, discrete translational, n-fold rotational, spatial inversion and time-reversal symmetries will be focussed on, as these either lead to the emergent quantum phenomena of interest or constitute important considerations in neural network design.

The custom models will be formulated from the same linear-algebra building blocks as most neural networks, such as affine transformations, their subset known as discrete convolutions, dot-products, and various other tensor operations. Each of these operations has a particular performance optimality when operating on another tensor with a certain representational symmetry. Representational symmetries are defined as symmetries in the arrangement of elements in the tensor or their particular values. However, often in physical symmetries, it is not the value of the particular elements nor how they are arranged which indicates the symmetry, it is instead how the tensor as a whole transforms, such as invariance under a specified operation. In effect, these physical symmetries are found in linear combinations of the tensor's basis rather than

being aligned with just a single element of the basis. This important mismatch can be detrimental when applying common computer vision architectures to physical systems.

Therefore, the primary challenge is to transfer these physical symmetries into representational symmetries, such that the building blocks each function optimally as intended. For example, convolution layers, a subset of affine transforms defined by a sparse-matrix multiplication with repeated localised clusters of tunable parameters, likewise require a repeated local structure in surrounding elements, at a particular scale, to be present across the tensor for it to perform optimally. It is this representational symmetry which n-fold rotational symmetry can be converted into.

2.1.1 Novel Rotational Symmetry Encoding

The global n-fold rotational symmetry is most apparent in the reciprocal space, where the space is invariant under discrete $\frac{2\pi}{n}$radian rotations of the space about a particular axis, shown in *Eqn.* 4. The discrete convolution operation requires a repeating structure displaced across the image, however, for optimality, this structure should remain in a similar orientation throughout the tensor, whereas the discrete rotational symmetry results in the rotated repeated structure[1].

$$\hat{H}_{ij}\left(\vec{k}\right) = \hat{H}_{ij}\left(e^{i\frac{2\pi k}{n}\hat{m}\cdot\vec{\mathbf{X}}}\vec{k}\right) \tag{4}$$

With $\vec{\mathbf{X}}$ being the special-orthogonal Lie-group generators, $\hat{m}$ the rotation axis and $k \in \mathbb{Z}$.

There are two methods which may leverage this symmetry when using discrete convolution, both are largely equivalent. The two-dimensional discrete convolution [18] is illustrated in *Fig.* 1 and denoted as $f_{\mathbf{K}}$ for a particular tunable kernel $\mathbf{K}$.

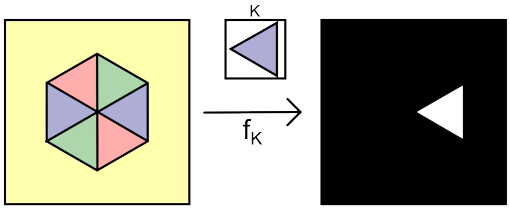

Figure 1: Depiction of how a discrete convolution might emphasise a region which has similarity to the kernel in their representations. The region represents a sampled two-dimensional slice of the Brillouin zone with 2-fold rotational symmetry. However, the convolution has not highlighted the similar rotated structure. This is suboptimal as the kernel cannot be reused; instead, two separate kernels would be required. This slows the network's learning and produces poorer performance from a lack of generalisation.

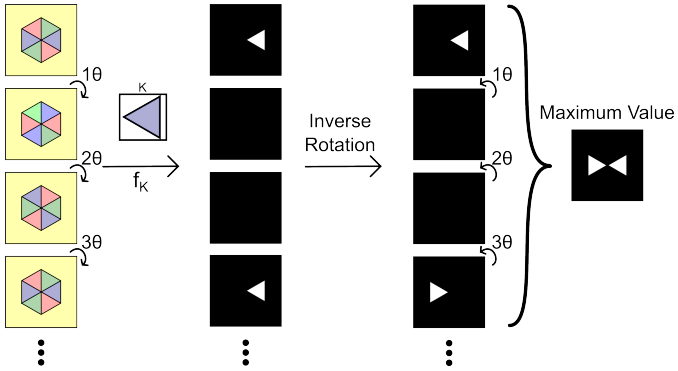

Figure 2: Illustrated is the algorithm for transferring rotational symmetries into translational symmetries, such that the problem is optimal for discrete convolution. Here $\theta = \frac{\pi}{3}$ for clarity, however, it should be considerably smaller in cases where the discrete rotation is present but for an unknown rotation.

The symmetry can then be made representational by adding additional dimensions to the input or the kernel, commonly referred to as adding new channels. These channels are created by concatenating the rotated representation, or appearance, of the kernel by interpolating bilinearly. The rotation angle is θ, defined by $(\theta \ll 2\pi) \cap \left(\frac{2\pi}{\theta} \in \mathbb{N}\right)$. Equivalently, the input tensor can be rotated instead, making the operation a (d+1)-dimensional convolution as demonstrated in *Fig.* 2. Then convolution can proceed as normal, with the expanded input channels or stacked kernels. The stacked dimension of the resultant tensor should then be inversely rotated by the respective angle. Finally, the maximum value should be taken across this tensor

[1] Unless it also has a corresponding local rotational invariance

axis. It may be desirable to add a depth encoding, to indicate how rotated the similar feature is, analogous to the positional encodings of a transformer [19].

Overall this series of operations make any rotational symmetry more interpretable to the machine-learning algorithm by transferring it into a translational symmetry. Therefore, the network does not need to generate approximate rotated copies of each kernel which would otherwise slow down learning and take priority over other important unique kernels.

2.1.2 Hermitian Symmetry

In the reciprocal space, the Hermitian symmetry ($\hat{H}_{ij} = \hat{H}^*_{ji}$), which ensures physical quantum measurement values due to real eigenvalues, exists already in a representational form due to the transpose. Therefore, no adjustments need to be considered. However, in the real-space hoppings, the hermitian symmetry can be detrimental to performance.

This is because the real-space hoppings follow the relation in *Eqn.* 5. These complex values are equivalent to two real-valued numbers which are the preferred input for deep-learning algorithms. There is a choice in representation between $re^{i\theta}$ and $a + bi$ with $a, b, r \in \mathbb{R}$ and $\theta \in [0, 2\pi)$. The modularity in $\theta \in (\mathbb{R} \mod 2\pi)$ would have to be applied throughout the network with varying modulus and this would be impractical.

$$\hat{H}_{ij,\vec{A}} = \hat{H}^*_{ji,-\vec{A}} \tag{5}$$

Alternatively, the real and imaginary components can be used. However, any convolution along axis $\vec{A}$ followed by a global operation also along $\vec{A}$ results in a failure when tuning the convolution kernel. This is because the Hermitian symmetry results in the imaginary components forming pairs which sum to zero, this results in parameter updates, shown in *Eqn.* 1, becoming zero and slowing, or even halting, learning.

This risk is easily mitigated by cropping the input to exclude the duplicated information present due to hermitian symmetry. In effect, if hoppings from the $\vec{A}$ neighbouring cell are included then those from $-\vec{A}$ should not be included. Removing this redundant data is highly beneficial in multiple ways. The smaller input requires less memory, less tunable parameters or calculations and therefore faster network training. In addition, the reduction in redundant parameters may encourage the network to pursue a physical understanding of the system, as opposed to an overfitted, shortcutted understanding. This should improve the generality and performance of the model.

2.1.3 Time-Reversal and Spatial-Inversion

The presence of time-reversal or spatial-inversion symmetries, shown in *Eqns.* 6 and 7 respectively [20], characterise the type of topological nature of the material when under the appropriate distortion. The properties of these symmetries are explained, and their implications are discussed, in the prior project [7]. It is these symmetries which the machine learning algorithm must model to correctly classify the material's state.

$$H\left(\vec{k}\right) = \sigma_y H\left(-\vec{k}\right)^T \sigma_y \tag{6}$$

$$H\left(\vec{k}\right) = P H\left(-\vec{k}\right) P^{-1} \tag{7}$$

With σ_y being the second Pauli matrix and $P = \mathrm{diag}\left(\pm 1, \cdots, \pm 1\right)$. Since these symmetries can be pivotal for characterising the state, such as separating Dirac and Weyl semimetal states, the neural network model will be required to interpret these physical symmetries regardless, to achieve good performance. Despite being physical symmetries, they also result in appreciable differences in the representation of the Brillouin zone by relating regions of $\vec{k}$ to $-\vec{k}$. Therefore, no further steps need to be taken to ensure the model can analyse these symmetries optimally.

Further, creating perturbations respecting the time-reversal symmetry and breaking spatial-inversion symmetry scaling with an external parameter λ, enables more varied data to be generated to train the neural network. This has a two-fold advantage. The first is that a larger number of Weyl semimetal states are generated for the dataset, as the gapless region grows with larger $|\lambda|$ [21], therefore, there is a great likelihood that the model will learn the identifying markers of this state. Secondly, dataset expansion prevents the model from overfitting to a simpler task [22], which would be a high-dimensional embedding of a linear regression model. Instead, it must learn to approximate the true physical behaviour of the system to obtain better accuracy on this, more varied, dataset.

The time-reversal symmetric perturbations can be generated for the real-space hoppings using a series of operations on an already hermitian array, denoted $B_{ij\vec{A}}$, as described in *Sec.* 2.1.2. These operations are described in *Eqns.* 8 to 11.

$$\Delta\hat{H}_{ij\vec{A},\uparrow\downarrow} = \frac{1}{2}\left(B_{ij\vec{A},\uparrow\downarrow} + B^T_{ij\vec{A},\uparrow\downarrow}\right) \tag{8}$$

Where $\Delta\hat{H}_{ij\vec{A},\uparrow\downarrow}$ denotes the overlap between the central i^{th} orbital spin-up state to the $\vec{A}$ displaced j^{th} orbital spin-down state. This sub-matrix of matrix $\Delta\hat{H}_{ij\vec{A}}$, describes the perturbation, and this sub-matrix is symmetrised.

$$\Delta\hat{H}_{ij\vec{A},\downarrow\uparrow} = \Delta\hat{H}^{\dagger}_{ij\vec{A},\uparrow\downarrow} \tag{9}$$

Here, *Eqn.* 9 describes copying the transpose-conjugate of $\hat{H}_{ij\vec{A},\uparrow\downarrow}$ to $\hat{H}_{ij\vec{A},\uparrow\downarrow}$. Then *Eqn.* 10 copies the conjugate-transpose hoppings to neighbouring cell $\vec{A}$ to those at $-\vec{A}$.

$$\Delta\hat{H}_{ij-\vec{A}} = \Delta\hat{H}^{\dagger}_{ij\vec{A}} \tag{10}$$

Finally, the overlap of the spin-up states with spin-up states is made equal to those of the spin-down interaction with other spin-down states in *Eqn.* 11.

$$\Delta\hat{H}_{ij\vec{A},\downarrow\downarrow} = \Delta\hat{H}^{*}_{ij\vec{A},\uparrow\uparrow} \tag{11}$$

The matrix $\Delta\hat{H}$ should then be normalised by dividing through by the root-mean-squared of all the elements and then multiplying by the external parameter λ. This creates noise, of varying strengths, in the material's atomic-orbital overlaps. These perturbations correspond to various mechanical distortions which can be performed on the crystal and are able to expand the dataset such that the model is encouraged to generalise its understanding to a wider variety of crystalline material environments.

2.1.4 Discrete Translational Symmetry

A pure real-space crystal is characterised by a periodicity in integer multiples, $\vec{A}$, of the lattice vectors $\vec{r}$ as shown in *Eqn.* 2. For deep-learning models which operate on these hopping coefficients, this periodicity may be leveraged to reduce the required number of tunable parameters.

This results in an assumption that like-atomic orbitals may have similar values, relative to the other orbital hoppings, across the varying displacements of $\vec{A}$ cells. For example, we may expect the hopping of Bismuth $P_{z\uparrow}$ to Tellurium $P_{z\downarrow}$ to be more prominent than Iodine $P_{z\downarrow}$ to Iodine $P_{x\uparrow}$ when the overlap is over any number of neighbouring cells or value of $\vec{A}$. It is highly likely the overall magnitude of the interaction varies across $\vec{A}$ but there may be some consistency in the relative overlaps over a constant distance $\sum_i A_i\vec{r_i}$. This periodic behaviour in $\vec{A}$ may allow a reduction in tunable kernel parameters as a single kernel can be convolved across all $\vec{A}$ instead of having dedicated parameters for each. This architecture is developed further in *Sec.* 2.4.1.

However, analogous considerations in the reciprocal-space are much more significant; allowing for a universal architecture across many crystalline materials. As previously stated, the Brillouin zone is a resultant finite region, common to all crystals, and defines the particular crystal's electronic behaviour. Making a consistent representation across all possible Brillouin zones enables a neural network to operate upon any given Brillouin zone using the same model. This allows for learnt behaviours from one crystalline system to be reapplied to another. This possibility allows condensed matter models to be applied and trained on a much wider domain, increasing their potential.

The discrete translational symmetry results in corresponding reciprocal space lattice vectors, $\vec{b_i}$, defined by $\vec{b_i}\vec{r_j} = 2\pi\delta_{ij}$. These can be used as a consistent way to span the reciprocal-space unit-cell which encompasses the full Brillouin zone for any crystal. For one-dimensional or two-dimensional lattices, $\vec{b}_{2,3} = \vec{0}$ can be used to ensure they are also interpretable by the network. Using the unit-cell approach ensures that no matter the shape of the Brillouin zone, such as a hexagonal prism or cubic, it remains a consistent representation of the network by being contained within the larger parallelepiped cell.

To preserve all information, the conversion from the real-space to reciprocal-space must be injective, which requires n sampling along each reciprocal lattice vector, respecting $n^3 \geq i * j * m$, where m counts all unique combinations of $\vec{A}$. The sampling can then be performed through a linear combination of the reciprocal space vectors, whilst the discrete translational symmetry can restrict the sampling region to be $a_{1,2,3} \in [-0.5, 0.5]$ for $\sum_i a_i\vec{b_i}$. This also sets the centre of the Brillouin zone, $\vec{k} = \vec{0}$, to be the centre in the representation. The discretely sampled reciprocal unit-cell, $H_{a_1a_2a_3ij}$, is shown in *Eqn.* 12, for $a_k \in (\mathbb{Z} \cap [0, n-1])$.

$$\hat{H}_{a_1a_2a_3ij} = H_{ij}\left(\sum_{k=1}^{3}\frac{(2a_k - n + 1)\,\vec{b_k}}{2n-2}\right) \tag{12}$$

Finally, a modification of discrete convolution is discussed in *Sec.* 2.6, which acts to fold opposite edges of the discretely sampled cell, such that the periodicity is represented, in an effective discrete 3-torus space [23]. Whilst the edges or the parallelepiped cell are mapped to the edges of the representation. However, the sampling in *Eqn.* 12 causes a resultant overcounting of the edge samples when folded into a torus, thus a correction is given in *Eqn.* 13.

$$\hat{H}_{a_1a_2a_3ij} = H_{ij}\left(\sum_{k=1}^{3}\frac{(2a_k - n)\,\vec{b_k}}{2n}\right) \tag{13}$$

2.2 Novel Crystalline Material Optimising Through Gradient Descent

Before considering applications of deep-learning to classifying the BiTeI state, an approach of adapting the underlying crystal to exhibit desirable properties is possible through gradient descent. This methodology is applicable to the prior semester's analytical approach, but can also be generalised to systems modelled using deep-learning.

Since the prior analytical approach to calculating tight-binding band structure [7] involved fully differentiable operations, it is possible to perform the gradient descent algorithm, shown in *Eqn.* 1. This allows for the electronic band-structure to be tweaked to a desirable form, and small updates will alter to the hopping coefficients until the system of interest matches the modified band structure. If the algorithm for density functional theory [24] is also made differentiable, this approach can be also used to modify the underlying unit-cell.

However, this has the possibility of producing unphysical results, whereas only modifications which can be achieved through mechanical distortion are desired. A simple technique to achieve this would be to take the original relative coordinates of atoms in the basis, then premultiply these with a tunable matrix $T \in \mathbb{R}^{3\times3}$, before passing them to the density-functional-theory algorithm and the tight-binding algorithm. Therefore, by performing gradient descent on T, the crystal can be modified through a physical distortion which results in a close approximation to the desired band structure. Gradient-clipping [25] can also be used for a similar result.

This technique can also be generalised for external effects such as an applied magnetic field and applied to deep-learning classification algorithms for various emergent properties.

2.3 Overview of Fully-Connected Approach

A fully-connected neural network is often considered the simplest deep-learning architecture; it has no particular application and constitutes a series of affine transformations with intermediate non-linear transformations. However, with no express use case, it also does not leverage any symmetries or repeated similarities to optimise its function.

This form of architecture will be used as a control network, to compare with the latter neural networks. Fully connected networks do not constitute one architecture, but instead, a functional class characterised by the number of layers and the number of neurons in each of these layers. These choices are referred to as hyper-parameters or architectural parameters. In *Fig.* 3 two fully-connected networks are depicted of different sizes.

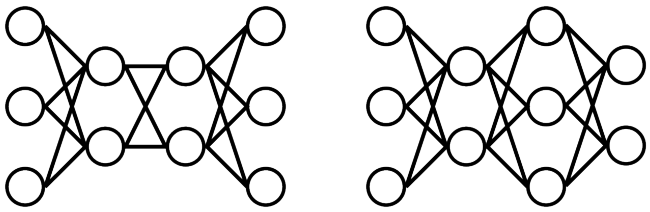

Figure 3: A pictorial representation of two fully connected neural networks, where each node is a neuron, holding activation a, and connection a function $f(x) = \sigma(wa + b)$, where σ is an activation function, and tunable parameters w and b. The particular architectures may be represented as left: $[3, 2, 2, 3]$ and right: $[3, 2, 3, 2]$.

Due to their general scope of application, these networks can be applied to both the real-space $\hat{H}_{ij\vec{A}}$ or sampled reciprocal-space $\hat{H}_{a_1a_2a_3ij}$. For later comparisons, two fully-connected networks for the real-space were constructed. One may be considered a small fully-connected network with structure $[17496, 100, 3]$ and a larger network $[17496, 500, 500, 500, 100, 100, 3]$. Each layer is interspaced with a Leaky-ReLU activation function $f(x) = \max(0.01x, x)$. These are represented in *Fig.* 4.

In this, and the following models, the number of input hoppings is truncated to the range: $\vec{A} \in \{0, 1, 2\}^3$. This encourages the network to take a physical understanding of the system. With a larger input number, there would be insufficient samples for modelling the nature of the decision boundaries, this problem is known as the curse of dimensionality [27, 28]. This is evident when the number of samples is on the order of the number of inputs. In this scenario, an alternative basis can be constructed where each sample is represented in a single dimension with a value of one, otherwise, its absence is indicated by a zero value, known as one-hot encoding [29]. With this basis, the network can perform a very simple task of placing the respective decision boundaries between zero and one, which is not fulfilling our desired task of understanding a crystalline system. Therefore, by reducing the input dimensions, the model is motivated to learn a realistic method of obtaining its classification.

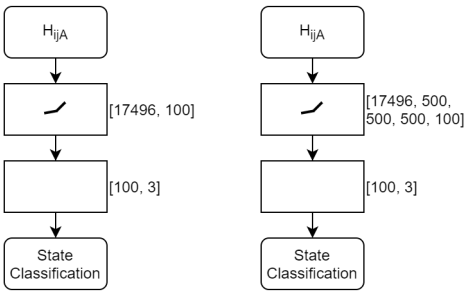

Figure 4: Left diagram depicts the small fully-connected neural network to operate on the real-hoppings, $\hat{H}_{ij\vec{A}}$. Whilst the right diagram shows a much larger network. These networks are drawn using Bird's convention [26], which will similarly be used for all future network diagrams.

2.4 Overview of Convolutional Architectures

Tuned parameters are trained concurrently, however, with each greater degree of freedom the number of local minima is likely to increase. This may slow, or prevent, the desired learning. In addition, if similar repeated structures occur throughout the input, independent parameters will have to converge on the same configuration multiple times. This is unnecessary, increases the chance of incorrect modelling, and may lead to a reduction in model generality.

Convolutional architectures mitigate these risks by reusing tuned parameters to pick up on these repeated local structures in the tensor representation. They consist of a set of n tunable kernels which are iterated across the tensor and perform a dot product with a localised area. This small area is known as the receptive field of the respective neuron. When the local area has a similarity with the specific kernel, then the resultant dot-product is large and positive. Therefore, convolutions are highly efficient when the representation has these properties, such as in computer vision tasks where similar shapes and textures repeatedly occur.

Due to the discrete translational symmetries of the Brillouin zone, repeated structures are also expressed in the representation, so it will be beneficial to employ convolutional architectures for this task. It is also expected that smaller-scale structures may also be repeated throughout the unit-cell or differing crystals, furthering the use for convolution. Likewise, it can be applied to the assumed approximate translational symmetries throughout $\vec{A}$ in the real-space, since it is the resonances in these values which define the resultant reciprocal-space.

Moreover, multiple convolutional operations can be applied sequentially to the data. This results in an increasingly complex conglomerate kernel operating on the results of prior simpler kernels, allowing for the recognition of progressively intricate patterns. Each component kernel can also compensate for one another in the following dot product, therefore generalising the response for a range of complex patterns. This abstraction enables increasingly complex patterns to be grouped under umbrella categories.

Abstraction is particularly important for condensed matter physics, where a range of similar Brillouin zones, across different crystals, can be grouped under an emergent state. This property is reasonably unique to convolution, whereas a fully-connected layer can be thought of as a basis change without necessarily increasing complexity. In addition, a specific hopping may take a range of values, whilst the crystal remains within a particular state. This variation results in a range of corrections to the Brillouin zone's shape, which this sequential convolutional structure offers a generalisation to.

It is expected that the reciprocal-space has a range of defining structures across several complexity scales. This is due to the application of the Fourier transform, which indicates the length scale of patterns. It suggests multiple convolutional layers will be advantageous in solving the problem, and is discussed further in *Sec.* 2.7.

2.4.1 Novel High-Dimensional Hybrid Convolutional Architecture

The real-space hoppings indicate the transition probability from orbital i to j across $\vec{A}$ cell displacements. At any particular constant $\vec{A}$, the respective matrix elements will have a range of possible structures. It is expected that a linear combination of these structures may persist across further values of $\vec{A}$, so there may be some form of repeated structure to leverage.

A novel architecture, using the repeated parameter concept of convolutions to increase performance when operating on real-space hoppings, will be demonstrated. However, unlike convolution, no striding of the input occurs. This is because no repeated structure across different hoppings, for constant $\vec{A}$, is expected to be present since their indexing is arbitrarily assigned when making the matrix. This results in an effective fully-connected network which operates on a matrix $\hat{H}_{ij}$, which is then reused across all $\vec{A}$, like a convolution, defining this hybrid approach.

The architecture is defined by kernels of size $K \in \mathbb{R}^{18\times18\times2}$, which are convolved across dimensions of $\vec{A}$. However, to detect the potentially multiple structures, it is essential that $1 \ll n$ kernels are used. Most importantly, this mitigates bottlenecking [30, 31] in the architecture, defined as important information being lost due to an insufficient number of

subsequent neurons. Despite this, a small reduction of neurons is desirable to remove any redundant information, so n is chosen to satisfy $1 \ll n < 18 \times 18 \times 2$. Therefore, the first operation in this architecture is shown in *Eqn.* 14, using Einstein's summation convention [32] and a tunable bias parameter $b^{(1)} \in \mathbb{R}^{\vec{A} \times n}$. The summation convention is also used in all the following equations.

$$a^{(1)}_{\vec{A}n} = \hat{H}_{ij\vec{A}} K_{ijn} + b^{(1)}_{\vec{A}n} \tag{14}$$

Since repeated structures were only expected across $\vec{A}$, only a fully-connected network should follow, as opposed to a further convolution, as it may be detrimental to assume further common structure. Therefore, the tensor $a^{(1)}_{\vec{A}n}$ is multiplied with a weight tensor $w_{\vec{A}nm}$ and summed with a bias vector $b^{(2)}_m$. The number of neurons m is also chosen from $1 \ll m \ll n$, to prevent bottlenecking whilst reducing redundancy. This is shown in *Eqn.* 15.

$$a^{(2)}_m = a^{(1)}_{\vec{A}n} w_{\vec{A}nm} + b^{(2)}_m \tag{15}$$

Finally, a large further fully-connected network operates on the vector of m neurons, completing the network. This architecture differs significantly from others in the literature, so represents a custom neural network purpose-built for condensed matter applications rather than repurposed from another application. It is symbolically depicted in *Fig.* 5, where $n = 256$ and $m = 1024$ were chosen. However, due to operating on the real-space hoppings, where a set of neighbours $\left\{\vec{A}\right\}$ are chosen specifically for BiTeI, it generalises poorly to other crystalline systems. Therefore, further architectures are designed to operate on the shared unit-cell of the reciprocal-space.

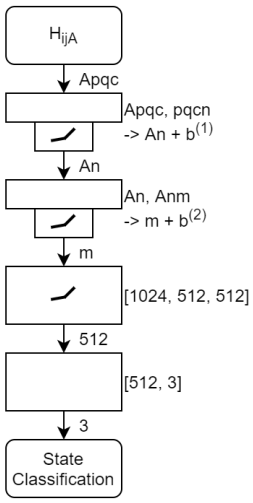

Figure 5: Depicts a novel neural network architecture for predicting material classifications from real-space orbital hopping coefficients. It is purpose-built for BiTeI crystals but can be adapted for other crystalline systems. However, it is likely each model can only be applied to a single system under various mechanical distortions.

2.4.2 Spatial Equivariance and Spatial Invariance

As discussed, the shared parameters of convolutional kernels are useful when there are repeated relations between elements, either along a tensor or throughout different samples of the dataset. This reuse of parameters is attributed to their widespread success. However, it also imparts an approximate spatial equivariance of the representation.

Spatial equivariance of a neural network layer, given in *Eqn.* 16, occurs when a transformation $\hat{T}$ applied to the input $\mathbf{X}$ commutes with the layer f, or the $\hat{T}$ acting on the output is replaced, bijectively, to another transform $\hat{T}'$. The latter resolves the problematic occurrence of a smaller output tensor, $\text{size}\left(f\left(\mathbf{X}\right)\right) \leq \text{size}\left(\mathbf{X}\right)$, resulting in trivially non-commuting $\hat{T}$ and f. However, in neural convolutional layers, this is only an approximate equivariance as it generally breaks down near the edges of the tensor. Therefore, only the subspace of central elements have true equivariance.

$$f\left(\hat{T}\mathbf{X}\right) = \hat{T}' f\left(\mathbf{X}\right) \tag{16}$$

Padding involves surrounding the tensor with constant elements, artificially increasing its size without changing the original information. This has two primary benefits: alleviating the mismatching tensor shapes, such that the stronger $\hat{T} = \hat{T}'$ equivariance is enforced, alongside ensuring all elements contribute evenly to the sum of activations of the following layer. However, the translational equivariance remains solely resolved in the centre of the tensor. In addition, the output's

edges typically have smaller magnitude activations due to the usual constant-zero padding. Both problems compound over sequential convolutions, as the edge effects begin influencing more central activations. Thus, the network is only equivariant under an increasingly small central region. Over many layers, this breaks the equivariance entirely and may contribute to poorer performance in very deep convolutional neural networks [30].

The equivariance of the layer is an important consideration when designing condensed matter neural networks, as significant features are expected to occur across most of the unit-cell. Therefore, any edge effects, whilst negligible in computer vision, may be detrimental when determining the state of the material, if they have differing treatment under convolution. Hence, it is essential that any activations affected by the arbitrary padding, and also the broken edge equivariance, are resolved such that the sub-architecture is fully equivariant[2].

Furthermore, the network overall needs to be spatially invariant in its representation. This is because the unit-cell can be shifted arbitrarily by any constant vector, whilst the emergent state is invariant to the choice. This is characterised by a global spatial invariance shown in *Eqn.* 17. However, it is not desirable for the convolution layers, or prior layers, to be invariant as this would destroy beneficial information about the spatial arrangement of elements. Consequently, any local structures within the unit-cell would be disrupted and thus, so is the optimality of convolution. Therefore, the proposed network as a whole must be spatially invariant, whilst containing an enclosed equivariant convolution sub-architecture. It is also crucial that this sub-architecture is equivariant, as otherwise a following invariant network cannot be constructed.

$$f\left(\hat{T}\mathbf{X}\right) = f\left(\mathbf{X}\right) \tag{17}$$

A local spatially invariant function is also desirable to reduce the tensor size in a global equivariant manner. The benefits of shrinking the tensor size include removing redundant information, improving computation times, and increasing the receptive field size for neurons deep into the sequential structure, preferably so it spans the entire input tensor by the final layer. Thus, the whole reciprocal unit-cell is considered when deciding the state. Standard convolution could be utilised to preserve the global equivariance, however, the equivariation of activations would be less predictable with $\hat{T}'$ not resembling the transform of $\hat{T}$, leading to checkerboard average derivatives [33]. Therefore, convolution layers should always preserve the size of the tensor, requiring convolutional strides to be of length one. Instead, a function with local spatial invariance is needed, fulfilled by a pooling layer [34].

Pooling layers operate upon a local area, much like convolution, however, their operation belongs to the functional class $\mathcal{A}$, which is defined as any function whose result is invariant under the permutation of elements in the local area, ensuring the lesser spatial invariance. This removes any spatial arrangement information. The resulting layer is quasi-invariant under small shifts but is also globally equivariant. Despite, $\hat{T} \neq \hat{T}'$, the transforms are more comparable than the convolution layer. The functional class also includes functions of the form shown in *Eqn.* 18, with $a_i \in \mathcal{A}$ whilst f_i can be any function. $\mathbf{S}$ is defined as a subspace of $\mathbf{S} \subseteq \mathbf{X}$ representing the localised nature. This stricter definition includes functions such as mean and standard deviation.

$$a_1(\mathbf{S}) = f_1\left(\sum_i f_2\left(\mathbf{S}_i, a_2\left(\mathbf{S}\right)\right)\right) \tag{18}$$

Furthermore, promoting the localised pooling operation to a global one, thus using the entire tensor, allows information to be extracted from the sequential convolutions whilst ensuring a resultant global spatial invariance - given that the convolutions have a global spatial equivariance. Overall, this novel architecture ensures the physics is unaffected by the choice of the coordinate system, whilst increasing the accuracy of the model. It also helps in maintaining a feasible computing time and memory usage for the model.

Finally, the pooling functional class also includes operations such as sorts, minimums and maximums. Notably, sorting the tensor preserves the intrinsic dimension, maximising the possible information. This can be demonstated: if $\{a_i\}_{\forall i}$ is a list, such that $a_i \in (-\infty, \infty)$, and this is sorted with the smallest element being $a_0 \in (-\infty, \infty)$, then all greater elements can be re-parameterised as $\left\{a_0, a_0 + b_1, \cdots, a_0 + \sum_j b_j\right\}$ with $b_i \in [0, \infty)$. Therefore, the spatial arrangement information is lost through a smaller resultant domain after a sort. This is an important way of maximising information, despite using a global pooling operation. Given enough computational resources, the sort followed by a fully-connected layer, to remove redundancy, would be the optimal architecture, however, this is currently computationally inviable. Therefore, a combination of maximum, minimum and averages are used.

Overall, by considering crucial symmetry features of condensed matter systems, and the general coordinate invariance of physics, a functional class has been minimised for possible architectures. This approach results in distinct architectures from the larger literature and defines a set of novel machine learning models purpose-designed to analyse crystalline materials such as BiTeI. It requires an amended convolutional layer, such that it has $\hat{T} = \hat{T}'$ global equivariance, which is then sequentially stacked. Any information is then obtained using global pooling to ensure global spatial invariance. Specific architectures

[2]The equivariance is intentionally absent in the hybrid network of *Sec.* 2.4.1, due to the kernel not being iterated across the tensor, as we did not expect local repeated structures for constant $\vec{A}$.

which fulfil these criteria, and a methodology for proving their resultant invariance, are discussed in the following sections.

2.5 Defining The Problem of Gradient Diffusion

Vanishing and exploding gradients [35, 36] are common problems encountered when designing and training neural network models. Vanishing gradients are defined by progressively smaller gradient updates, tending to zero, for parameters far removed from the output. This results in slower learning progress and a sub-optimal model. It is attributed to the derivative of activation functions tending to zero, known as saturation, given in *Eqn.* 19. It can also occur when the gradient is less than one, such that the updates are scaled down after every layer in a large network. Introducing alternative activation functions [37] or batch normalisation [38] usually resolves the problem. Exploding gradients are characterised by the opposite behaviour of growing, unstable, and diverging gradient updates. It is also remedied by using batch normalisation.

$$\lim_{|x|\to\infty}\left(\frac{\mathrm{d}}{\mathrm{d}s}\sigma\left(s\right)\Big|_{s=x}\right)=0 \tag{19}$$

However, the invariant nature of the proposed networks highlights another problem which typically co-occurs with vanishing gradients. We define this as gradient diffusion, characterised by a failure of neural differentiation[3], due to gradients becoming quasi-homogenous. We theorise this often occurs in deep neural networks when successive convolutions result in a central limit theorem-like effect for gradient update values, such that neurons all learn to respond similarly as they are provided similar updates.

This is detrimental as it limits the complexity of the resultant network. Due to its association with deep networks, it may be mistakenly considered as vanishing gradients. However, it is not resolved using different activation functions or batch normalisation. It is expected to slow initial learning until a trickle-down effect from later convolution kernels correctly differentiating, begins the differentiation in the next layer. It is expected to be present in nearly every, very deep, machine learning model to date.

This problem is uniquely important when using true global equivariant convolutions with globally invariant information retrieval. For example, if the global pooling operation is chosen to be the mean, then every activation will acquire an equal gradient. This occurs when a gradient is propagated invariantly to an equivariant network, all kernels receive exactly the same update. Hence, there is no trickle-down effect to reduce the problem and the kernels fail to diversify their action. This can be alleviated by using extra, or a better choice of, pooling operations, such as the extended functional class of minimums, maximums and sorting. Despite the invariance of these functions, they result in unevenness in the activation gradients, therefore irregular updates can cause proper neuron differentiation. This is shown in *Eqn.* 20, for three global pools with functions: minimum, maximum and mean. Notationally, δ_i indicates a zero tensor of equal size to $\mathbf{X}$ with an element of value one at index i.

$$\mathrm{d}\mathbf{X}=\underbrace{\frac{1}{\mathrm{size}\left(\mathbf{X}\right)}\mathrm{d}P_{mean}}_{\text{homogeneous updates}}+\underbrace{\delta_{\mathrm{argmin}(\mathbf{X})}\mathrm{d}P_{min}+\delta_{\mathrm{argmax}(\mathbf{X})}\mathrm{d}P_{max}}_{\text{inhomogeneous updates}} \tag{20}$$

The vanishing, exploding, homogeneous and inhomogeneous gradient problems are illustrated in *Fig.* 6.

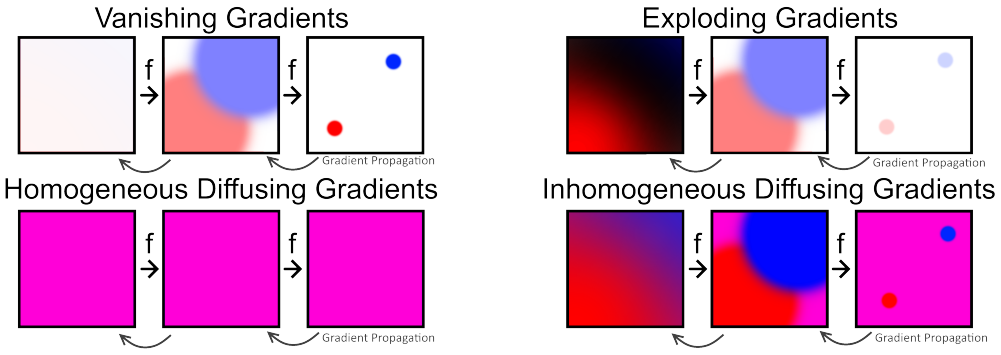

Figure 6: Displays four problems which can result from differing gradient behaviours when using a convolutional layer f. All diagrams depict the diffusing gradient problem. Top-left shows the addition of the vanishing gradient problem, whilst the top-right image shows the exploding gradient problem. If an invariant global average pool is used, it results in the homogenous diffusing problem shown at the bottom-left. The use of maximum and minimum pools results in the slightly improved problem of inhomogeneous gradients shown in the bottom-right. The colours indicate the sign and magnitude of the gradient updates being performed, with each box intended to represent a layer in a sequential convolutional architecture. The gradients are back-propagated from right to left in each series of three boxes.

[3] In this instance, biological differentiation.

Diffusing gradients is an essential consideration in the network design of *Sec.* 2.7, as these suffer minimal vanishing gradient effects, but are still prone to gradient diffusion due to the global pooling. Consequently, for the proposed networks, diffusing gradients do not coexist with vanishing gradients, isolating the problem of gradient diffusion for the first time. Therefore these networks highlight that the gradient diffusion problem is a distinct issue, which should be mitigated for network optimality.

2.6 Toroidal Convolutions

The discussion of *Sec.* 2.4.2 required a hypothetical architecture, containing a global spatial equivariant set of sequential convolutions, with a global spatial invariant method of extracting the information they concentrate. Edge effects were shown to break the equivariance of standard convolution, which then in turn breaks the invariance of the overall network. Therefore, a function with the characteristics of convolution, to optimally detect repeated local structure, whilst removing edge effects is desired.

A straightforward way to achieve this is to, in principle, fold the tensor into an n-torus hence removing any edges to cause edge effects. As a result, the tensor no longer features a centre of its representation, nor any special element, thus ensuring its equivariance. An illustration is shown in *Fig.* 23 in appendix A. This can be implemented in code by tiling the tensor $\mathbf{X}$ in the desired directions, then cropping the tensor to a standard padding amount, to ensure the original size of $\mathbf{X}$ is maintained when performing discrete convolution over the modified tensor. This is equivalent to folding the representation into an n-torus and the overall operation we name toroidal convolution.

This algorithm is demonstrated for a 2-torus folding of an image in *Fig.* 7. It can also be equivalently achieved using rolling [39] of the tensor in the desired directions. In either case, it is important to use the discrete sampling of *Eqn.* 13 to prevent double counting the edge, which would disrupt the equivariance. Also, a stride length as a factor of the tensor length is essential to ensure equivariance, but it is preferable to use a stride length of one and local pooling to reduce the tensor size. Toroidal convolution can be applied in all the same scenarios as typical convolution, such as sequentially stacking layers to yield more abstract observations.

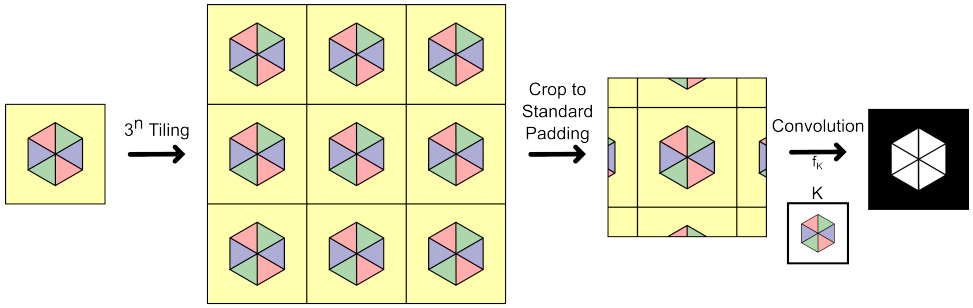

Figure 7: Shows how a standard image, with two flat rectangular spatial dimensions with one colour channel dimension, can be modified such that it behaves as a 2-torus, also retaining one rectangular channel dimension, when convolution is applied.

This is also physically significant as it correctly reproduces the discrete translational symmetry of the crystal. Therefore, it is expected that networks which feature toroidal convolution to have greater accuracy when discerning the emergent state of a material. In effect, these networks are able to represent the full reciprocal-space by using the 3-torus folding of the unit-cell. Sufficient layers should also be used to ensure information across the full space can be integrated into each activation. In combination with the spatially invariant global pooling, it generates an architecture which determines the material state independently of the chosen coordinate system.

Due to the full equivariance of the sequential convolutions, it is expected that applications outside of condensed matter physics may benefit from toroidal convolution. For example, object segmentation in self-driving cars. This is because the camera may not be able to orientate to centre the object of interest, therefore it may benefit from a toroidal convolution which does not suffer from edge effects. In addition, the proposed invariant global pooling will also be advantageous for object detection in similar circumstances. Therefore, these architectures which are purpose-built for crystalline materials, can in turn affect architectures for computer vision tasks.

Variations on the procedure can be implemented, such as an (n+m+c) flat rectangular convolution, converted to an n-toroidal, m-flat rectangular convolution with c channel dimensions. Thus, the periodicity does not need to occur in every spatial dimension, much like the channel dimension treatment. Furthermore, reflections in the tiling procedure can be implemented such that they become n-dimensional analogues of the Möbius strip, which can also be physically significant in condensed matter systems [40, 41].

2.7 Sequential Layer's Activation Complexities

It is expected that the reciprocal-space has defining structures across different length scales. For example, the closing of a bandgap, which can be localised to a single point, defines the difference between insulator-like and conductor-like states. Meanwhile, the topological invariant $\mathbb{Z}_2$ of a material is a result of the Berry phase [42, 43] which is found through a path integral along the Brillouin zone, so consists of a large-scale structure. It is desirable that the network can interpret these different scale structures, as they are often crucial in determining the state.

Sequential convolutions can fulfil this requirement as each new layer integrates a wider receptive field to construct more complex information. This results in neurons early in the convolutional pipeline responding to smaller-scale, simplistic, but highly specific structures whilst the opposite, larger abstract structures activate neurons deeper into the stack. However, as given by the above examples, it should not be assumed that the only meaningful information is derived from only these larger-scale, most complex, structures. Instead, information from all scales should be accessible to the later invariant sub-network.

There are many possible architectures which fulfil this requirement. However, we will later propose a novel form of neural network that could be much more effective, particularly when considering the physical symmetries of the problem.

2.7.1 Standard Sequential and Residual Models

The vast majority of neural networks, particularly classification networks, can be considered as only sequential models. However, it is possible to partition these into two separate networks with distinct purposes. The first can be defined as an abstraction pipeline, which progressively removes redundancy and integrates wider information into more complex but general patterns, such as the aforementioned convolutional stack. In addition, there is an interpretive network which takes the results of the abstractive pipeline and converts them into a more meaningful form for the experimenter, typically fulfilled by a fully-connected network. In reality, these networks have an overlapping function and present as a seamless structure, but it is convenient in this work to consider them as separate.

In these models, it is assumed that all useful information is contained only within the final results of the pipeline, and it is this information to which the interpretive network has access. Yet, in crystalline state classification, it is expected that important features are distributed throughout the pipeline, due to their associated scales and complexities. Therefore, it could be detrimental to perform further abstracting operations, such as convolution, on this information as it would obfuscate it.

Residual networks [30] mitigate this issue, by adding bypass connections to each operation, which allows less complex information to propagate unhindered to the interpretive network. This results in nested functional classes, with each additional layer expanding the range of possible network functions, with more powerful networks, whilst also keeping the functionality of the prior network. The residual connection is used to recover the identity operation if the alternative operation is not desired, this is because abstraction operations struggle to recreate the identity operation but can easily converge to performing the null operation with the residual performing the identity. Overall, this can be interpreted as each layer performing perturbative corrections to the original input, if beneficial. This is summarised in *Eqn.* 21, where f_i are abstractive operations, with functional class $\mathcal{F}_1$ nested in $\mathcal{F}_2$, as can be seen for $f_2(\mathbf{X}) \to \mathbf{0}$, which also highlights the perturbative nature.

$$(f_1(\mathbf{X}) + \mathbf{X}) \in \mathcal{F}_1 \subset \mathcal{F}_2 \ni (f_2(f_1(\mathbf{X}) + \mathbf{X}) + f_1(\mathbf{X}) + \mathbf{X}) \tag{21}$$

These networks have had widespread success in computer vision tasks, and it is expected that they offer an improvement in classifying crystalline systems since simplistic features of the reciprocal-space can be interpreted. In *Fig.* 8 an architecture alongside its modification to use residual connections is shown. The residual connections do not affect the global equivariance of the pipeline. It is also important, for condensed matter applications that the information is made to be global spatially invariant after the abstraction pipeline. As mentioned, this can be fulfilled by a global pooling layer, per channel.

However, there are also several major drawbacks, which may result in sub-optimal performance, particularly for the classification of crystalline systems.

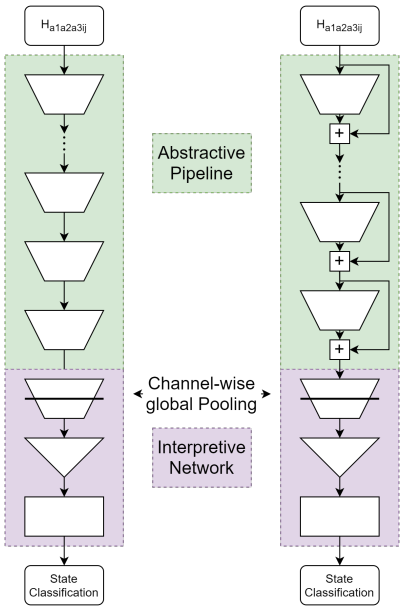

Figure 8: Left shows the standard construction of a sequential convolution model, whilst the right diagram shows the same model with additional residual connections. Each model is segmented into two sub-networks, an abstractive pipeline, in green, and an interpretive network, in purple. The global pooling per channel is also indicated.

2.7.2 Defining Deficits of the Residual Models

Despite residual models being an improvement upon the standard pipeline models, there remain four primary downsides which may severely hamper the classification of crystalline states.

1. Residual networks continue to assume that all useful information originates from a constant complexity level, even if this information stems from an earlier operation.
2. It assumes a common basis for information, throughout the structure, as can be seen by the perturbative correction interpretation.
3. A consistent basis, in turn, requires a consistent size for tensors $\text{size}\left(f_i\left(\mathbf{X}\right)\right) = \text{size}\left(\mathbf{X}\right)$.
4. The structure does little to reduce the gradient diffusion problem.

The first deficit is the most crucial for condensed matter networks. As previously discussed, it is expected that important information regarding structures in the crystal's reciprocal-space is concurrently distributed across multiple layers in the pipeline, reflecting the different scales and complexities of their origin. Despite residual networks assuming important information can occur earlier in the pipeline, it is still assumed that optimal information is only found within only one layer, with all downstream layers bypassed. In effect, it continues to be a standard sequential model but can truncate itself to become a smaller pipeline model, in an optimisable manner. This is further highlighted by the nested functional classes. However, this structure can still not retrieve information from multiple layers at once, as needed for state classification.

Meanwhile, the most far-reaching problem is the assumption of a shared basis. This results from the use of an elementwise-summation in the architecture, where the same indexed elements, per layer, are assumed to represent a common quantity which can therefore be combined together[4]. This is directly contradicting the principle of progressive layers representing increasingly complex and abstract structures. This has fallen into an elementwise fallacy where common-sized tensors encode shared information, which is not true in physical scenarios such as the clear difference between information encoded by electromagnetic field tensors and stress-energy tensors, despite their common size and transform properties.

It is likely that the network will eventually converge on a quasi-common basis, allowing for the desired perturbative corrections, or bypass, but this process is inefficient, restricts the functional class of performing models, and may require significant training time for the convergence. This may also explain the success of batch normalisation, which prevents covariate shifts and brings the two bases into closer alignment for quicker convergence.

This problem can be reinforced by a common oversight when motivating residual networks. It is shown that a neural network struggles to reproduce the identity $h\left(X\right) = x$ but can converge on the null operation $h\left(X\right) = 0$ with ease. Therefore, the $h\left(X\right) = f\left(X\right) - x$ trick exploits the unevenness, resulting in a now traded bias towards a resultant identity after the

[4]This is only reasonable when the operations are presupposed to become the identity, but this does not generally apply across all prior layers.

residual connection is recombined. However, this argument crucially admits a contradiction when asserting that $h(X) = f(X) - x$ is then equally easy to converge to as $h(X) = f(X)$, requiring an evenness which the prior argument has already established is invalid. There is evidently an inequivalence in the learnable functions $h(X)$ can adopt. Therefore, it would be expected that the neural network favours an unknown particular span for $h(X)$. This difference in convergence times is established in the original paper [30].

Qualitatively a preference could be posited for $h(X) = f(X)$ as this is what the layer would optimise to when unmanipulated, being a preference for neurons representing increasingly abstract features as opposed to that of a quasi-common perturbative basis for the residual case. Though the former basis would be advantageous to having layers representing distinctly more complex features, assuming a preferred convergence for this is merely an appeal-to-natural behaviour. Overall, the determination of the preferred basis is non-trivial and experimentation should yield further insights.

Overall, this drawback does not outweigh the improvement that residual networks offer upon the standard pipeline models, but there is still certainly scope for further improvement. For example, a concatenation, either direct-sum, *Eqn.* 22 showing before and after basis, or tensor-product, *Eqn.* 23 likewise showing before and after basis, can combine the information whilst respecting their original basis. The latter form of concatenation may also have significance in representing quantum states since it is also required to correctly represent entanglement. However, it does result in an exponential number of neurons required at each layer, as opposed to a constant growth for direct sum.

$$\{\hat{e_1}, \hat{e_2}\} \oplus \{\hat{e_1}', \hat{e_2}', \hat{e_3}'\} = \{\hat{e_1}, \hat{e_2}, \hat{e_1}', \hat{e_2}', \hat{e_3}'\} \tag{22}$$

$$\{\hat{e_1}, \hat{e_2}\} \otimes \{\hat{e_1}', \hat{e_2}', \hat{e_3}'\} = \{\hat{e_1}\hat{e_1}', \hat{e_1}\hat{e_2}', \hat{e_1}\hat{e_3}', \hat{e_2}\hat{e_1}', \hat{e_2}\hat{e_2}', \hat{e_2}\hat{e_3}'\} \tag{23}$$

A remedy to the previous problem can also be adapted to remove the constraint of a common tensor size throughout the calculation, as highlighted in problem three. In *Sec.* 2.4.2, it was shown that a reduction in tensor size, using a localised global equivariant pooling, would be beneficial. However, in residual models, this requires a step which breaks the nested functional classes. Whenever pooling is performed in a residual model, by definition a residual connection cannot exist between the before and after tensors due to their incompatibility in elementwise summation. Therefore, the pooling is performed without a residual connection, which in turn does not ensure the prior network is in a nested functional class of the extended network with pooling. This is depicted in *Fig.* 9[5]. Allowing pooling, whilst ensuring a nested functional class is a further criterion for both a purpose-built condensed matter network and wider applications in computer vision.

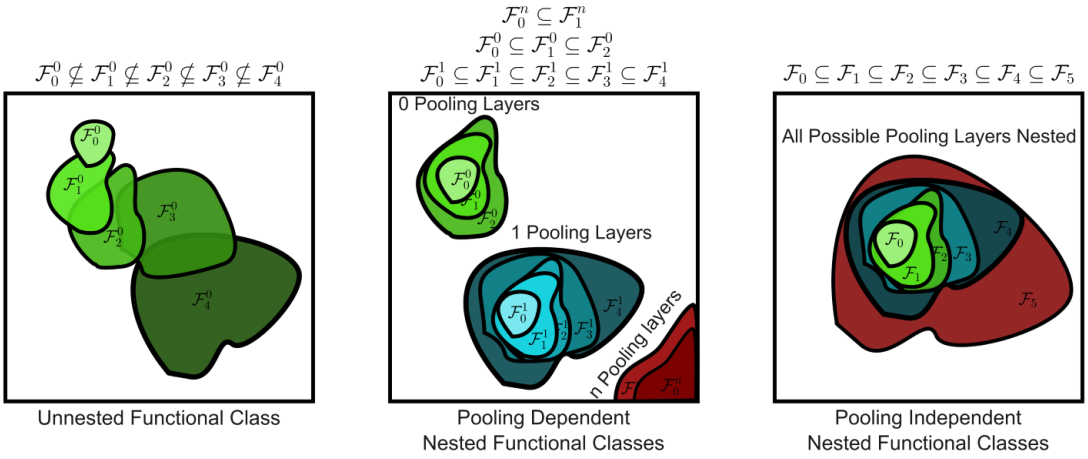

Figure 9: This illustration shows the nature of unnested functional classes (left), pooling dependent nested functional classes (centre) and full nested functional classes (right). The plot is inspired by a prior source [44]. Standard sequential convolution, with varying layers, can be considered as the functional class hierarchy of the left image. The residual network hierarchy is depicted in the central box. Each time a local pooling layer is added, a displaced nested functional class is formed. This is because padding breaks the nesting, due to being unable to reproduce the identity operation. Therefore, if the number of pooling operations is considered on equal footing to the number of convolutional layers, residual networks do not form a nested functional class. A definition of nested function classes, with variable pooling layers, is shown on the right. The upcoming medium extractor of *Sec.* 2.8 falls into this definition.

The concatenations of *Eqns.* 22, but not 23 are also shown as a solution to problem four, with mismatching sizes of the original basis. The former concatenation method has already been utilised in both the Dense network [45] and Inception network [46] architectures, however, in both cases, a very large growth occurs in the number of channels, requiring a rapid

[5]Despite being unlikely, technically convolution can reproduce the identity operation making them nested. Therefore, the captioning of *Fig.* 9 is incorrect. For this discussion this will be ignored.

reduction in tensor size to make it computationally feasible. In addition, they still require consistent tensor sizes, so only partially resolve problems one and two. Hence, neither is an ideal solution to a nested functional class network.

Finally, residual networks continue to suffer from gradient diffusion. This is because, in their initialisation, they continue to diffuse the gradients at each layer as the residual function takes training time to become the identity operation. The identity operation, by definition, does not compound the gradient diffusion as the jacobian is one-hot, as shown in *Eqn.* 24. So it is expected that this trickle-down effect occurs faster since the network can more easily converge on an identity operation. The convergence may be hastened by using a trainable parameter β such that the residual connection is redefined as $\beta f(\mathbf{X}) + (1-\beta)\mathbf{X}$ so that only one parameter controls the convergence. However, the bypass only converges in the last few layers, so diffusion still occurs in all layers prior to the bypass beginning. This reduces the problem a little but it is still expected to impede performance. This is also an issue for vanishing and exploding gradients, which are likewise partially mitigated, and present as the original motivation for residual networks [30].

$$\mathcal{J} = \frac{\partial \hat{I}(\mathbf{X}_i)}{\partial \mathbf{X}_j} = \delta_{ij} \tag{24}$$

Overall, an architecture which resolves these weaknesses of residual networks would not only be singularly important for purpose-built condensed matter networks but also may have wider applications in the common domains of deep-learning such as computer vision. In the following section, network architectures of this form are defined.

2.8 The Medium Extractor - A Novel Architecture

A set of architectures is desired that mimics the residual network's nested functional class property, addresses the aforementioned four major drawbacks and is compatible with the earlier full global equivariant of the convolution sub-architecture with global invariance of the whole network. This set of features is made possible through the proposed medium extractor architecture, which is a novel architecture for machine learning and can be applied to problems using arbitrary media, such as the unit-cell, images or spectrograms, whilst extracting information from any complexity level.

Like the residual architecture, activations are drawn out after each abstraction layer, however, unlike residual networks, dense networks and inception networks, it is not recombined later in the pipeline but instead provided directly to the interpretation network in an invariant manner. This is demonstrated in *Fig.* 10. Despite the appearance of a much more complicated structure, its implementation is code is simple and straightforward.

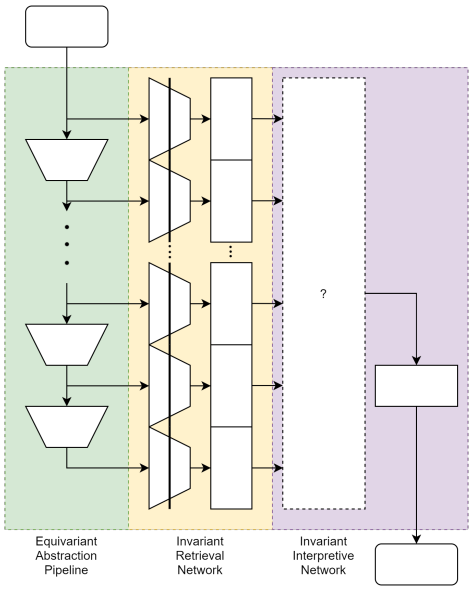

Figure 10: Shows the medium extractor architecture, with an undetermined interpretation network. It features a sequential equivariant convolution pipeline much like predecessor models and can include local pooling following the same structure. Its novelty arises from a global pool, to ensure invariance, applied per channel and per convolution layer. These are then fed into a general function defining a pre-interpretive step, followed by the interpretation network. Applying this network to non-physics tasks may require the removal of the global pooling, as the spatial invariance may not be desirable.

A separate fully-connected architecture is used between the pooling and black-box function for each information read-out, this can be combined into the black-box function using a sparse fully-connected network but is left separate for clarity. However, in either case, it is necessary as a basis transform is needed to convert from the basis used by convolution to another for the black-box function, as it may be sub-optimal to assume a common basis.

The global pooling also helps prevent massive growth in the number of neurons, such as in dense and inception networks, as each global pool retrieves only a vector with a dimension equal to the number of channels of $\mathbf{X}$, so it does not scale with the, per channel, size of $\mathbf{X}$. Hence, does not require a rapid reduction in tensor size to make model training attainable.

Alongside this fix, this general architecture meets all the predefined criteria. First, it can be shown to represent a nested functional class of architectures. This can be shown by stepwise recovering a network with one fewer convolutional layers. This is easily demonstrated, by having the fully-connected network corresponding to the last convolutional layer tending to the null operation, which is equivalent to removing the layer. The proof is shown in *Eqn.* 25, which requires direct sum concatenation and also results in a consistent growth in the number of neurons. Using this definition, a standard residual network can be recovered[6]. The nature of nested functional classes also gives a clear definition of model complexity.

$$(0 \times f_2(f_1(\mathbf{X}))) \oplus (\alpha_1 \times f_1(\mathbf{X})) \cong \alpha_1 f_1(\mathbf{X}) \in \mathcal{F}_1 \subset \mathcal{F}_2 \ni (\alpha_2 \times f_2(f_1(\mathbf{X}))) \oplus (\alpha_1 \times f_1(\mathbf{X})) \tag{25}$$

Next, it can simultaneously read out information from any given layer, so it can concurrently understand various scale features of the reciprocal-space. This can be seen by the interpretive network producing non-null operations across multiple readouts, allowing integration of all their respective information. The information from each read-out does not need to pass through any further abstraction operations, so can be directly accessed. This resolves problem one of *Sec.* 2.7.2.

Problems two and three, are satisfied by using a concatenation step, as opposed to elementwise addition, in the black-box function of *Fig.* 10. Since each readout is a vector with a number of dimensions, equal to the channels of $\mathbf{X}$ multiplied by the number of different global pooling operations, they are most generally incompatible with elementwise addition anyway[7].

Finally, problem four is eliminated, as the network is expected to perform a ground-up training routine rather than a trickle-down one. This is substantially different from all previous network architectures and we would expect very fast progress right from the start of training. In effect, the medium extractor is able to damp outputs from all untrained layers, whilst sending concentrated learning gradients to the earlier layer. Once this is trained, then the next layer up begins receiving meaningful inputs and can then become undamped to receive concentrated gradients and initiate its training. Therefore, the learning of the earliest layers is not rate-limited by the differentiation of the final layers. This sequential training of each layer is a signature of this model and will be confirmed through analysis of the black-box function.

Problem four can be further mitigated by using multiple different functions in the global pooling steps. If the standard global average pooling [47] is used, this continues to result in homogeneous gradients, as discussed in *Sec.* 2.5. However, there is no need to restrict to a single pooling operation, multiple can be used, with their respective vectors direct-sum concatenated together. Therefore, choices such as mean, minimum and maximum would result in inhomogeneous gradients, which is expected to improve performance. These also keep the dimensionality of vectors to scale with the channels of $\mathbf{X}$, not the size. However, given more computing power, sorting could instead be used which would scale with $\mathbf{X}$, and would lead to greater inhomogeneous gradients.

In conclusion, this design has all the major features of residual networks whilst correcting its weaknesses. The architecture can also recover a residual network, showing that the residual network's functional class is itself a subset of the medium extractor's[8].

This architecture marks a significant departure from a standard pipeline design, much like long-short-term memory networks [48] and to some degree transformer architectures [19]. Since it is purpose designed for condensed matter applications, it is expected to have a particularly good performance in this domain, though is proposed to have wide application in many machine learning applications.

2.8.1 Tuned and Free Attention

The transformer architecture benefits from a mathematical attention function, which allows information from different sources to be combined in varying magnitudes, depending on the scenario. This definition can be generalised to the medium extractor model, allowing information from each complexity level to be read out based on its need in determining the current crystalline system's state. This can be incorporated as the black-box function of *Fig.* 10.

The base case is tuned attention, which is static per sample. In effect, the network learns which layers are on average important and emphasises these. This is simply implemented by the black-box function being a direct-sum concatenation of its inputs, then passing them through a fully-connected network. Except through parameter updates, it does not enable live rebalancing of emphasis across the complexity levels in the convolution pipeline for each sample. This simple implementation is displayed in *Fig.* 11. However, it is predicted to perform poorer than dynamically shifting attention. It is still expected that it will perform the signature ground-up training, which should be clear in the fully-connected network's weights.

[6] In its invariant form, otherwise, the global pooling step can be removed to recover its general form.

[7] Unless the fully-connected network unnecessarily reduces them to a consistent size vector.

[8] The medium extractor is also not mutually exclusive with a residual design too, since bypass connections can be added to the convolutional stack. Although there appears little point in doing this as this would reintroduce problems two and three. Though this combination will still be tested, to isolate these problems and show their detrimental effect.

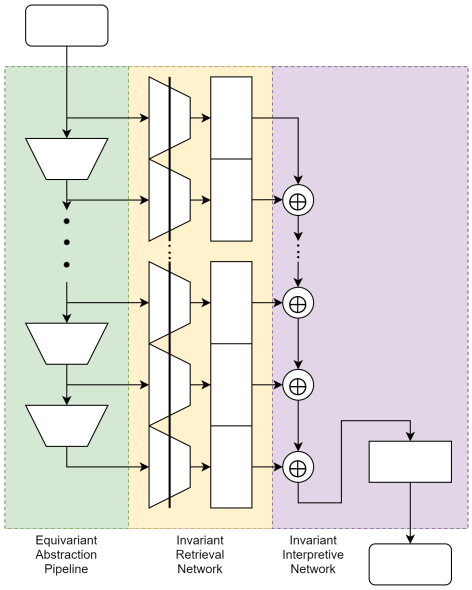

Figure 11: Shows the tuned attention form of the medium extractor. It features a direct sum of each read-out, which is then passed through a fully-connected network. This allows the network to learn which read-outs are more significant and weight these more strongly in its decision-making, such as crystalline state classification.

Free attention is more similar to that seen in transformers, with the network dynamically assigning importance to a particular read-out's information. First, a similar construction to the tuned medium extractor is needed, but instead of a vector giving the state classification being produced, a vector representing the significance of each layer's information is formed. Therefore, for n total convolutional and padding operations, this network should have $n + 1$ neurons for each respective read-out.

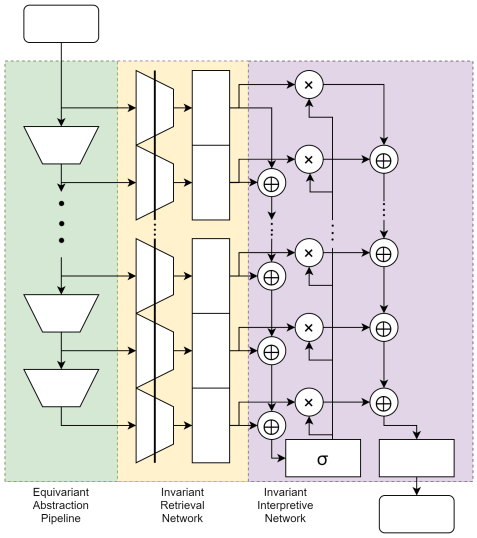

Figure 12: Shows the free attention form of the medium extractor. As opposed to the tuned form, it can dynamically redistribute its attention based on the given sample.

This vector must contain elements in the range $[0, 1]$ and can be normalised or not. Both sigmoid [49], which is unnormalised, and softmax [50], which is normalised, are activation functions that can be applied to this vector to achieve this. The normalised case results in conserved attention, so it must be redistributed across all read-outs when placing emphasis on a particular read-out. In addition, this requires a normalisation of each read-out to ensure the proper functioning of the attention mechanism, otherwise tuned parameters can compensate for the effect. This can, preferably, occur within the sequential convolution stack, or alternatively after the global pooling. Batch normalisation could be used, though this leads to inconsistent equivariance. Instead, normalising each individual tensor may be preferable, to maintain a more constant equivariance.

Each of these attention values then modulates each read-out, before direct-sum concatenating them and passing the

resultant tensor through a fully-connected architecture to determine the crystalline state. This architecture includes an activation non-linearity, due to the modulation interaction, so may benefit from a reduced learning rate to combat chaotic instabilities. This free medium extractor architecture is illustrated in *Fig.* 12. Its code implementation is also uncumbersome.

Overall, this form of neural network incorporates the spatial equivariance and overall invariance needed for representing the discrete symmetry of the reciprocal unit-cell, alongside making its state classification decision based on a range of varying complex structures throughout the unit-cell. Therefore, it constitutes a purpose-built neural network for condensed matter physics, which correctly respects, and leverages, the symmetries of the problem to its benefit. Therefore, compared to repurposed architectures from computer vision, it is expected to have improved accuracy on tasks such as material state classification, critical temperature prediction, and magnetic behaviour prediction, alongside many other crystalline system characteristics. It may also have a wider impact on other deep-learning domains, which should be explored.

2.9 Generating Datasets for Benchmarking on BiTeI

The real-space networks of *Secs.* 2.4.1 and 2.3, are to be evaluated on a dataset of hoppings from the crystal BiTeI. Four emergent states are possible for this crystal under various mechanical distortions: the trivial insulator, topological insulator, Weyl semimetal and Dirac semimetal. The Dirac semimetal only occurs when the crystal is both time-reversal and spatial-inversion symmetric, so in practice, not a single sample will occur in the dataset due to the infinitesimally small chance that sampled hoppings will have both symmetries.

As described in the prior project [7], samples will be interpolated between two instances of BiTeI under different hydrostatic pressures. One at ambient pressure, in a trivial insulating state, with the other at higher pressure, in a topological insulating state. Interpolating between these pressures produces a range of BiTeI configurations in both insulating and Weyl semimetal states, which are identified using the procedures in the prior project. A dataset will be compiled using this procedure.

Circumstances when networks are trained for a few epochs may benefit from a dataset containing an equal number of samples from each state, to avoid network bias. However, this is not necessary for the networks described in this project.

Despite each sample containing many hoppings, even after the truncation described in *Sec.* 2.3, therefore a high extrinsic dimension [7], using interpolation results in an intrinsic dimension of one. This classification of an embedded one-dimensional manifold discourages the network from understanding the physical nature of the system. The addition of noise, described in *Sec.* 2.1.3, increases the intrinsic dimension and another dataset will be assembled based upon this procedure. Again, the classification of these perturbed hoppings will be achieved using the methodology developed in the prior project.

Proving that the perturbations increase the intrinsic dimensionality can be undertaken using singular value decomposition [51] (SVD) of the dataset or visually using t-stochastic neighbour embedding [52,53] (t-SNE). The latter generalises to curved manifolds and is the one used in this experiment. For SVD, the number of non-zero singular values indicates the intrinsic dimensionality, whilst the t-SNE visualisation would show more structure than a line. A more quantitive approach could be achieved using a neural autoencoder [54], with a bottleneck layer with varying numbers of neurons. This would also generalise to curved manifolds with the number of neurons at which the network error drops would indicate the intrinsic dimension. Though this is out-of-scope for this project, as the exact intrinsic dimension is not needed, just the knowledge that it has increased.

The time-reversal and hermitian symmetries of the dataset can also be confirmed, by transforming the respective Hamiltonians and seeing if they are unchanged.

2.10 Sample Rolling of Classical Computer Vision Tasks

Proving the invariance, and thus equivariance, of the network is troublesome using a crystalline system, due to the inherent implementation complexity and very long computation times required. Therefore, to ensure the desired performance it is best to evaluate performance on existing neural network benchmarking datasets. This is made possible due to the cross-compatibility of the proposed condensed matter networks with computer vision problems.

Initial results will be gathered on the MNIST dataset [16] and then confirmed using the more demanding CIFAR10 dataset [17]. In each case the image will be rolled [39] by various amounts, allowing the invariance to be quantitatively deduced. To achieve this, a hyper-parameter, $\chi \in \mathbb{Z}^+$, named the "degree of rolling" is used to produce two uniform random integers, $\omega_x \in [-\chi, \chi]$ and $\omega_y \in [-\chi, \chi]$. The parameter ω_x defines the number of rolls in the horizontal axis, and likewise ω_y in the vertical axis. The procedure, with differing random variables ω drawn per sample and per batch, will be performed throughout the training and evaluation of the networks. Therefore, varying χ indicates the invariance, where the invariant model's accuracy will be unaffected by the change.

MNIST, though quick to train on, is not optimal to infer invariance as it contains large areas of zero-values. This offers a poor distinction between the toroidal convolution and the zero-padded convolution as equivariance is only broken for large χ. Specifically, $20 < \chi \approx 28$, where edge effects begin to occur. CIFAR is better as edge effects occur even for unrolled samples of $\chi = 0$. CIFAR also has a greater variation in object scale and absolute position, much like features in crystalline systems. However, in both cases, the object of interest is still approximately centralised to standardise the dataset. This is unrealistic

to the general applications of toroidal convolution, so an unstandardised dataset such as ImageNet may be the best domain. Unfortunately, due to its size, it is out-of-scope for our computer hardware.

Finally, these images do not feature discrete translational symmetries which define them topologically as a 2-torus. Consequently, a discontinuous border is created when the folding occurs. This does not impede the measurement of equivariance and also has a negligible effect on performance since the network will learn to ignore this feature and the images contain many discontinuous borders regardless.

3 Results & Discussion

3.1 Properties of BiTeI Dataset

Both the unperturbed and perturbed BiTeI datasets were generated with 60000 samples each. These were each partitioned into a training set of 54000 chosen samples and a testing set of 6000 samples, these datasets do not have overlapping samples and every sample is unique. The structure of each dataset is shown in *Tabs.* 1 and 2 respectively.

State Classification	Training Samples	Testing Samples	Total Samples
Trivial Insulator	42613	4697	47310
Topological Insulator	10802	1229	12031
Weyl Semimetal	585	74	659

Table 1: Number of samples representing each material state in the unperturbed dataset

State Classification	Training Samples	Testing Samples	Total Samples
Trivial Insulator	41145	4604	45749
Topological Insulator	10781	1203	11984
Weyl Semimetal	2074	193	2267

Table 2: Number of samples representing each material state in the perturbed dataset.

The number of Weyl semimetal classifications in the perturbed dataset is notably larger. This is in agreement with the expected broadening of the gapless region of the phase space, with a larger magnitude of inversion-symmetry breaking terms in the Hamiltonian [20, 21]. This is also demonstrated in *Fig.* 13, where the minimum bandgap is plotted for interpolated samples, with various magnitudes of noise added.

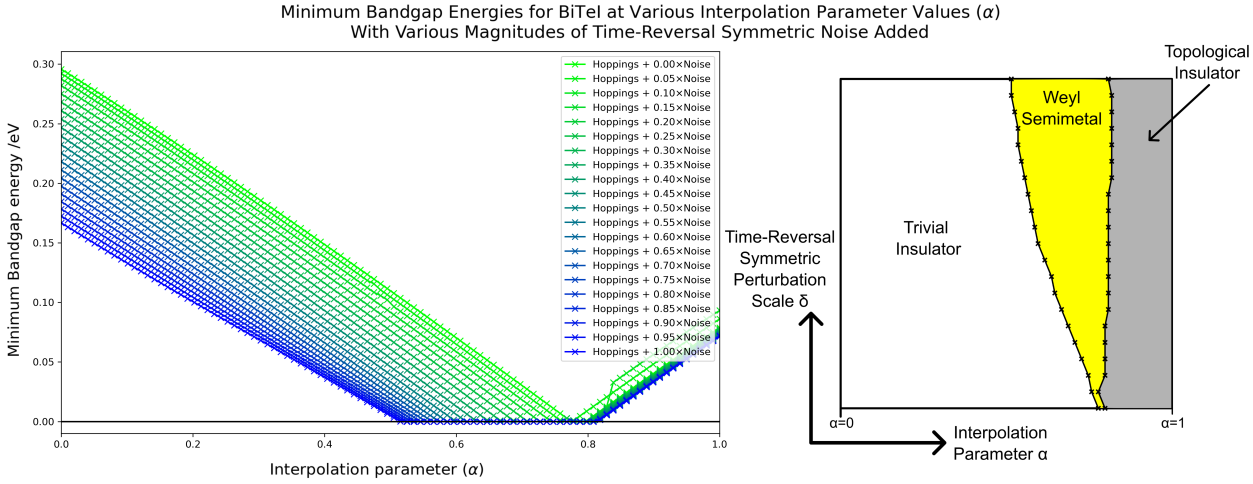

Figure 13: Left shows the minimum bandgap energies of BiTeI along an interpolation line with various magnitudes of noise added: $E_{\min}\left((1-\alpha)\,\hat{H}_{\text{Tri.}} + \alpha\hat{H}_{\text{Top.}} + \delta\Delta\hat{H}_{\text{Noise}}\right)$. It can be seen that increasing the magnitude of time-reversal noise increases the range of the gapless Weyl semimetal state. On the right is a recreation of Murakami and Kuga's [21] state space plot, using real data from the BiTeI state boundaries from the left plot. This also demonstrates the broadening of the gapless state range. The noise parameter $\Delta\hat{H}_{\text{Noise}}$ is a constant throughout, with mean $\left(\left|\Delta\hat{H}_{\text{Noise}}\right|\right)$ = mean $\left(\left|\hat{H}_{\text{Tri.}}\right|\right)$.

It was found that the addition of time-reversal symmetric noise successfully increased the intrinsic dimensionality of the dataset, as can be seen in *Fig.* 14 and a t-SNE embedding image in *Fig.* 24 in appendix B. The noise is also shown to be

symmetric under hermitian and time-reversal transformations in *Fig.* 25 of appendix B.

Overall, this analysis has shown that time-reversal symmetric perturbations can be successfully implemented in code and added to the existing interpolated Hamiltonians. This reproduced the expected broadening of the gapless state's range along the interpolation line. In addition, the data manifold's dimension was shown to increase. This ensures that the neural network uses a physical method of solving the problem, as opposed to a shortcut which doesn't learn the nature of the crystalline system. This work enabled the compilation of two very large datasets, with and without these perturbations, for training the neural networks of *Sec.* 3.3.

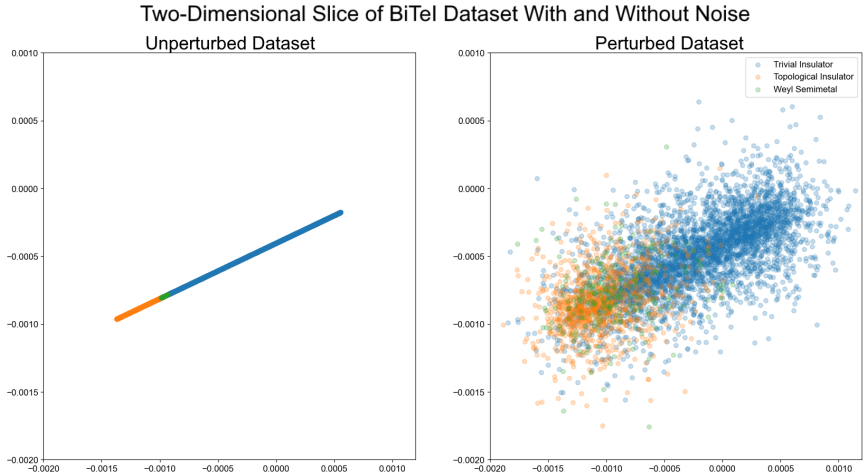

Figure 14: Shows a two-dimensional slice of Hamiltonian samples in the unperturbed (left) and perturbed (right) datasets. It can be seen that the unperturbed dataset is an embedding of a one-dimensional manifold, whereas the perturbed dataset demarcates a high-dimensional manifold's volume. The colour of the sample indicates the crystalline state.

3.2 Evaluation of Invariance on Computer Vision Datasets

Initially, results were gathered on the simpler MNIST dataset to determine if a network architecture is invariant, and if not how much it could approximate invariance. The fully-connected architecture does not have any repeated tuned parameters, so it is not expected to exhibit spatial invariance. This was confirmed in the analysis shown in *Fig.* 15, for networks of structure: small $[784, 20, 10]$, medium $[784, 512, 256, 10]$ and large $[784, 525, 525, 525, 525, 525, 525, 10]$. All networks are trained using cross-entropy loss [55] and a learning rate of 0.0001, across 10 epochs with 3 repeats, with resampled random initialisations of the networks, for each value of χ.

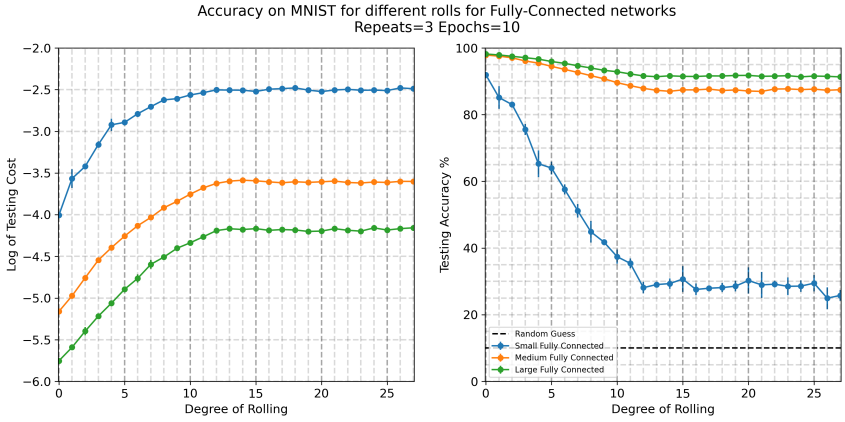

Figure 15: Shows how the performance of a network varies with increasing amounts of sample rolling. On the left is the logged cost whilst on the right is the accuracy on MNIST. Three different fully-connected networks were evaluated with various amounts of neurons. The larger networks are able to learn a better approximate spatial invariance. A total of 252 networks were trained to produce this plot.

The larger fully-connected networks appear to have the capacity to learn an approximate spatial invariance as shown by their higher accuracy at large χ, whilst the small network has the poorest performance. All networks have the highest accuracy at $\chi = 0$, as expected, and appear to stagnate in performance at larger χ. Therefore, using a fully-connected network is undesirable for condensed matter physics, due to its inherent bias towards certain coordinate systems.

In *Fig.* 16, the effect of reducing the tensor size using local pooling versus convolution was tested, across three variations of the networks.

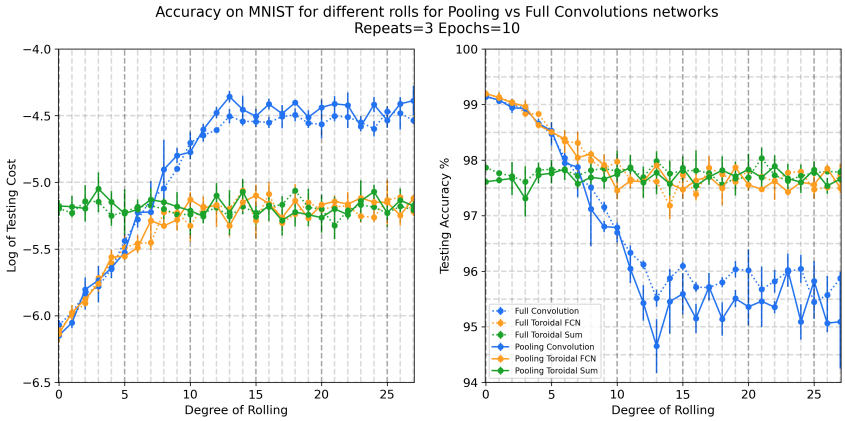

Figure 16: Likewise shows the logged cost (left) and accuracy on MNIST (right) for a range of networks. The architectures are equally structured, with "convolution" being the standard convolution, "ToroidalSUM" being all toroidal convolution layers and "ToroidalFCN" being all toroidal layers except the final one which is standard convolution, which in this case is equivalent to a Fully-Connected Network. The "Sum" suffix indicates the global average pool being used as the final pipeline layer. As expected the convolutional architectures outperform the fully-connected architectures across all χ. The network architectures are displayed in *Figs.* 26 and 27 of appendix C. Networks prefixed with "Pool" indicates the use of a local pooling operation to reduce tensor shape, whilst "Full" indicates a convolution is used, with stride$\neq 1$ to reduce the tensor shape. These networks have an equal number of trainable parameters and a total of 504 networks were trained to produce this plot.

It was expected that local pooling would be better since $\hat{T}$ resembles $\hat{T}'$ more, but this was not found as both variations resulted in statistically equal accuracy. This indicates that the network can compensate for the variability caused by the trainable convolution layer. Therefore, when constructing the condensed matter network, either can be used, however, we continue to use pooling.

The spatial invariance is clearly present in the Toroidal-Sum network, with performance independent of χ. However, the convolutional and Toroidal-FCN have better performance at small χ, indicating that some breaking of the spatial invariance was advantageous. This may be a result of insufficient convolutional layers used, so not all information from across the input was integrated into the receptive field of the final layer neurons, as well as the gradient dilution problem occurring generally. Overall, the novel Toroidal-FCN appears to be the optimal network but doesn't have the invariance needed for physics.

In the following results, CIFAR10 is used instead of the MNIST datasets, as this harder task provides a better distinction between the network's performances. The epochs are increased to 15, and the learning rate to 5×10^{-5}, due to the increased difficulty of the task. In *Fig.* 17, the standard pipeline networks are compared to a residual network.

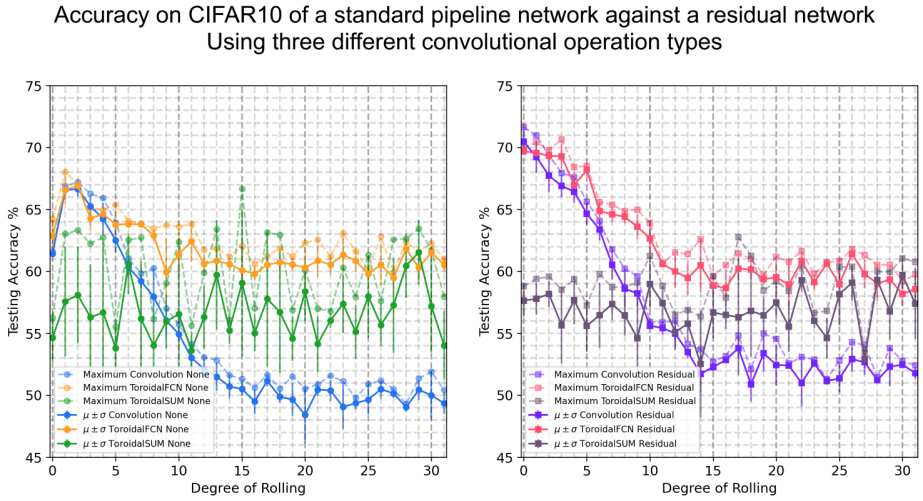

Figure 17: Left shows the accuracy on MNIST for several standard pipeline networks, whilst on the right shows their residual equivalents. An equal number of tuned parameters and kernel sizes are used in all models. The graph displays both the mean accuracy with errors, alongside the maximum accuracy across the three repeats. A total of 576 networks were trained to produce this plot.

Both residual and standard pipeline networks have similar performance except for small χ, where residual networks have a significant improvement. The standard pipeline networks reach a maximum performance $\chi \approx 2$, whilst the residual network has a much higher performance at $\chi = 0$. This is likely for two reasons. The first is that the networks are too deep, for the simplest task at $\chi = 0$, and thus the identity operation is being utilised to bypass the layers, as intended. It also may infer that there is uncertainty in the standardisation, as it is difficult to consistently centre the complex images present in the CIFAR10 dataset. This would explain the rise at $\chi \in \{2, 3\}$, as there exist more samples with the uncertainty in centring. Again, the toroidal-FCN performs the best across all χ, but the convolution is marginally more successful for $\chi = 0$, and the toroidal-sum is consistent with an invariant network. The success of convolution is likely due to the lesser impact of the gradient diffusion problem.

Moreover, the lower overall performance of toroidal-sum is likely a direct result of homogenous gradient diffusion. Therefore, these repurposed computer vision architectures, made to be invariant, appear to be sub-optimal for modelling crystalline systems. The medium extractor resolves this problem and is compared with residual networks in *Fig.* 18

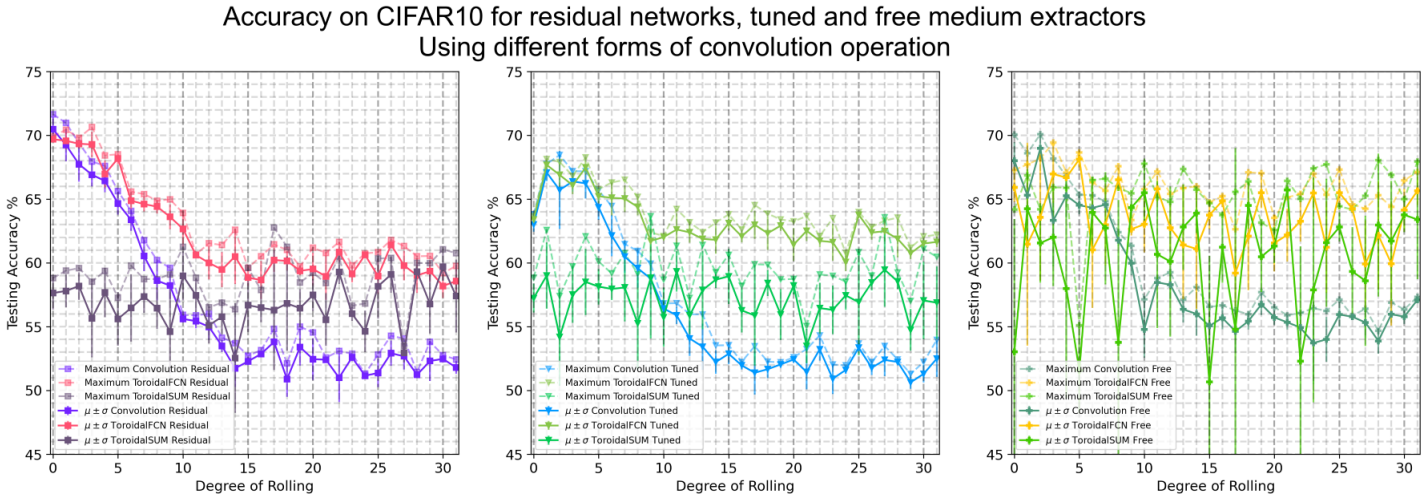

Figure 18: Shows the residual network (left), tuned attention medium extractor (centre) and free attention medium extractor (right), each with a standard convolution, toroidal-FCN convolution and toroidal-sum convolution variant of the abstraction pipeline. An additional 576 networks were trained to produce this plot. Despite the residual network, tuned and free attention medium extractors having the same number of trainable parameters in the abstraction pipeline, the free attention form has additional parameters in the interpretation networks due to its implementation. Therefore, it is important to note that it has a slight advantage, however, this does not account for the vastly improved performance of the toroidal networks. The growth in the interpretation network is also disproportionately small compared to the additional number of read-outs, therefore, the networks should remain comparable.

As before, the convolutional architecture is shown to quickly deteriorate in performance for large χ, whilst toroidal-sum is invariant and toroidal-FCN is the optimum across χ. The tuned medium extractor slightly outperforms the residual network for large χ but continues to have poorer performance for smaller degrees of rolling. This is likely due to over-abstraction and a failure of the tuned medium extractor to bypass the final layers. Though it appears the mechanism functioned as intended for larger χ since a greater performance is observed. This suggests that there is a benefit in withdrawing information from multiple layers concurrently, confirming the first problem proposed in *Sec.* 2.7.2.

The free attention medium extractor is successful across all χ, particularly for the toroidal-sum variant. This confirms that there is a large benefit in dynamically allocating attention across the various convolutional layers. In contrast with the tuned medium extractor, it indicates that each sample has important information occurring across convolution layers, but the particular layers are dependent on each sample. The common dip around $\chi = 0$ is also absent, indicating that the network successfully bypasses redundant convolution layers much like the residual network, inferring it correctly formed a nested functional class. In comparison to the tuned-attention case, it suggests that the need to bypass particular layers fluctuates with the sample, explaining the poorer performance of the tuned-attention model. The free-attention model does feature some instabilities, likely due to the activation non-linearity. A lowered learning rate should alleviate this problem, however, the primary cause is later found to be the attention mechanism.

Overall, the free attention medium extractor appears to parallel the success of the residual model for smaller χ, with growing benefits for larger χ. This implies it will be highly successful in real-world uncentered problems. Testing on ImageNet is needed to confirm this. This is an early indication that custom machine-learning algorithms for condensed matter systems are particularly successful. However, the highest performance overall was observed for the high-bandwidth medium extractors shown in *Fig.* 19.

The term "high bandwidth" is used to refer to multiple forms of global pooling being used. Namely, mean, maximum and minimum global poolings were used per layer, with the respective vectors direct-sum concatenated. This architecture

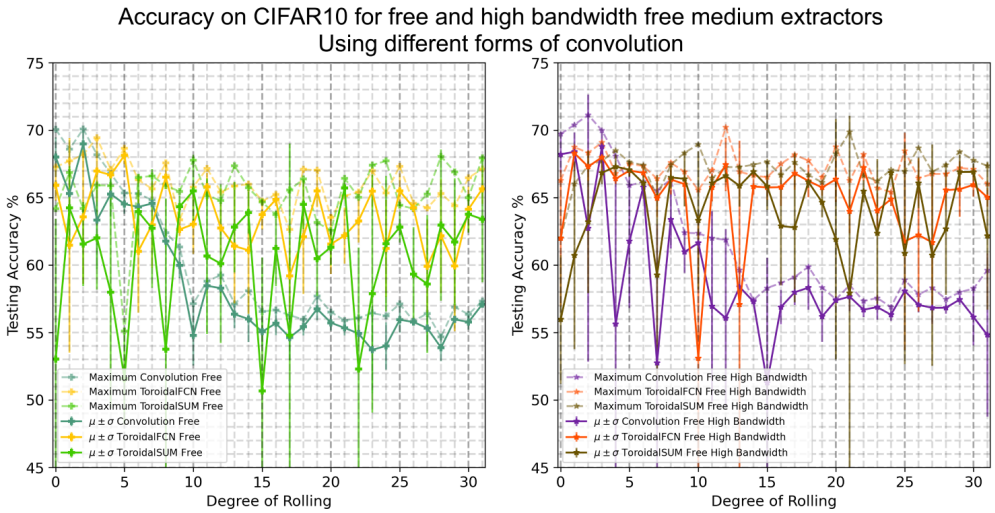

Figure 19: The standard free-medium extractor is shown on the left, using only global mean pooling, whilst the high bandwidth free medium extractor is on the right. This latter network uses mean, maximum and minimum global pools per channel per layer. An additional 288 networks were generated for this plot.

is particularly significant as it isolates the gradient diffusion problem. The greater accuracy achieved by this network corroborates the existence of the problem, as both vanishing and exploding gradients are mitigated by the use of Leaky-ReLU, batch-normalisation and dynamic attention forming a nested functional class. Reduced bottlenecking may also factor into the improvement, however, this does not explain the equalisation between the toroidal-sum and toroidal-FCN networks. Due to the toroidal-FCN containing an invariance-breaking layer, only inhomogeneous gradient diffusion occurs across its architectures. Yet inhomogeneity only occurs for toroidal-sum in the high bandwidth architecture. Therefore, their indistinguishable performance in the high bandwidth case, suggests it is a direct result of, particularly the homogeneous, gradient diffusion being alleviated. Although homogeneous and inhomogeneous gradient diffusion is also generally resolved by the attention mechanism. Therefore, the discrepancy in accuracy experimentally validates the presence of this problem. Consequently, this problem should always be considered in the development of spatially invariant networks, such as those required in physics applications.

Finally, the existence of problems two, and by consequence three, of *Sec.* 2.7.2 can be established using a hybrid architecture between residual networks and high bandwidth free attention medium extractors. The hybrid network consists of the medium extractor with residual bypass connections. This architecture selectively reintroduces these two problems, so a drop in performance would directly infer their presence. This is observed in *Fig.* 20.

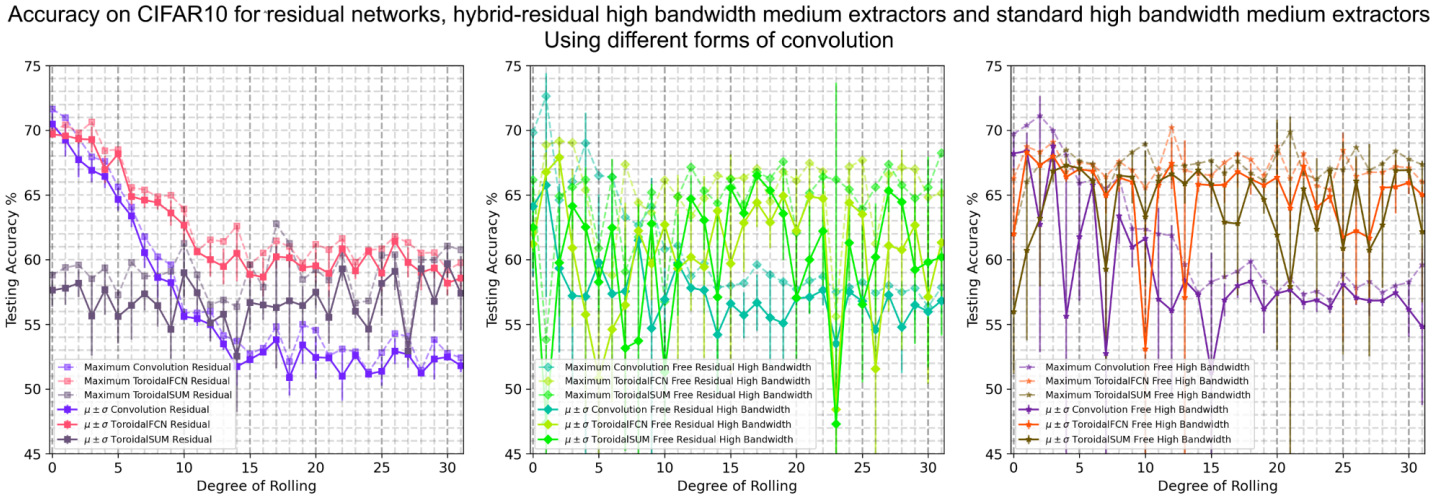

Figure 20: Shows network performance of residual networks (left), hybrid high bandwidth residual free medium extractors (centre) and high bandwidth free medium extractors (right). The hybrid network is shown to have performance between the residual network and high bandwidth network, indicating that the suggested four deficits of residual networks are correct assertions.

The sandwiched performance of the hybrid network not only directly confirms the existence of deficits two and three of residual networks, but also infers problems one and four, as the standard high bandwidth medium extractor is shown to have

generally greater performance over the standard residual network across all $\chi > 0$. This is further corroborated by *Fig.* 21 showing the performances of the toroidal-sum networks collated for all χ and across all architectures.

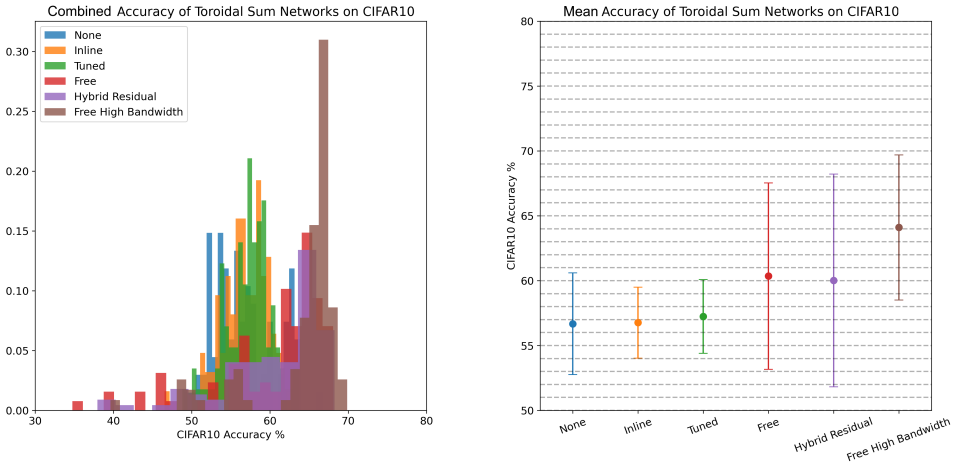

Figure 21: Shows the combined accuracies of toroidal-sum architectures across all χ. It indicates that the high bandwidth, free attention model is the optimal network for spatial invariance. The improvement of the medium extractors over the prior models helps confirm the deficits proposed in *Sec.* 2.7.2. Inline is used synonymously with residual. In addition, the best performances reached by the medium extractors infer the existence of the gradient diffusion problem.

As expected, the dynamic attention signature of progressing up increasing layers was observed during training, confirming the ground-up learning behaviour. This is shown in *Fig.* 22.

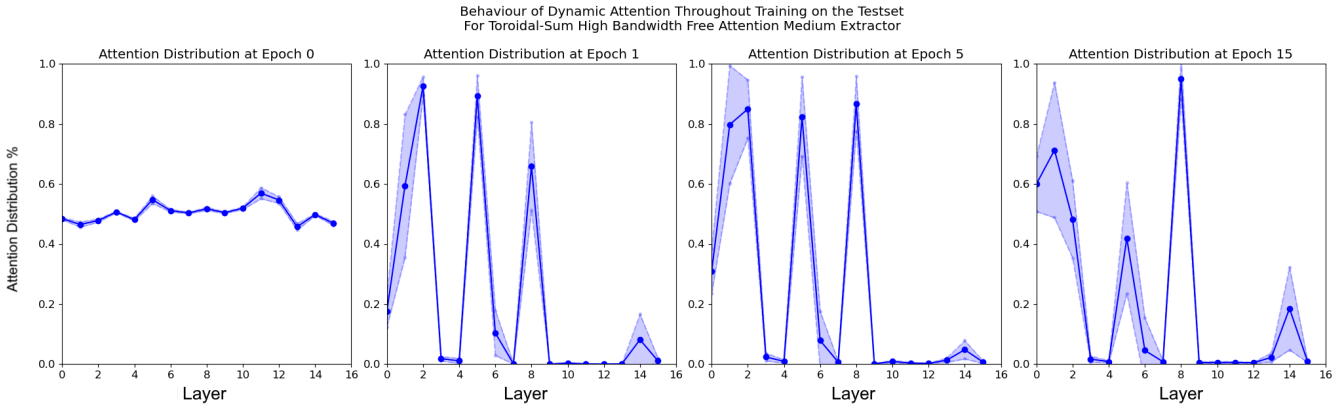

Figure 22: Shows the mean and standard deviation of attention distribution across 1250 samples of the test set, at various epochs along the training process. The results are from the high bandwidth, free attention, toroidal-sum medium extractor. It shows that before training, epoch=0, the attention is evenly distributed, it then redistributes to prioritise early layers. Once these are trained, their attention is damped whilst the next layer becomes undamped. This behaviour then progresses. It can be seen that layers 2, 5, 8 and 14, appear to have particularly important information consistently, indicating these scales have significant structures in this problem. The error bars indicate the standard deviation of attention across the samples.

It was also found that the prior instability is primarily a result of the common sigmoid saturation, as shown in *Fig.* 28 in appendix D. In effect, layers are permanently damped to zero, detrimentally affecting performance. This is alleviated by a rescaling of the attention as $\epsilon + (1 - \epsilon)\,\sigma\,(x)$ for $0 \leq \epsilon \ll 1$ and sigmoid activation $\sigma\,(x)$, to ensure attention always remains non-zero. Preliminary results indicate this is a successful remedy, but there was insufficient time for extensive testing. The results of all networks, evaluated on the CIFAR10 dataset, are collated in *Fig.* 29 in appendix D.

Therefore, it can be concluded that the high bandwidth, free attention medium extractor, using a full toroidal convolution abstraction pipeline, is the best architecture for predicting the properties of crystalline systems. Residual networks, without invariance, still have a good performance on highly standardised tasks but the broken invariance is sub-optimal for condensed matter problems. This indicates that a custom neural network approach to modelling crystalline systems is highly effective and significantly outperforms repurposed computer-vision-originating designs. It also suggests these models may have widespread implications for standard machine learning domains, however, more testing, such as on ImageNet, is required to determine whether they are truly state-of-the-art. The problems proposed earlier in the project were also isolated to confirm their existence and demonstrate the detriment. This may also have a wider impact on other domains of machine learning.

3.3 Machine Learning applied to Condensed Matter Physics

Finally, the accuracy of network architectures developed in *Secs.* 2.3 and 2.4.1 are benchmarked using the datasets developed in *Sec.* 3.1. The results for the unperturbed dataset are tabulated as confusion matrices in *Tabs.* 3 through 5. Each data point represents ten repeats of identical architectures with different initialisation, trained over 25 epochs. The average and uncertainty of each network's accuracy over these ten repeats are displayed.

		Training Set Classification (%)			Testing Set Classification (%)		
		Trivial	Topological	Weyl SM.	Trivial	Topological	Weyl SM.
Small Fully	Trivial	100.0 ± 0.1	0.0 ± 0.0	47.1 ± 29.4	100.0 ± 0.1	0.0 ± 0.0	55.5 ± 29.1
Connected	Topological	0.0 ± 0.1	99.7 ± 0.5	43.4 ± 31.7	0.0 ± 0.1	99.6 ± 0.8	34.5 ± 30.5
Prediction	Weyl SM.	0.0 ± 0.0	0.3 ± 0.5	9.5 ± 10.5	0.0 ± 0.0	0.4 ± 0.8	10.0 ± 10.6

Table 3: Confusion matrices for accuracy of the small fully connected network presented in *Sec.* 2.3 on the unperturbed training and testing dataset of *Sec.* 3.1.

		Training Set Classification (%)			Testing Set Classification (%)		
		Trivial	Topological	Weyl SM.	Trivial	Topological	Weyl SM.
Large Fully	Trivial	99.7 ± 0.7	0.5 ± 1.3	36.3 ± 42.3	99.7 ± 0.7	0.7 ± 1.9	38.2 ± 42.9
Connected	Topological	0.1 ± 0.3	97.6 ± 3.4	24.1 ± 32.3	0.1 ± 0.2	96.9 ± 4.2	21.8 ± 31.1
Prediction	Weyl SM.	0.2 ± 0.4	1.9 ± 2.4	39.7 ± 33.5	0.2 ± 0.5	2.4 ± 3.0	40.0 ± 34.6

Table 4: Confusion matrices for accuracy of the large fully connected network presented in *Sec.* 2.3 on the unperturbed training and testing dataset of *Sec.* 3.1.

		Training Set Classification (%)			Testing Set Classification (%)		
		Trivial	Topological	Weyl SM.	Trivial	Topological	Weyl SM.
Novel	Trivial	99.7 ± 0.4	0.7 ± 2.0	29.5 ± 37.6	99.8 ± 0.3	1.0 ± 2.6	32.6 ± 37.9
Convolution	Topological	0.0 ± 0.1	98.8 ± 2.6	34.8 ± 43.3	0.0 ± 0.1	98.4 ± 3.2	33.8 ± 43.4
Prediction	Weyl SM.	0.2 ± 0.4	0.5 ± 0.7	35.7 ± 35.6	0.2 ± 0.3	0.6 ± 0.9	33.6 ± 33.9

Table 5: Confusion matrices for accuracy of the novel convolutional architecture presented in *Sec.* 2.4.1 on the unperturbed training and testing dataset of *Sec.* 3.1.

It can be seen that the small fully connected network performs the worst, likely due to insufficient parameters to correctly model the behaviour. Whilst both the large fully connected and novel convolution models had comparable accuracy, likely due to it being an embedding of the problem of classifying a one-dimensional line. Similar performance is also seen between the testing and training sets across all models, indicating that overfitting did not become a problem in the modelling.

The Weyl semimetal state was most often misclassified, particularly as a trivial insulator. This reflects the underlying proportions of the dataset, with trivial insulators being the most common example. This could be improved in three ways: using an alternative cost, resampling the dataset and longer training times.

The cost function used to train these networks effectively tries to improve the overall accuracy of the network, this gives a bias towards predictions classifying a material as a trivial or topological insulator. Instead, a cost function can be constructed to maximise the trace of these confusion matrices. Therefore, it prioritises correct classification even for samples which are underrepresented in the dataset, such as the Weyl semimetal.

Resampling the dataset, to include equal samples of each classification could also be performed, however, this is not ideal. Since the true classification is calculated independently for every sample, it is not possible to guess which classification it will become ahead of analysis[9]. Therefore, resampling the dataset would involve only removing information which is undesirable, as typically a larger dataset should always be prioritised.

Finally, longer training times would likely resolve the poor performance on Weyl semimetals. This is because the other two classifications have nearly 100% accuracy, therefore, the only way to continue to reduce the cost is to begin classifying Weyl semimetals correctly.

The corresponding confusion matrices for the same networks, but trained on the perturbed dataset, are shown in *Tabs.* 6 through 8.

[9]It is possible to predict the classification for the unperturbed samples, as boundaries α_1 and α_2 were found to suitably distinguish the states in the previous project [7]. However, this does not generalise to the perturbed dataset.

		Training Set Classification (%)			Testing Set Classification (%)		
		Trivial	Topological	Weyl SM.	Trivial	Topological	Weyl SM.
Small Fully	Trivial	94.3 ± 0.9	25.3 ± 3.5	50.2 ± 8.2	95.8 ± 0.9	17.2 ± 3.6	52.2 ± 9.1
Connected	Topological	5.7 ± 0.9	74.7 ± 3.5	49.8 ± 8.2	4.2 ± 0.9	82.8 ± 3.6	47.8 ± 9.1
Prediction	Weyl SM.	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0

Table 6: Confusion matrices for accuracy of the small fully connected network presented in *Sec.* 2.3 on the perturbed training and testing dataset of *Sec.* 3.1.

		Training Set Classification (%)			Testing Set Classification (%)		
		Trivial	Topological	Weyl SM.	Trivial	Topological	Weyl SM.
Large Fully	Trivial	94.0 ± 1.3	24.1 ± 4.3	47.7 ± 10.4	95.4 ± 1.4	16.4 ± 4.2	49.3 ± 11.6
Connected	Topological	6.0 ± 1.3	75.9 ± 4.3	52.3 ± 10.4	4.6 ± 1.4	83.6 ± 4.2	50.7 ± 11.6
Prediction	Weyl SM.	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0

Table 7: Confusion matrices for accuracy of the large fully connected network presented in *Sec.* 2.3 on the perturbed training and testing dataset of *Sec.* 3.1.

		Training Set Classification (%)			Testing Set Classification (%)		
		Trivial	Topological	Weyl SM.	Trivial	Topological	Weyl SM.
Novel	Trivial	93.7 ± 1.3	23.7 ± 4.1	46.0 ± 9.2	95.2 ± 1.4	15.6 ± 3.9	47.7 ± 11.2
Convolution	Topological	6.3 ± 1.3	76.3 ± 4.1	54.0 ± 9.2	4.8 ± 1.4	84.4 ± 3.9	52.3 ± 11.2
Prediction	Weyl SM.	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0

Table 8: Confusion matrices for accuracy of the novel convolutional architecture presented in *Sec.* 2.4.1 on the perturbed training and testing dataset of *Sec.* 3.1.

Using this dataset, the accuracies are shown to reduce across all classifications compared to the unperturbed dataset. This is unsurprising due to the increased difficulty of the problem, as the network has to fit non-linear physical boundaries compared to the previous linear unphysical boundaries.

The trace of the testing confusion matrices are 178.6 ± 3.7, 179.0 ± 4.4, and 179.6 ± 4.1 for the small fully connected, large fully connected and novel convolutional networks respectively. Therefore, it suggests that the novel convolutional network is marginally better, but this is not a significant result. In all cases, the Weyl semimetal was entirely misclassified, and the network did not classify a single sample as a Weyl semimetal. This is likely primarily due to the poor choice of the cost function. Given more time, the three aforementioned methods to mitigate poor performance could be implemented.

However, the results showed that performance on the unperturbed dataset is misleading and does not reflect the network's true performance on general crystalline state classification. It is indicative of the network using a probabilistic shortcut to achieve higher accuracy, rather than having a physical understanding of the material. Therefore, in future testing, it is important to use a varied dataset, with as high as possible intrinsic dimension, to ensure an accurate benchmarking of the models is achieved.

Overall, there was some success in correctly classifying the trivial and topological states for a mechanically distorted bismuth-telluride-iodide crystal from their real space orbital overlap data. However, more fruitful results are expected from a universal network operating upon the reciprocal unit-cell. Using the latter form of input, the various forms of medium extractors can be applied with better-expected results. In addition, it would allow a universality to the modelling, where any crystal under any distortion can be inputted, which is not possible using the highly specific real-space hoppings. It is also expected that these architectures will generalise well in predicting any material property associated with the electronic band structure. Due to time constraints, these further applications could not be explored but will be in follow-up work.

4 Conclusion

The primary goal of developing a custom machine learning approach to condensed matter physics was achieved, using a multitude of methods. A novel convolutional design was developed to analyse the orbital overlaps of a BiTeI crystal, to determine its state classification. This approach had success at distinguishing trivial and topological insulators, under a range of mechanical distortions. Performing this classification required a benchmarking dataset to be developed on a well-understood crystalline system, which BiTeI fulfilled. Time-reversal symmetric perturbations were successfully added to the Hamiltonian. This expanded the range of conditions to which this machine learning model could be applied, and some success was seen in the resultant classifications. Although, further work is needed to improve its classification of semimetallic states. A gradient descent method of fine-tuning the distortions applied to crystals to yield useful behaviours, was also outlined.

Moreover, a review of several deep-learning techniques identified and defined several problems which arise in common architectures, including broken equivariance and invariance of convolutional layers, the homogenous and inhomogeneous gradient diffusion problems, four deficits of residual networks and the broken nesting caused by pooling operations. Each of these was isolated and proven to be detrimental to the network trained on a standardised dataset. The resolution of these problems was crucial for progress in a machine learning architecture, custom designed for modelling crystalline systems.

This architecture was shown to be universally applicable to all crystalline systems, by performing analysis on the reciprocal-space such as the electronic band structure of the valence and conductive bands. It is expected that this network will have a good performance in predicting many properties of crystalline systems. Preliminary results also indicated that the spatial invariant forms of this architecture were highly successful, and may be better suited to condensed matter physics than the repurposed computer-vision architectures. These new architectures are very distinct when compared to other approaches found in the literature. Unfortunately, there was insufficient time to both compile a dataset for the reciprocal-space application and train the newly developed machine learning models, due to several key developments arising only in the final stages of the project. Further testing is also needed to determine whether there is widespread applicability of these architectures in the most common domains of machine learning, such as computer vision. We propose that extensive analysis of the models on the ImageNet dataset may be most appropriate.

Overall, this project was a resounding success, with nine problems newly identified, proven and resolved, one novel technique for fine-tuning mechanical distortions applied to crystalline systems, one novel technique of transferring physical symmetries into representational symmetries, a novel technique for analysing spatial invariance, and sixteen new architectures developed purposely for condensed matter physics. It is hoped that this work will generate further avenues of research to explore and experimentally validate, alongside applications of the methods to determine the properties of condensed matter systems with increased accuracy and with the appropriate physical symmetries considered.

References

[1] A. S. Fuhr and B. G. Sumpter, "Deep generative models for materials discovery and machine learning-accelerated innovation," *Frontiers in Materials*, vol. 9, 2022.

[2] C. Yue, S. Jiang, H. Zhu, L. Chen, Q. Sun, and D. W. Zhang, "Device applications of synthetic topological insulator nanostructures," *Electronics*, vol. 7, no. 10, 2018.

[3] M. He, H. Sun, and Q. L. He, "Topological insulator: Spintronics and quantum computations," *Frontiers of Physics*, vol. 14, p. 43401, May 2019.

[4] X. Liu, Y. Si, K. Li, Y. Xu, Z. Zhao, C. Li, Y. Fu, and D. Li, "Exploring sodium storage mechanism of topological insulator bi2te3 nanosheets encapsulated in conductive polymer," *Energy Storage Materials*, vol. 41, pp. 255–263, 2021.

[5] C. Beenakker, "Search for majorana fermions in superconductors," *Annual Review of Condensed Matter Physics*, vol. 4, no. 1, pp. 113–136, 2013.

[6] S.-M. Huang, S.-Y. Xu, I. Belopolski, C.-C. Lee, G. Chang, B. Wang, N. Alidoust, G. Bian, M. Neupane, C. Zhang, S. Jia, A. Bansil, H. Lin, and M. Z. Hasan, "A weyl fermion semimetal with surface fermi arcs in the transition metal monopnictide taas class," *Nature Communications*, vol. 6, p. 7373, Jun 2015.

[7] G. Bird, "Developing a robust material classification dataset for application to machine learning." `https://github.com/GeorgeBird1/AcademicWork/blob/main/Masters%20Thesis%20Semester%201.pdf`, Nov 2022.

[8] K. Ishizaka, M. S. Bahramy, H. Murakawa, M. Sakano, T. Shimojima, T. Sonobe, K. Koizumi, S. Shin, H. Miyahara, A. Kimura, K. Miyamoto, T. Okuda, H. Namatame, M. Taniguchi, R. Arita, N. Nagaosa, K. Kobayashi, Y. Murakami, R. Kumai, Y. Kaneko, Y. Onose, and Y. Tokura, "Giant rashba-type spin splitting in bulk bitei," *Nature Materials*, vol. 10, pp. 521–526, Jul 2011.

[9] J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah, M. Spitzer, and S. Zhao, "Applications of machine learning in drug discovery and development," *Nature Reviews Drug Discovery*, vol. 18, pp. 463–477, Jun 2019.

[10] J. Carrasquilla and R. G. Melko, "Machine learning phases of matter," *Nature Physics*, vol. 13, pp. 431–434, May 2017.

[11] Y. Zhang and E.-A. Kim, "Quantum loop topography for machine learning," *Physical Review Letters*, vol. 118, may 2017.

[12] T. Konno, H. Kurokawa, F. Nabeshima, Y. Sakishita, R. Ogawa, I. Hosako, and A. Maeda, "Deep learning model for finding new superconductors," *Physical Review B*, vol. 103, no. 1, p. 014509, 2021.

[13] N. Sun, J. Yi, P. Zhang, H. Shen, and H. Zhai, "Deep learning topological invariants of band insulators," *Physical Review B*, vol. 98, aug 2018.

[14] J. F. Rodriguez-Nieva and M. S. Scheurer, "Identifying topological order through unsupervised machine learning," *Nature Physics*, vol. 15, pp. 790–795, Aug 2019.

[15] J. Ding, H.-K. Tang, and W. C. Yu, "Rapid detection of phase transitions from monte carlo samples before equilibrium," *SciPost Physics*, vol. 13, sep 2022.

[16] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010.

[17] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[18] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[20] S. Murakami, "Phase transition between the quantum spin hall and insulator phases in 3d: emergence of a topological gapless phase," *New Journal of Physics*, vol. 9, pp. 356–356, sep 2007.

[21] S. Murakami and S. ichi Kuga, "Universal phase diagrams for the quantum spin hall systems," *Physical Review B*, vol. 78, oct 2008.

[22] X. Ying, "An overview of overfitting and its solutions," in *Journal of physics: Conference series*, vol. 1168, p. 022022, IOP Publishing, 2019.

[23] E. W. Weisstein, "n-torus."

[24] P. Hohenberg and W. Kohn, "Inhomogeneous electron gas," *Phys. Rev.*, vol. 136, pp. B864–B871, Nov 1964.

[25] T. Mikolov *et al.*, "Statistical language models based on neural networks," *Presentation at Google, Mountain View, 2nd April*, vol. 80, no. 26, 2012.

[26] G. Bird, "Bird's convention for diagramatic neural networks." `https://github.com/GeorgeBird1/Diagramatic-Neural-Networks`, Nov 2022.

[27] R. E. Bellman, *Dynamic programming*. Princeton university press, 2010.

[28] K. Koutroumbas and S. Theodoridis, *Pattern recognition*. Academic Press, 2008.

[29] D. Harris and S. L. Harris, *Digital design and computer architecture*. Morgan Kaufmann, 2010.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[31] J. Li and D. Liu, "Information bottleneck theory on convolutional neural networks," 2021.

[32] A. Einstein, "The Foundation of the General Theory of Relativity," *Annalen Phys.*, vol. 49, no. 7, pp. 769–822, 1916.

[33] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, 2016.

[34] H. Gholamalinezhad and H. Khosravi, "Pooling methods in deep neural networks, a review," 2020.

[35] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, vol. 5, pp. 157–66, 02 1994.

[36] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," 2013.

[37] A. L. Maas, A. Y. Hannun, A. Y. Ng, *et al.*, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, p. 3, Atlanta, Georgia, USA, 2013.

[38] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

[39] "Pytorch implementation of tensor rolling." `https://pytorch.org/docs/stable/generated/torch.roll.html`.

[40] J. Wang, S. Valligatla, Y. Yin, L. Schwarz, M. Medina-Sánchez, S. Baunack, C. H. Lee, R. Thomale, S. Li, V. M. Fomin, L. Ma, and O. G. Schmidt, "Experimental observation of berry phases in optical möbius-strip microcavities," *Nature Photonics*, vol. 17, pp. 120–125, Jan 2023.

[41] G. Liu, M. Pi, L. Zhou, Z. Liu, X. Shen, X. Ye, S. Qin, X. Mi, X. Chen, L. Zhao, B. Zhou, J. Guo, X. Yu, Y. Chai, H. Weng, and Y. Long, "Physical realization of topological roman surface by spin-induced ferroelectric polarization in cubic lattice," *Nature Communications*, vol. 13, p. 2373, May 2022.

[42] M. V. Berry, "Quantal phase factors accompanying adiabatic changes," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 392, no. 1802, pp. 45–57, 1984.

[43] S. Pancharatnam, "Generalized theory of interference, and its applications," *Proceedings of the Indian Academy of Sciences - Section A*, vol. 44, pp. 247–262, Nov 1956.

[44] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "8.6 residual networks - dive into deep learning," *arXiv preprint arXiv:2106.11342*, 2021.

[45] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *CoRR*, vol. abs/1608.06993, 2016.

[46] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014.

[47] M. Lin, Q. Chen, and S. Yan, "Network in network," 2014.

[48] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, p. 1735–1780, nov 1997.

[49] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *Proceedings of the International Workshop on Artificial Neural Networks: From Natural to Artificial Neural Computation*, IWANN '96, (Berlin, Heidelberg), p. 195–201, Springer-Verlag, 1995.

[50] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing: Algorithms, architectures and applications*, pp. 227–236, Springer, 1990.

[51] V. Klema and A. Laub, "The singular value decomposition: Its computation and some applications," *IEEE Transactions on automatic control*, vol. 25, no. 2, pp. 164–176, 1980.

[52] G. E. Hinton and S. Roweis, "Stochastic neighbor embedding," *Advances in neural information processing systems*, vol. 15, 2002.

[53] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[54] D. Ballard, "Modular learning in neural networks," 1987.

[55] I. J. Good, "Rational decisions," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 14, no. 1, pp. 107–114, 1952.

[56] "Pytorch's implementation of conv2d." `https://pytorch.org/docs/stable/generated/torch.nn.Conv2d.html`.

Appendices

A Neural Architectures

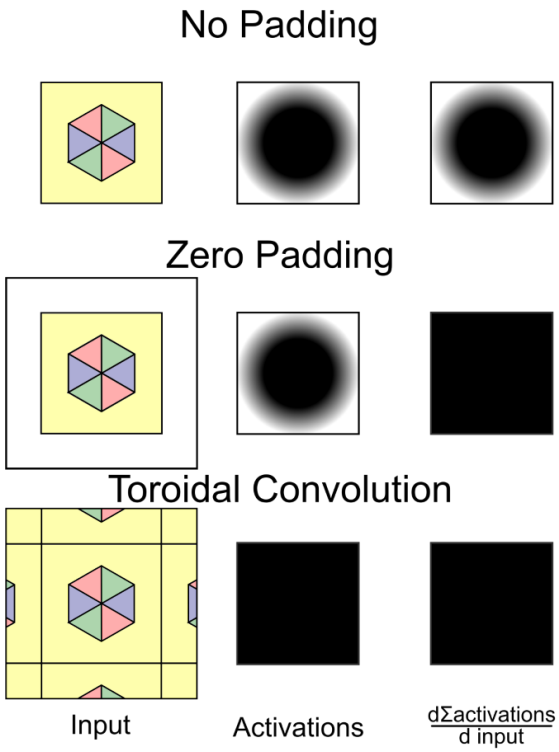

Figure 23: This figure shows how no-padding in convolution results in edge effects for both the activations and the differential between the sum of the activations with respect to the input, thus disrupting equivariance. Zero-padding alleviates the latter, by ensuring that all input elements contribute evenly to the sum of the activations, resulting in an equivariant backpropagation step. Toroidal convolution fixes both problems, with no edge effects in either the activations or the derivative of the sum of the activations. Therefore, toroidal convolution has a full global equivariance. The darkness of the activations indicates the expected generalised magnitude of activations. The darkness of the derivative column indicates the general contribution of each input pixel to the sum of the activations.

B BiTeI Dataset

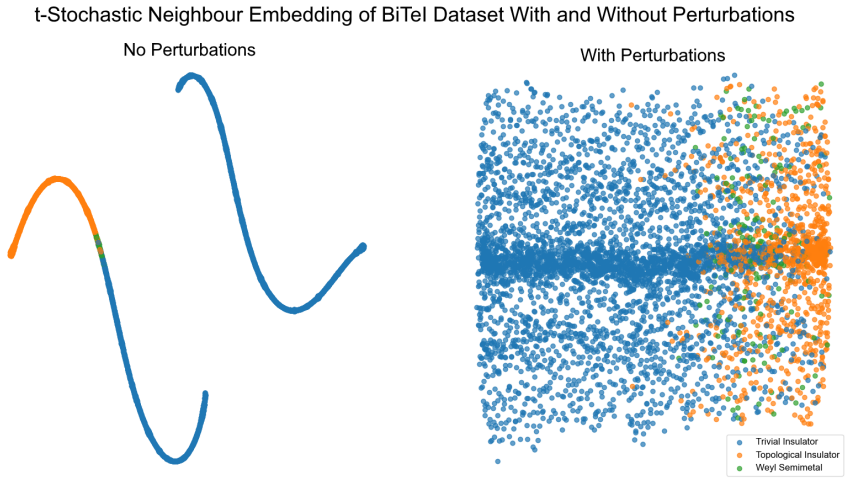

Figure 24: Shows a t-stochastic neighbour embedding of Hamiltonian samples in the unperturbed (left) and perturbed (right) datasets. The curved shape and discontinuity of the line in the left image are a result of the embedding algorithm. It can be seen that the unperturbed dataset is an embedding of a one-dimensional manifold, whereas the perturbed dataset demarcates a high-dimensional manifold's volume. The colour of the sample indicates the crystalline state.

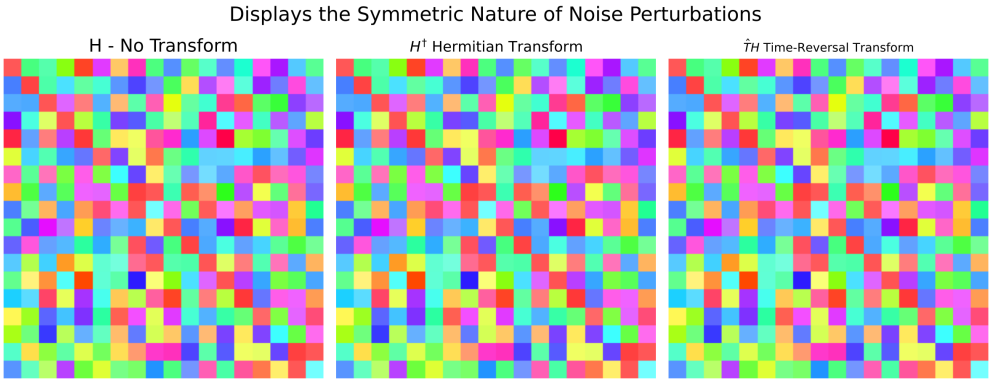

Figure 25: Shows the hopping perturbations for $\vec{A} = \vec{0}$, where the colour indicates the complex argument and the opacity indicates the absolute value. It can be seen that the hoppings are unchanged under the hermitian and time-reversal symmetry as desired. This has also been confirmed for all values of $\vec{A}$.

C MNIST

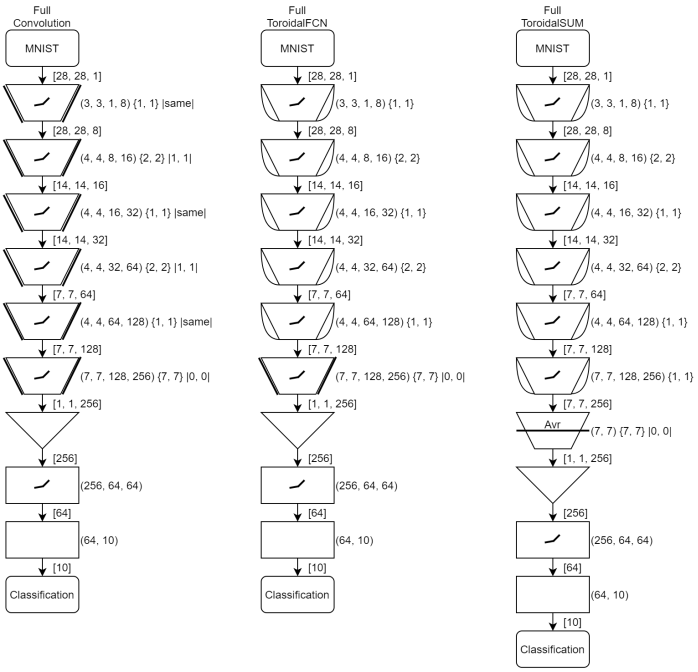

Figure 26: Shows the architectures for the "Full Convolution" (left), "Full Toroidal FCN" (middle) and "Full ToroidalSUM" (right). The Toroidal FCN network has an invariance-breaking final convolutional layer. All networks have equal numbers of parameters. |same| indicates the "SAME" padding [56].

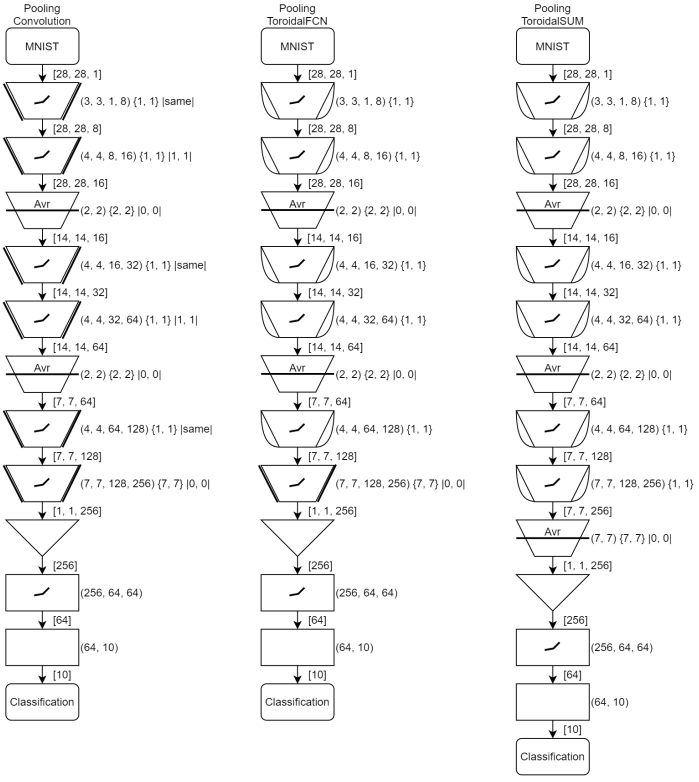

Figure 27: Shows the architectures for the "Pooling Convolution" (left), "Pooling Toroidal FCN" (middle) and "Pooling ToroidalSUM" (right).

D CIFAR10

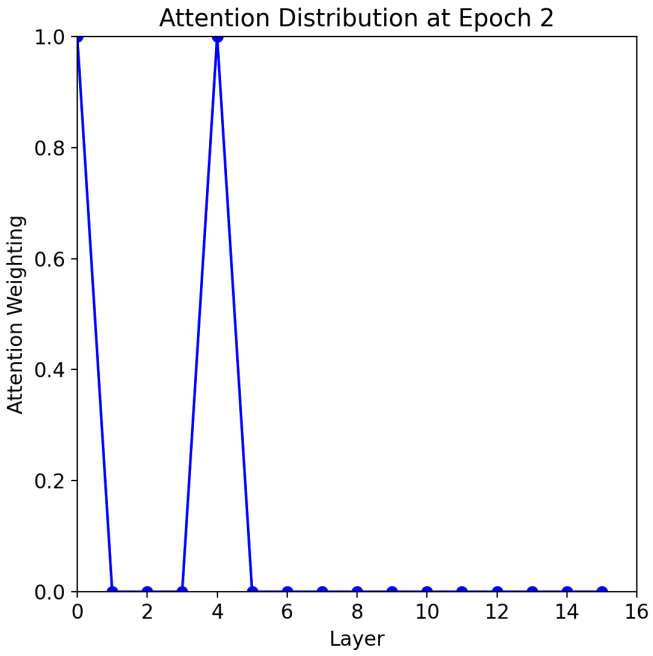

Figure 28: This figure indicates the attention distribution at epoch=2 for a network exhibiting unstable behaviour. This attention distribution shows a saturation on layers zero and four, with no dynamic behaviour between samples shown by the standard deviation of zero. This indicates that the network is unaware of the value of other layers since they are damped to zero. This in turn limits the performance. This attention distribution was found to remain constant across all future epochs.

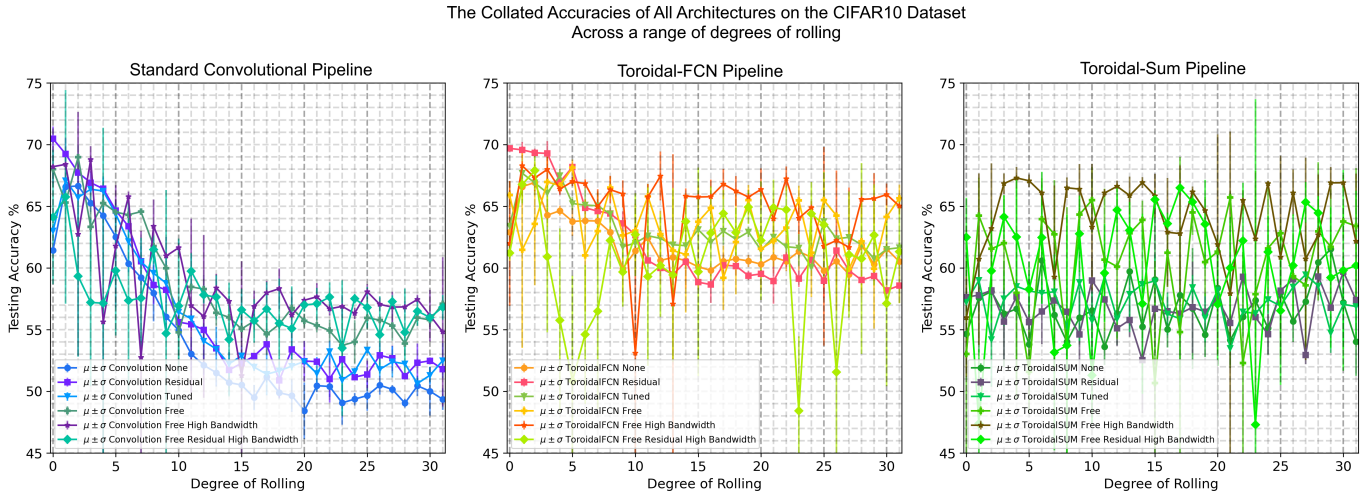

Figure 29: Shows the combined accuracies, on CIFAR10, of all tested networks against the degree-of-rolling χ, with results separated into columns of the particular pipeline architectures. Standard convolution is shown (left), Toroidal-FCN (centre) and Toroidal-Sum (right).