Learning Chern Numbers of Topological Insulators With Gauge Equivariant Neural Networks

Master's thesis in Engineering Mathematics and Computational Science

LONGDE HUANG

DEPARTMENT OF MATHEMATICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Learning Chern Numbers of Topological Insulators with Gauge Equivariant Neural Networks

LONGDE HUANG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Division of Algebra and Geometry
Chalmers University of Technology
Gothenburg, Sweden 2025

Learning Chern Numbers of Topological Insulators with Gauge Equivariant Neural Networks
LONGDE HUANG

Supervisor: Jan Gerken, Department of Mathematical Sciences
Examiner: Jan Gerken, Department of Mathematical Sciences

Master's Thesis 2025
Department of Mathematical Sciences
Division of Algebra and Geometry
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 79 357 8503

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Learning Chern Numbers of Topological Insulators with Gauge Equivariant Neural Networks

LONGDE HUANG
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

Equivariant network architectures are a well-established tool for predicting invariant or equivariant quantities. However, almost all learning problems considered in this context feature a global symmetry, i.e. each point of the underlying space is transformed with the same group element, as opposed to a local "gauge" symmetry, where each point is transformed with a different group element, exponentially enlarging the size of the symmetry group. Gauge equivariant networks have so far mainly been applied to problems in quantum chromodynamics. Here, we introduce a novel application domain for gauge-equivariant networks in the theory of topological condensed matter physics. We use gauge equivariant networks to predict topological invariants (Chern numbers) of multiband topological insulators. The gauge symmetry of the network guarantees that the predicted quantity is a topological invariant. We introduce a novel gauge equivariant normalization layer to stabilize the training and prove a universal approximation theorem for our setup. We train on samples with trivial Chern number only but show that our models generalize to samples with non-trivial Chern number. We provide various ablations of our setup. Our code is available at `https://github.com/sitronsea/GENet/tree/main`.

Keywords: Geometric Deep Learning, Gauge Equivariant Networks, Condensed-Matter Physics.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Jan E. Gerken, who provided insightful guidance and steady support throughout the research journey. His expertise and professionalism helped me overcome the challenges of this project, and his commitment to both the research and my embarking on the academic path has been invaluable.

This thesis is based on a collaborative research project within our group, which resulted in a preprint currently available on arXiv under the same title. Some of the material in this thesis includes contributions from my co-authors, Oleksandr Balabanov, Hampus Linander, Mats Granath, Daniel Persson, and Jan E. Gerken, to whom I extend my sincere appreciation for their knowledge and collaboration, particularly in shaping the research framework and conducting experiments.

Lastly, I would like to thank my family, especially my parents, for their unconditional love and encouragement throughout my life. My heartfelt gratitude also goes to my partner, Xiaolingzi Hu, whose love, companionship, and trust have been a constant source of motivation and strength. Finally, I deeply appreciate all my friends, particularly Wenjun Huang, Tianyi Gu, and Chenyu Zhang, whose emotional support and valuable advice have accompanied me along the way, despite the great distance separating us.

This work has benefited from computations enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

Longde Huang, Gothenburg, May 2025

Contents

1

Introduction

1.1 Overview

The mathematical fields of geometry and topology provide a rigorous language for describing the shape of data, capturing information from local properties such as curvature and dimensionality to global characteristics such as connectedness or genus. In practical machine learning tasks, classical methodology of treating samples as vectors in a flat Euclidean space tends to break down as the data grows in complexity[1]. Researchers therefore move to analyze them as a complex subspace, and take advantage of its topological and geometric features to conduct more efficient classifications and studies.

Geometric deep learning, a subfield of machine learning emerge to this end. It utilizes the geometric and topological in complex data to construct more efficient neural network architectures[2]. This approach has been successfully applied in a variety of domains, from medical imaging [3] to high-energy physics [4] and quantum chemistry [5]. In particular, machine learning for quantum physics has grown explosively over the last decade, with applications in condensed matter physics, materials science, quantum information and quantum computing [6, 7, 8, 9]. Within this broad area, we focus on machine learning of topological states of matter.

Topology, a branch of mathematics, focuses on the characteristics of objects that are preserved under a continuous transformation. For instance, a donut is topologically equivalent to a coffee cup, but not to a ball, based on the topological characteristic of genus (the number of holes). Topological insulators are materials whose interior is insulating yet whose surface or boundary supports robust conduction. This behavior arises from the different topologies of the electronic band structures inside and outside of the material. The topologies are characterized by quantities which do not change under continuous deformations respecting the underlying system's physical symmetries; these are known as topological invariants.

The study of topological insulators, a class of materials which has been one of the

main areas of interest in condensed matter physics over the last two decades [10, 11], with a broad range of applications, including spintronics and magnetoelectronics [12], photonics [13], quantum devices [14], and quantum computing [15, 16, 17], and there has been a lot of attempts to explore it with deep neural networks. Early work includes Carrasquilla & Melko [18], who used supervised learning on small CNNs for the Ising lattice gauge theory, and van Nieuwenburg et al. [19], who developed an unsupervised "learning by confusion" method to study phase transitions including topological order in the Kitaev chain. Other unsupervised approaches employ diffusion maps [20] and topology-preserving data generation [21]. Of particular relevance are studies that used CNNs and supervised learning to predict U(1) topological invariants [22, 23]; subsequent work extended this to unsupervised settings with topology-preserving data augmentations [24, 25].

Previous studies on topological insulators are limited to single band insulators. In contrast, our focus will be on multi-band topological insulators, in which several wave functions are combined into a vector. Previous works could construct deep learning systems which predict Chern numbers for materials with only one filled band [23, 24]. Perhaps surprisingly, these models fail to learn Chern numbers for higher-band systems, pointing to a fundamental challenge in the multi-band regime. In contrast, our model is able to predict Chern numbers of materials with at least seven filled bands.

We identify the gauge symmetry of the system as the central reason for the failure of traditional approaches in the high-band setting. Over a discretized lattice, the physical feature of the material could be represented with a collection of unitary matrices at each lattice site. Usual group equivariant networks $\mathcal{N} : X \to Y$ satisfy the constraint $\mathcal{N}(\rho_X(g)x) = \rho_Y(g)\mathcal{N}(x)\ \forall g \in G$ with symmetry group G and representations $\rho_{X,Y}$ on the input- and output spaces, respectively. In contrast, a gauge equivariant network satisfies a much stronger condition in which the group element g can depend on the input x: $\mathcal{N}(\rho_X(g_1)x_1, \ldots, \rho_X(g_n)x_n) = (\rho_Y(g_1)y_1, \ldots, \rho_Y(g_n)y_n)^\top$. In our case, the input space X is the Fourier transform of the position, the so-called Brillouin zone of the material, and $G = \mathrm{U}(N)$ for a system with N filled bands. In this context, Chern number is invariant under any local transformation via $U(N)$-group action

$$W_x \sim \Omega_x^\dagger W_x \Omega_x$$

Where N is the number of filled bands that share the same eigenvalue. When $N > 1$, the transformation becomes non-abelian (non-commutative), and this mathematical difficulty implies the need of a tailored structure for this task.

Group equivariant neural networks for continuous (Lie) groups have been widely explored. Classical methodology includes the work of Finzi et al. [26], which introduced an architecture that preserves Lie group equivariance by directly solving the constraints of the underlying Lie algebra, and Wang et al. [27], which constructed an equivariance relaxation mechanism for 2D symmetry. More specifically, orthogonal group equivariance, which is a real-valued counterpart of the unitary group equivariance considered in our work, has also been tackled, such as the architecture

introduced by Lim et al. [28], which directly operates on the eigenvalues of matrices, and Lawrence et al. [29], which generalized the method of group convolution with a uniform sampling over the group. However, they are not applicable for the site-dependent $U(N)$ transformations, as the gauge group's exponential growth renders direct application infeasible.

We instead propose to use a gauge-equivariant network for learning topological invariants such as the Chern number. Gauge equivariant networks have been studied in two main settings. First, the gauge symmetry concerns local coordinate changes in the domain of the feature maps [1, 30, 2]; models respecting this symmetry were introduced in [31, 32]. Second, relevant for lattice quantum chromodynamics (QCD), the gauge transformations act on the co-domain of the feature maps. Applications in lattice QCD include gauge-equivariant normalizing flows [33, 34, 35, 36, 37] and neural-network quantum states [38]. In contrast, our model builds on a gauge-equivariant prediction network developed for lattice QCD [39]. This is a novel application for gauge equivariant neural networks. In particular, we cast the problem at hand in a form in which we can use an adapted version of the Lattice Gauge Equivariant Convolutional Neural Networks (LGE-CNNs) [39] to learn multiband Chern numbers.

1.2 Summary of Results

In this work, we develop GEBLNet, a U(N)-gauge–equivariant neural network architecture for predicting Chern numbers of multiband topological insulators. The model operates locally on Wilson loops, is trained end-to-end with a mixed global and standard deviation loss, and is theoretically guaranteed to learn any gauge-invariant topological characteristics. Meanwhile, we design tailored metrics to evaluate prediction accuracy, robustness, and scalability. Our investigation reveals several remarkable properties of our network in the context of learning:

Model level results.

(1) A stack of Gauge-Equivariant Bilinear Layers (GEBL) equipped with our novel trace normalization layer yields stable training even for 7-band systems, where a conventional ResNets diverge.

(2) A purely local kernel (convolution size 0) is sufficient; adding spatial kernels seldom improves and may degrade accuracy, illustrating the dominance of in-site information over inter-site coupling.

(3) Our model could be generalized to multiband systems with a higher spatial dimensions, and have the potential to learn higher order Chern numbers in a significantly more complicated setting.

Theory–level results

(1) We prove a universal approximation theorem showing that GEBL stacks can approximate all compact group invariant functions to arbitrary precision.

(2) Utilizing the spectral-orbit decomposition $\mathrm{U}(N)/\mathrm{Ad} \simeq \mathrm{U}(1)^N/S_N$, we derive an efficient diagonal-orbit sampler that simplify training data generation, while preserving a uniform distribution on equivalence classes.

Empirical results

(1) On 5×5 lattices with $N = 4$–7 filled bands, GENet attains an accuracy between 93–96%, surpassing deep MLP/ResNet baselines by $> 60\%$.

(2) Accuracy degrades linearly by only $\approx 3\%$ when evaluating on 10×10 lattices, demonstrating strong size extrapolation.

(3) GENet extends to higher order Chern numbers with a mean absolute error of 0.25, well within the rounding threshold used in physics.

These results establish gauge-equivariant neural networks as an efficient, theoretically sound, and highly generalizable tool for exploring complex topological phases of matter.

1.3 Outline

The structure of this thesis is as follows. Chapter 2 reviews the physical background required to understand the problem, starting from the fundamentals of quantum mechanics and gradually introducing the concept of Chern numbers. It also presents the mathematical tools concerning unitary groups and class functions, which play a central role in both our data generation scheme and the theoretical analysis of model approximation. Chapter 3 describes the proposed network architecture, including the gauge-equivariant building blocks and a universal approximation theorem tailored to our setup. Chapter 4 provides an in-depth empirical study of the model's behavior and performance. In particular, it includes an ablation study comparing our model with other baselines, explores generalization from trivial to non-trivial samples and from small to large grids, and extends the method to higher-dimensional topological insulators. We conclude in Chapter 5 with a summary of our findings, current limitations, and directions for future research.

2

Background

In this section, we introduce the concept of multiband insulators through building from the basic formulation of quantum mechanics.

2.1 Quantum Mechanics and Hamiltonians

Quantum mechanics provides a fundamental framework for the description of microscopic particle behaviours. We represent the position, momentum, and other information of particles with a "state", with all possible states forming an infinite-dimensional complex Hilbert space. For each observable physical quantity, we associate with it a Hermitian operator. The eigenvectors of the operator are called eigenstates, and the corresponding spectra are all possible outcomes from measuring. The evolution of a quantum state is governed by the Schrödinger equation

$$i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle = H|\psi(t)\rangle, \tag{2.1}$$

where $\hbar$ is the reduced Planck's constant, $|\psi(t)\rangle$ is the state vector, and H is the Hamiltonian operator, corresponding to the total energy of the system.

Due to the lattice crystalline structure of the atoms, the potential $V(\mathbf{r})$ is periodic along each basis of the lattice,

$$V(\mathbf{r}) = V(\mathbf{r} + \mathbf{R})$$

By Bloch's theorem, eigenstates of an electron in such a periodic potential can be written in the form

$$\phi(\mathbf{r}) = e^{i\mathbf{p}\cdot\mathbf{r}}u(\mathbf{r})$$

where $\mathbf{p}$ is the momentum, $u(\mathbf{r})$ shares the same periodicity with the lattice. In the same merit of classical Hamiltonian mechanics, we could perform a Fourier transformation from the position space to the momentum space

$$\psi(\mathbf{p}) = \frac{1}{(2\pi\hbar)^{\frac{n}{2}}}\int d^n r e^{i\mathbf{p}\cdot\mathbf{r}/\hbar}\psi(\mathbf{r}),$$

and consider the primitive cell in the reciprocal momentum lattice, defined as the Brillouin zone. In this thesis, it is topologically equivalent to a n dimensional torus T^n, whose dimension is the same as the lattice.

Multiband insulators are materials in which multiple electronic bands are involved in the insulating behavior. By electronic band, we mean an eigenstate of the Hamiltonian, namely an energy level for the electrons to dwell in. In this scenario, since we restrict to a finite number of relevant bands or eigenstates, the Hamiltonian is essentially reduced to a finite-dimensional operator, namely a complex Hermitian matrix. Therefore, we can formulate the Hamiltonian as a map from the Brillouin zone to Hermitian matrices.

2.2 Gauge Symmetry and Berry Curvature

By the previous formulation, the Hamiltonian is a map from the base manifold T^n (Brillouin Zone) to the space of Hermitian matrices. Thus for any given $p \in U_\alpha \subset T^n$, we could locally diagonalize $H(p)$ into $U_\alpha(p)^\dagger \Lambda_\alpha(p) U_\alpha(p)$. where $U_\alpha(p)$ represents the eigenstates, and $\Lambda_\alpha(p)$ represents the eigenvalues (all possible energy levels).

From now on, we make the assumption that $\Lambda(p)$ has the form $\mathrm{diag}\{E_+ I_{m+}, E_- I_{m-}\}$, i.e. a positive eigenvalue with m_+ multiplicity and a negative eigenvalue with m_- multiplicity. Physically, this means the insulator has m_- bands filled with electrons, whose energy levels are equal, and m_+ empty bands whose energy levels are also equal, but are too high for the electrons to leap into.

This allows a symmetry, or equivalently, an invariance under a group action: for any set of linearly independent eigenstates with negative eigenvalues $U^-(p)$, $\Omega(p)U^-(p)$ gives another set of eigenstates, where $\Omega(p)$ is unitary. For simplicity, we denote m_- as m, and $\mathrm{U}(m)$ is defined as the gauge group.

The Berry connection is defined as a 1-form A such that

$$\forall X \in T_p M,\ X_p A(p) = U^\dagger(p) X_p U(p)$$

With an explicit coordinate system, the Berry connection can be written as

$$A(p) = \sum A^i(p) = U^\dagger(p) \partial_i U(p) \mathrm{d}x^i.$$

Note that due to the non-uniqueness of $U(p)$, the Berry connection is not well-defined globally. For instance, if $U_\beta(p) = U_\alpha(p)\Omega_{\alpha\beta}(p)$, then

$$A_\beta = \Omega_{\alpha\beta}^\dagger A_\alpha \Omega_{\alpha\beta} + \Omega_{\alpha\beta}^\dagger \mathrm{d}\Omega_{\alpha\beta}$$

However, if we consider $F_\alpha = \mathrm{d}A_\alpha + A_\alpha \wedge A_\alpha$, we see that these local differential forms are actually compatible, up to conjugation. We note $\Omega_{\alpha\beta}$ as g, A_α as A, $g^\dagger \mathrm{d}g$

as X for now.

$$\begin{aligned} \mathrm{d}(A_\beta) =& \mathrm{d}(g^\dagger A g + X) \\ =& \mathrm{d}g^\dagger \wedge Ag + g^\dagger \mathrm{d}Ag + g^\dagger A \wedge \mathrm{d}g + \mathrm{d}X \\ =& - X \wedge g^\dagger Ag + g^\dagger dAg + g^\dagger Ag \wedge X + \mathrm{d}X \\ A_\beta \wedge A_\beta =& g^\dagger A \wedge Ag + g^\dagger Ag \wedge X + X \wedge g^\dagger Ag + X \wedge X \\ \Rightarrow F_\beta =& g^\dagger F_\alpha g \end{aligned} \tag{2.2}$$

We define F to be the Berry curvature.

2.3 Chern numbers

We now assume the base manifold (Brillouin zone) to be two dimensional, and consider the determinant bundle associated to the current $\mathrm{U}(n)$-principle bundle E, namely the line bundle of filled band eigenstates. This is defined by taking the exterior algebra $\bigwedge^m E = \{\psi_1(p) \wedge \psi_2(p) \wedge \cdots \wedge \psi_m(p) = \wedge^m \psi_i(p)\}$. Then the new curvature A^{det} induced by Berry curvature has the following property

$$A^{\mathrm{det}}(\wedge^m \psi_i(p)) = \sum_i \psi_1(p) \wedge \cdots \wedge A\psi_i(p) \wedge \cdots \wedge \psi_m(p) \tag{2.3}$$

$$= \sum_{i,j} \psi_i(p) \wedge \cdots \wedge A^{ij}(p)\psi_j(p) \wedge \cdots \wedge \psi_m(p) \tag{2.4}$$

$$= \mathrm{Tr}A \wedge^m \psi_i(p) \tag{2.5}$$

In other words, the induced (local) one form has the simple form of $\mathrm{Tr}A$. Notice that $\mathrm{d}A^{\mathrm{det}} = \mathrm{Tr}F$ is globally defined; furthermore, $\mathrm{Tr}F = \mathrm{Tr}(\mathrm{d}A + A \wedge A) = \mathrm{d}\mathrm{Tr}A = \mathrm{d}A^{\mathrm{det}}$. We now show the correlation between $\mathrm{Tr}F$ and Čech cohomology.

Recall that $A_\beta = \Omega^\dagger_{\alpha\beta} A_\alpha \Omega_{\alpha\beta} + \Omega^\dagger_{\alpha\beta} \mathrm{d}\Omega_{\alpha\beta}$, therefore $\omega_{\beta,\alpha} = \mathrm{Tr}A_\beta - \mathrm{Tr}A_\alpha = \mathrm{Tr}\Omega^\dagger_{\alpha\beta} d\Omega_{\alpha\beta}$. Since $\Omega_{\alpha\beta} \in U(n)$, we have

$$\omega_{\beta,\alpha} = \mathrm{Trd}\log\Omega_{\alpha\beta} = \mathrm{d}\log\det\Omega_{\alpha\beta} := \mathrm{d}\phi_{\alpha\beta} \tag{2.6}$$

On the other hand, we know that the gauge transformation on overlapping covers satisfies $\Omega_{\alpha\beta}\Omega_{\beta\gamma}\Omega_{\gamma\alpha} = I$, which implies $\omega_{\beta,\alpha} + \omega_{\alpha,\gamma} + \omega_{\gamma,\beta} = \mathrm{d}(2\pi i n)$, $n \in \mathbb{Z}$.

In other words, $\{\phi_{\alpha\beta}\}$ forms a 2-cocycle. With the isomorphism between Čech cohomology and de Rham cohomology, we conclude that $[\mathrm{Tr}F/2\pi i]$ is exactly the image of $[\{\phi_{\alpha,\beta}\}]$, from $\check{\mathrm{H}}^2(T^n, \mathcal{F}) \to \mathrm{H}^2_{\mathrm{dR}}(T^n, \mathbb{R})$. Specifically, if we assume $n = 2$ and a unit volume for the Brillouin zone, we have

$$c_1 := \int_{BZ} \frac{\mathrm{Tr}F}{2\pi i} \in \mathbb{Z} \tag{2.7}$$

We define c_1 as the first Chern number.

A nontrivial Chern number means that it is impossible to find smoothly varying eigenvectors over the entire Brillouin zone. There are generalizations to higher dimensional Brillouin zones, on which we performed experiments, see Section 4.5.

2.4 Class Functions

As defined in (2.8), the discretized Chern number is invariant under adjoint actions of unitary matrices. In particular, the local quantity $f(g) = \mathrm{ImTr}\log g$ is a so-called class function on the gauge group $U(N)$.

Definition 1. A (complex) class function on a (compact) Lie group G is a function $F : G \to \mathbb{C}$ such that

$$F(h^{-1}gh) = F(g),\ \forall g, h \in G$$

We denote the space of square integrable class functions over G as $L^2_{class}(G)$. It is a closed subspace of $L^2(G)$, hence a Hilbert space itself with the induced inner product.

The orbit set of fluxes under the gauge group action is $U(1)^N/_{S_N}$, that is, the unordered tuple of eigenvalues. Consequently, class functions can only depend on those eigenvalues. The following theorem [40] formalized this intuition and characterized the structure of square integrable class functions.

Theorem 1 (Peter-Weyl)**.** *Let $L^2_{class}(G)$ denote all class functions on G, then all irreducible representations of G are finite-dimensional, and*

$$L^2_{class}(G) = span\{\chi_\rho\}$$

Where χ_ρ represents the character of the irreducible representation ρ.

For a finite dimensional irreducible representation ρ, its character is a symmetrical polynomial, when restricted to its maximal torus, which, for the unitary group $U(N)$, is isomorphic to the set of $U(1)^N$, in other words, the ordered tuple of eigenvalues. These polynomials can be expressed as linear combinations of Schur polynomials.

Proposition 2. *Square-integrable class functions can be expressed as linear combinations of symmetric polynomials, namely*

$$\forall f \in L^2_{class}(G),\ f|_{T^N}(\lambda_1, \lambda_2, \cdots, \lambda_N) = \sum_{k=0}^{\infty} c_k e_k,$$

where

$$e_k = \sum_{k_i \geq 0, \sum k_i = k} \prod_i \lambda_i^{k_i}, c_k \in \mathbb{C}$$

Furthermore, the standard Newton's identities state that

Proposition 3. *For symmetric polynomials, we have*

$$ke_k = \sum_{t=0}^{k} (-1)^{t-1} e_{k-t} p_t.$$

Where $p_t(z_1, \cdots, z_N) = \sum z_i^t$ is the t_{th} equal power polynomial.

Therefore,

Proposition 4. *$\forall f \in L^2_{class}(G)$, f could be approximated by a series of functions $f_n(p_1, \cdots, p_n)$ on the first n equal power polynomials.*

Note that for any matrix W, p_n is essentially $\mathrm{Tr}W^n$. This proposition plays a crucial part in the proof of the universal approximation theorem for our model, presented in Section 3.3.

2.5 Approximation in Discretized Brillouin Zones

In practice, we consider a discretization of the Brillouin zone into a rectangular grid with periodic boundary conditions, of which there are a total of $N_x \times N_y = N_{\text{site}}$ grid points. On this grid, one can define the following discrete, integer Chern number $\tilde{C}$ which converges to C for vanishing grid spacing [41]

$$\tilde{C} = \sum_{(i,j)} \mathrm{Im}\,\mathrm{Tr}\,\log\, W_{i,j}\,, \tag{2.8}$$

where $W_{i,j} \in \mathrm{U}(N)$ is the Wilson loop at grid point $\vec{k} = (i, j)$, defined by

$$W_k = W_{i,j} = U^x_{i,j} U^y_{i-1,j} U^x_{i-1,j-1} U^y_{i,j-1} \tag{2.9}$$

in terms of the link matrices $U^x_{i,j}, U^y_{i,j} \in \mathbb{C}^{N\times N}$. These links capture the overlap between the eigenvectors $\{v_n(k) : k \in BZ\}_{n=1}^N$ of the Bloch Hamiltonians of neighboring grid points and have components

$$[U^x_{i,j}]_{m,n} = v_m(k_{i,j})^\top v_n(k_{i-1,j}) \tag{2.10}$$

$$[U^y_{i,j}]_{m,n} = v_m(k_{i,j})^\top v_n(k_{i,j-1})\,. \tag{2.11}$$

The links are discrete analogs of the operator $\exp(i\mathcal{A}(k)dk)$. The Wilson loops correspond to closed 1×1 loops of the link variables. In higher dimensions, there are several Wilson loops W^γ_k per grid point k that are aligned with different directions γ in the lattice.

The learning task we will study in this article is to predict the discrete Chern number $\tilde{C} \in \mathbb{Z}$ given the Wilson loops $W_{i,j} \in \mathbb{C}^{N\times N}$ on the lattice. One of our models also uses the links as additional input. Although the Chern number is given by the innocuous-looking equation (2.8), learning it is not straightforward as detailed in Section 4.1. We will show that the main reason for the difficulty of learning (2.8) lies in its gauge invariance of $\tilde{C}$.

All topological indices, including the Chern number, are invariant under the gauge group $\mathrm{U}(N)$. Gauge transformations are local symmetries, i.e. at each lattice point i, j they act with a different group element $\Omega_{i,j} \in \mathrm{U}(N)$ by

$$U^{x,y}_{i,j} \to (\Omega_{i,j})^\dagger U^{x,y}_{i,j} \Omega_{i+1,j} \tag{2.12}$$

$$W^\gamma_{i,j} \to \Omega^\dagger_{i,j} W^\gamma_{i,j} \Omega_{i,j}\,, \tag{2.13}$$

where † denotes Hermitian conjugation. Hence, the total symmetry group is $U(N)^{N_{\text{site}}}$.

The gauge symmetry of the system implies the equivalence relation

$$W \sim \Omega^\dagger W \Omega \quad \forall\, \Omega \in \mathrm{U}(N) \tag{2.14}$$

on the set of Wilson loops at each grid point. According to the spectral theorem, there is exactly one diagonal matrix in each equivalence class (group orbit) with elements in U(1), up to reordering of the diagonal elements. Therefore, the set of equivalence classes for this relation is given by $\mathrm{U}(1)^N/_{S_N}$, the set of diagonal unitary $N \times N$-matrices up to permutation of the diagonal elements. This fact has been used in the construction of gauge equivariant spectral flows [34]. We will exploit it to simplify the data augmentation process.

2.6 Higher order Chern numbers

We hereby extend the definition of Chern numbers for two-dimensional Brillouin zones to general Brillouin zones with a spatial dimension of $2n$.

For two dimensional Brillouin zones, the linear space of two dimensional $U(n)$ -valued differential forms $\Omega^2(M, U(n))$. On the other hand, for $2n$ dimensional Brillouin zones, which are topologically equivalent to $T^{2n} = \mathbb{R}^{2n}/_{\mathbb{Z}^{2n}}$, there are $\binom{2n}{2}$ different oriented planes, i.e. for every two directions k_μ, k_ν, there is a planar curvature

$$F^{\mu,\nu} = \partial_{k_\mu} A^\nu(k) - \partial_{k_\nu} A^\mu(k) + [A^\mu(k), A^\nu(k)]\,. \tag{2.15}$$

Where $\mathcal{A}_\mu$ is analogously defined as in (2.2). Therefore, the Berry curvature locally has the form $F = F^{\mu,\nu} \mathrm{d}k_\mu \mathrm{d}k_\nu$. Similarly, there is a planar flux

$$W_k^{\mu,\nu} = U_k^\mu U_{k+\hat{\mu}}^\nu (U_{k+\hat{\nu}}^\mu)^\dagger (U_k^\nu)^\dagger. \tag{2.16}$$

It is easy to verify that $W_k^{\mu,\nu} = (W_k^{\nu,\mu})^\dagger$.

With this generalized form of Berry curvature, the n_{th} order Chern number of a $2n$-dimensional Brillouin zone is defined as

$$C_n = \left(\frac{1}{2\pi i}\right)^n \int_{BZ} \mathrm{Tr}\left[F(k)^n\right] d^{2n}k\,. \tag{2.17}$$

Here, $F(k)^n$ represents a wedge product of differential forms $F^{\mu,\nu}(k)\, dk_\mu\, dk_\nu$, which could be written equivalently as

$$\frac{2^n n!}{(2n)!} \sum_{\mu_1,\ldots,\mu_{2n}} \epsilon_{\mu_1,\ldots,\mu_{2n}} \prod_{t=1}^{n} F^{\mu_{2t-1},\mu_{2t}}(k). \tag{2.18}$$

It could be shown that C_n is always an integer, $\forall n \geq 1$.

On the discretized lattice, since the fluxes $W_k^{\mu,\nu}$ are an approximation of $\exp(F^{\mu,\nu})$, we calculate the discrete version of higher order Chern numbers with the following equation

$$\tilde{C}_n = \frac{n!}{(2n)!(\pi i)^n} \sum_k \sum_{\mu_1,\ldots,\mu_{2n}} \mathrm{Tr}\epsilon_{\mu_1,\ldots,\mu_{2n}} \prod_{t=1}^{n} \log W_k^{\mu_{2t-1},\mu_{2t}}. \tag{2.19}$$

When taking $n = 1$, Equation (2.19) coincides with (2.8). Since log function is analytical, thus could be represented by a power series, and $(\Omega^\dagger W \Omega)^n = \Omega^\dagger W^n \Omega$, we have

$$\tilde{C}_n(W_k^{\mu_{2t-1},\mu_{2t}}) = \tilde{C}_n(\Omega^\dagger W_k^{\mu_{2t-1},\mu_{2t}} \Omega),\ \forall \Omega \in U(N)$$

This discretized Chern number is an integer only in the continuum limit, therefore we use the MAE, i.e. mean absolute error instead for evaluation.

2.7 Sampling on Lie Groups

A specialized sampling method over compact Lie groups is required. Specifically, for a random variable X taking values in $U(N)$—the group from which link variables are drawn—we require the distribution of X to be invariant under left multiplication by group elements:

$$X \sim gX,\ \forall g \in U(N)$$

A practical method satisfying this requirement was proposed by Stewart [42], where unitary matrices are generated via the QR decomposition of complex Gaussian matrices. Specifically, let $A \in \mathbb{C}^{N\times N}$ be a matrix with i.i.d. standard complex normal entries (each with independent $\mathcal{N}(0,1)$ real and imaginary parts). Let $U = f(A)$ denote the unitary matrix produced by applying a fixed QR-based procedure to A. We assume this procedure is deterministic, i.e., the map $f : \mathbb{C}^{N\times N} \to U(N)$ is single-valued.

Proposition 5. *U and gU are identically distributed, $\forall g \in G$.*

Proof. By definition, $gU = f(gA)$. Then it suffices to show gA and A are identically distributed.

For complex matrices, we consider the two bijections. The first one is $p : A \to \begin{pmatrix} \mathrm{Re}A \\ \mathrm{Im}A \end{pmatrix}$. Then $p(gA) = \begin{pmatrix} \mathrm{Re}g & -\mathrm{Im}g \\ \mathrm{Im}g & \mathrm{Re}g \end{pmatrix} = \hat{g}p(A)$. It is easy to verify $\hat{g}$ is orthogonal. We then flatten the matrix with a vec operator

$$\mathrm{vec}(A) = (A_{11}, A_{12}, \ldots, A_{1N}, \ldots, A_{M1}, \ldots, A_{MN})$$

The following property is well known.

Proposition 6 (Vec Operator Identity)**.** *The vec operator and group actions are related via tensor products as vec$(gA) = (g \otimes I_N)$vec(A).*

Where $\otimes$ is the Kronecker product. Then $\mathrm{vec}(p(gA)) = (\hat{g} \otimes I_N)\mathrm{vec}(p(A))$. However, $\mathrm{vec}(p(A))$ is just $(\mathrm{Re}A_{ij}, \mathrm{Im}A_{ij})$, which follows the distribution $\mathcal{N}(0, I_{2N^2})$, and $\hat{g} \otimes I_N$ is still orthogonal, it follows that

$$\mathrm{vec}(p(gA)) \sim \mathcal{N}(0, (\hat{g} \otimes I_N) I_{2N^2} (\hat{g} \otimes I_N)^\top) = \mathcal{N}(0, I_{2N^2})$$

Therefore $\mathrm{vec}(p(gA)) \sim \mathrm{vec}(p(A))$. By bijectivity, $gA \sim A$. □

An immediate corollary is that this method induces a uniform distribution over $\det(U) \in S^1$. Since left-multiplying U by a scalar unitary matrix $e^{i\theta}I$ also preserves its distribution, we have

$$e^{i\theta} \det(U) = \det(e^{i\theta/N} IU) \sim \det(U). \tag{2.20}$$

3

Architecture and Setup

3.1 Network Architecture

As demonstrated in the previous section, the gauge symmetry present in this problem is completely local, thus has an enormous size. To be specific, since the Wilson loops at each site can be transformed independently, the total symmetry group is $U(N)^{N_{\text{site}}}$, an exponentially larger group than for more traditional group equivariant networks. This motivates a development of architectures that respect this gauge transformation, as we discuss in this chapter.

The input data in our network is the set of discretized Wilson loops $W_k^\gamma \in \mathrm{U}(N)$ and all equivariant layers in our setup operate on tensors of this form. The index γ counts the number of different orientations of the Wilson loops per site (in 2D, there is only one) for the input and serves as a general channel index in deeper layers. Hence, our layers operate on complex tensors of the shape $N_{\text{ch}} \times N_{\text{sites}} \times N \times N$.

3.1.1 Gauge equivariant layers

Our model is composed of the following equivariant layers which were introduced in [39] as well as our new gauge equivariant normalization layer.

GEBL (Gauge Equivariant Bilinear Layers) Given an input tensor W_k^γ, the layer computes a local quantity per site as

$$W_k'^\gamma = \sum_{\mu,\nu} \alpha_{\gamma\mu\nu} W_k^\mu W_k^\nu \,, \tag{3.1}$$

where W' has N_{out} channels and $\alpha_{\gamma\mu\nu} \in \mathbb{C}^{N_{\text{in}} \times N_{\text{in}} \times N_{\text{out}}}$ are trainable parameters. Using (2.13), it can easily be checked that this layer is equivariant. In practice, GEBL includes also a linear and a bias term which are obtained by enlarging W with its Hermitian conjugate and the identity matrix. In order to merge two branches of the network, two different W can also be used on the right-hand side.

GEAct (Gauge Equivariant Activation Layers) Given a tensor W_k^γ, the layer maintains channel size N_{in} and serves as an equivariant nonlinearity defined by

$$W_k'^\gamma = \sigma(\text{Tr}W_k^\gamma)W_k^\gamma \,, \tag{3.2}$$

where σ is a usual activation function. In Section 3.3, we prove a universal approximation theorem for a certain type of σ. In practice, we use $\sigma(z) = \text{ReLU}(\text{Re}z)$ to avoid gradient vanishing, hence referring to this layer also as GEReLU.

GEConv (Gauge Equivariant Convolution Layers) This is the only layer in our setup which introduces interactions between neighboring points. It is also the only layer for which the link variables U_k^γ are used. Given a tuple (U_k^μ, W_k^γ), the layer performs a convolution as

$$W_k'^\gamma = \sum_{\mu,d} \sum_\sigma \omega_{\mu d\gamma\sigma} (U_{k+d\hat{\mu}}^{d\mu})^\dagger W_{k+d\hat{\mu}}^\sigma U_k^{d\mu} \,, \tag{3.3}$$

where $U_k^{d\mu} = U_k^\mu \dots U_{k+(d-1)\hat{\mu}}^\mu$ and $d\hat{\mu}$ is the length-d vector in the μ direction in the lattice. The output W' of this layer has the shape $N_{\text{out}} \times N_{\text{site}} \times N \times N$ and ω are trainable weights of shape $\text{dim} \times d \times N_{\text{out}} \times N_{\text{in}}$. Note that this layer does not update the links. Using the transformations (2.12) and (2.13), one can check that this layer is equivariant as well. When we take a zero convolution kernel size, the layer degenerates into an equivariant linear layer that is completely local.

Trace Layer Given a tensor W_k^γ, this layer maintains the channel size and takes the trace of the fluxes as

$$T_k^\gamma = \text{Tr}W_k^\gamma \,. \tag{3.4}$$

Since the trace is invariant under the transformations (2.13), this layer renders the features gauge invariant. The output has the shape $N_{\text{in}} \times N_{\text{site}}$.

Dense Layer After the trace layer, we perform a real valued linear layer on T_k as

$$T_k' = w_{\text{Re}}^\gamma \cdot \text{Re}(T_k^\gamma) + w_{\text{Im}}^\gamma \cdot \text{Im}(T_k^\gamma) + b \,, \tag{3.5}$$

where w_{Re}^γ, w_{Im}^γ and b are trainable parameters. The output has the shape N_{site}. Note that we only transform gauge invariant features with this layer since it does not respect the gauge symmetry.

TrNorm (Trace Normalization Layers) The bilinear GEBL-layers introduced above quickly lead to training instabilities when stacked deeply, as demonstrated in Section 4.4.1 below. To solve this problem, we introduce a novel gauge-equivariant normalization layer which we insert after the nonlinearities. Given an input tensor W_k^γ, this layer maintains the channel size and performs a channel-wise normalization as

$$W_k'^\gamma = \frac{1}{|\text{mean}_\gamma\{\text{Tr}W_k^\gamma\}|} W_k^\gamma \,. \tag{3.6}$$

Where the prefactor $\text{mean}_\gamma\{\text{Tr}W_k^\gamma\}$ is the mean value of traces over different channels. This operation is gauge equivariant since the prefactor is gauge invariant.

After the normalization, the output features W' satisfy $\text{mean}_\gamma\{\text{Tr}W_k'^\gamma\} = e^{i\phi_k}$ for some $\phi_k \in \mathbb{R}$. In practice, we introduce a thresholding mechanism, by taking $\max\{\varepsilon,\ \text{mean}_\gamma\{\text{Tr}W_k'^\gamma\}\}$ as the divisor.

3.1.2 Network Architecture

We now use the layers introduced in the previous section to construct three different equivariant network architectures.

GEBLNet (Gauge Equivariant Bilinear Network) GEBLNet is a model designed to operate purely locally, meaning that at each lattice site, the network processes information independently, without direct communication between neighboring sites during the feature extraction process. More precisely, the inputs to the model are the fluxes W_k^γ alone, where γ denotes the different channels and k labels the spatial lattice sites. The features at different sites remain separated throughout the network, only being combined at the final global aggregation step.

The processing at each site is performed by a sequence of building blocks, each consisting of a GEBL, a GEAct, and a TrNorm. The GEBL layers generate local higher-order interactions among input channels. The GEAct layers introduce nonlinearity in an equivariant manner. To address potential instabilities caused by stacking multiple bilinear layers, TrNorm layers are inserted after each activation step, normalizing the trace of the features channel-wise and maintaining stable training dynamics.

After several such blocks, the resulting feature tensor is passed through a Trace layer, which extracts gauge-invariant scalar quantities at each site by computing the trace over the group elements. Subsequently, a fully connected dense layer is applied to these local traces to transform the features into sitewise scalar predictions. Finally, these scalar predictions are summed across all lattice sites to produce a single global output, corresponding to the predicted Chern number of the sample.

GEBLNet is our primary model to study. An illustrative overview of the GEBLNet architecture is provided in Figure 3.1.

GEConvNet (Gauge Equivariant Convolutional Network) GEConvNet is an alternative architecture that incorporates local interactions between neighboring lattice sites through gauge-equivariant convolutions. While GEBLNet only operates locally within each site, GEConvNet allows limited communication between adjacent sites, enabling the network to capture short-range correlations.

More precisely, the inputs to GEConvNet are both the link variables U_k^μ and the fluxes W_k^γ, where μ indexes the directions of the links, γ denotes the channels, and k labels the lattice sites. The link variables are used to implement equivariant convolutions that respect the underlying $U(N)$ gauge symmetry of the system.

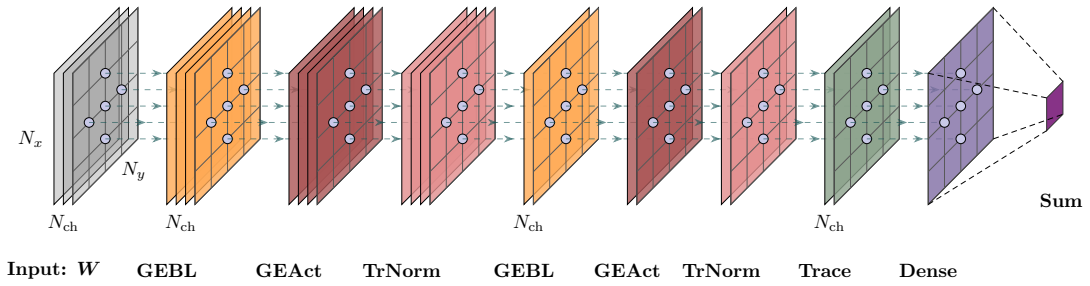

Figure 3.1: Architecture of GEBLNet. In this figure, the rectangles represent the spatial grid, and the number of layers (N_{ch}) represents the number of channels (γ). Each circle represents a site on the grid, and quantities on different sites do not interact with each other, until the last summation on grids.

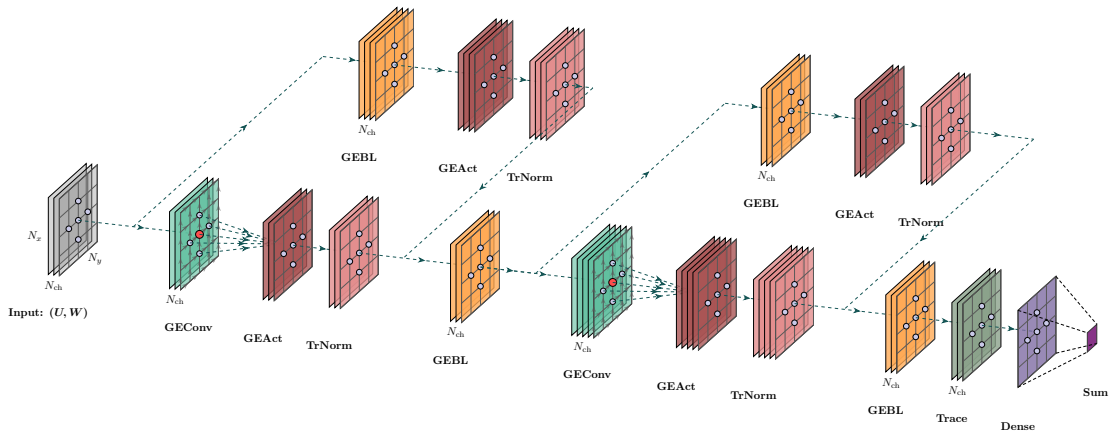

Figure 3.2: Architecture of GEConvNet. In this figure, the rectangles represent the spatial grid, the arrows on the grid represent links, and the number of layers (N_{ch}) represents the number of channels (γ). Each circle represents a site on the grid, and quantities on different sites do not interact with each other, except for the GEConv Layers, and the last summation on grids.

At each site, the processing is performed by two parallel branches: one branch consists of a sequence of a GEBL, a GEAct, and a TrNorm, while the other branch consists of a GEConv, a GEAct, and a TrNorm. The outputs of these two branches are then merged through an additional GEBL layer.

After several such blocks, the feature tensor is passed through a Trace to extract gauge-invariant scalar features by tracing over the group elements. A subsequent dense layer transforms these sitewise traces into scalar predictions. Finally, the outputs across all sites are summed to obtain a global prediction, corresponding to the Chern number of the sample.

By allowing controlled local interactions, GEConvNet offers a more expressive architecture compared to GEBLNet, potentially capturing subtle spatial structures in the data. However, it introduces additional complexity due to the use of link variables and neighbouring operations. An illustration of the GEConvNet architecture is provided in Figure 3.2.

We also introduce two baseline architectures for comparison, whose motivation and feasibility follows from Section 3.3. These two models also preserve gauge equivariance for the two-dimensional setting, but discard the freedom of gauge group action as in GEBLNet and GEConvNet, and operate exclusively on gauge invariant features that are extracted at the very first processing step. As is demonstrated later, they exhibit weaker expressibility and reduced scalability.

TrMLP (Trace Multilayer Perceptron) TrMLP is a simple model that operates on gauge-invariant quantities extracted directly from the input fluxes. Specifically, for each site k, it computes a characteristic feature vector

$$\left(\mathrm{Tr}(W_k^\gamma)^1, \ldots, \mathrm{Tr}(W_k^\gamma)^N\right),$$

where γ indexes the different input channels and N is a hyperparameter specifying the number of trace-based features per site. These vectors, already gauge invariant, are then passed through a multilayer perceptron (MLP), and finally summed across all sites to produce the global prediction for the Chern number.

DeepSpec (DeepSet on Spectral Data) This model preprocesses the flux data with a diagonalization, and then feeds the eigenvalues towards a DeepSet-like model, which is invariant under permutation of the input data. Specifically, for input with the form λ_k^i, the model calculates locally the output as

$$\mathcal{N}(\lambda_k^i)_k = \rho\left(\sum_i \phi(\lambda_k^i)\right)$$

where $\rho,\ \phi$ are arbitrary neural networks. Here, we set them as MLPs as well.

3.2 Training Setup

The models all follow the general strategy of calculating local quantities first, and summing over lattice sites. Therefore, we adopt two loss functions for our training: global loss L_g and standard deviation loss L_{std}. Specifically, given the network $f(W)$ and the Chern number $\tilde{C}$, the global loss calculates as

$$L_{\text{g}} = \|f(W) - \tilde{C}\|_1.$$

Meanwhile, L_{std} evaluates the entrywise standard deviation of the network output after the dense layer, denoted as $(g(W_k))_k$, namely

$$L_{\text{std}} \| \max(\{\text{std}(g(W_k)_k), \delta\}) - \delta\|_1,$$

where δ is a hyperparameter, set to 0.5 by default.

The standard deviation loss is necessary to prevent the model from collapsing to only zero outputs, since it forces the model to output locally differed quantities. This is particularly relevant for the training on trivial topologies only, as described below. The total loss function L_{total} adds these two terms, $L_{\text{total}} = L_{\text{g}} + L_{\text{std}}$.

With a total of 5000 training epochs, we adopt a default initial learning rate of 10^{-4} that decays by a factor of 0.1 at epoch 2500 and 3750, and an Adam optimizer with scheduling.

The width and depth of models vary across experiments and are specified in each corresponding setting, see Chapter 4.

For evaluation, we compute the accuracy by rounding the network output $f(W)$ to the nearest integer and comparing it with the Chern number $\tilde{C}$, unless otherwise specified.

3.3 Approximation properties of Gauge Equivariant Networks

In this section, we will present a universal approximation theorem for our models. We focus on GEBLNet, whose inputs are solely W_k.

Theorem 7 (Universal Approximation Theorem)**.** *For a compact Lie group G, and with the nonlinearity σ in GEAct taking the form $\tilde{\sigma} \circ Re$, where σ is bounded and non-decreasing, GEBLNet could approximate any class function on G.*

By Proposition 4, it suffices to show that GEBLNet can approximate any function $f(p_1, \ldots, p_n)$, where p_n is $\text{Tr}g^n$. Since G is compact, (p_i) takes value automatically on a compact set K_n, where the subscript denotes the dimension.

We formalize the network architecture GEBLNet. Given the flux tensor W_k, we stack the identity and its Hermitian conjugate to a second channel as

$$W_k'^{\gamma} = (W_{k,0}, W_{k,1}, W_{k,-1}) := (I, W_k, W_k^{-1}).$$

Afterwards, the tensor goes through several blocks, each containing three layers: GEBL, GEAct, and TrNorm.

After several blocks we calculate the trace per-channel and add a linear layer (the "Dense layer") at the end. The Dense layer acts on the real and imaginary parts separately. Then we take the sum over the site index (to calculate the topological invariant).

So the outputs have the following form:

$$W_k \longmapsto w \cdot \mathrm{TrNorm} \circ \hat{\sigma} \circ \mathrm{GEBL}_n \circ \cdots \circ \mathrm{TrNorm} \circ \hat{\sigma} \circ \mathrm{GEBL}_1(W_k)) + b.$$

Where $\hat{\sigma}(W_k^{\gamma}) = \sigma(\mathrm{Tr}W_k^{\gamma})W_k^{\gamma}$. We denote the set of these functions by $\mathcal{BLN}_{\sigma}(G)$, where the subscript σ indicates the choice of activation function. We further denote by $\mathcal{BLN}_{\sigma}^{k}(G)$ the subset of $\mathcal{BLN}_{\sigma}(G)$ with k blocks.

Since we attempted to learn local quantities $F(W_k)$, we omit the subscript k. Furthermore, we treat the flux W as an abstract element in the Lie group G, denoted as g. In this case where the input channel size is one, we propose the main result:

Theorem 8 (Universal Approximation Theorem)**.** *For any activation function $\sigma = \tilde{\sigma} \circ Re$, where $\tilde{\sigma}$ is bounded and non-decreasing, $\mathcal{BLN}_{\sigma}(G)$ is dense in $L^2_{class}(G)$.*

The proof of this will require the following lemma.

Lemma 9. *$\mathcal{BLN}_{\sigma}^{k}(G)$ is dense in $\{f(p_1, \cdots, p_{2^k}) : \|f\|_{\infty} < \infty\} \subset L^{\infty}(G)$, where $p_i = Trg^i$.*

Proof. We first assume that TrNorm is an identity layer, namely $\mathrm{TrNorm}(x) = x$. In this case, we prove this lemma by induction. For $k = 1$, the output has the following form

$$g \longmapsto \left(\sum_{i=0}^{2} \alpha_i^t g^t\right)_i \quad \longmapsto \quad \omega_i \mathrm{Tr}\sigma \left(\sum_{i=0}^{2} \alpha_i^t g^t\right)_i + b.$$

Note that $\mathrm{Tr}\hat{\sigma}(\sum_{i=0}^{2} \alpha_i^t g^t) = \sigma(\mathrm{Re}\alpha_i^t p_t)\alpha_i^t p_t$. For any channel index i, when taking only the real part (in other words, forcing $w_{i,\mathrm{Im}}$ in the dense layer to be zero), the output is simply

$$\begin{aligned} &\sigma\left(\sum_t \mathrm{Re}\alpha_i^t \mathrm{Re}p_t - \mathrm{Im}\alpha_i^t \mathrm{Im}p_t\right)\left(\sum_t \mathrm{Re}\alpha_i^t \mathrm{Re}p_t - \mathrm{Im}\alpha_i^t \mathrm{Im}p_t\right) \\ =&\hat{\sigma}\left(\sum_i \mathrm{Re}\alpha_i^t \mathrm{Re}p_t - \mathrm{Im}\alpha_i^t \mathrm{Im}p_t\right) \end{aligned}$$

Therefore, it is essentially a one-hidden-layer fully connected network on $\{(p_1, p_2)\} \simeq \mathbb{R}^4$. Thus the set is dense.

Assume this is the case for n, and we would like to prove the lemma for $n+1$. By denoting $2^n = N$, the layer input has the following form:

$$\tilde{\sigma}\left(\sum_{t=0}^{N} a_i^t(p_0, \cdots, p_{N/2})p_t\right)\left(\sum_{t=0}^{N} a_i^t(p_0, \cdots, p_{N/2})g^t\right),$$

Now the new "a_i^t"(denoted as b_i^t) takes the following form:

$$b_i^t = \sum_{p+q=t}\sum_{j,k} \alpha_{ijk}\tilde{\sigma}(a_j^t p_t)\tilde{\sigma}(a_k^t p_t)a_j^p a_k^q.$$

Consider the bijection $F : \mathbb{C}^{N+1} \to \mathrm{P}_N(\mathbb{C})$, given by $F(\vec{a}) = \sum_t a_t z^t$. Using this we define

$$\vec{a} * \vec{b} = F^{-1}(F(\vec{a})F(\vec{b})).$$

Then

$$\vec{b_i} = \alpha_{ijk}\tilde{\sigma}(\vec{a_j}\cdot\vec{p})\tilde{\sigma}(\vec{a_k}\cdot\vec{p})\vec{a_j}\vec{a_k} = \alpha_{ijk}H(p, \vec{a_j}, \vec{a_k}).$$

This forms a linear space $\mathcal{B}^{n+1} \subset (L^\infty(K_{n+1}))^{2N+1}$. For simplicity we henceforth omit the subscript on K_{n+1}.

We assume $(a_i^t)_{t=0}^N$ could approximate any constant function of $p_1, \cdots, p_{N/2}$. This is trivially true when $n = 1$, since it is a function on a constant and takes arbitrary constant values.

Denoting $e_0 = F^{-1}(1/d)$, where $d = \dim G$, we have $e_0 \cdot p = 1$, $\forall p \in K$. Since K is compact, there exists an open set U s.t. $e_0 \in \partial U$, and $b\cdot p \in (1, +\infty)$, $\forall b \in U, p \in K$.

On the other hand, it is easy to see that $\left\{b * b : b = \left(1, z, \cdots, z^N\right)\right\}$ is linearly independent as a subset. This way we could choose $2N+1$ elements $\{b^{z_t}\}_{t=0}^{2N}$ from its intersection with U, such that $\mathrm{span}\{b^{z_t} * b^{z_t}\} = \mathbb{C}^{2N+1}$.

Now given a constant vector $\vec{b} = (b_0, \cdots, b_{2N})$, there exits $\{\alpha_t\}$ such that $\vec{b} = \alpha_t b^{z_t} * b^{z_t}$. We want to show that $\vec{b}$ can be approximated by any precision ϵ.

Without loss of generality, assume $\sup\tilde{\sigma} = 1$ and $\inf\tilde{\sigma} = 0$. Then, for all ϵ, there exists $M_0 > 0$ such that for all $x > M_0/2$, $\tilde{\sigma}(x) \in (\sqrt{1-\epsilon}, 1)$. This gives

$$\left|\frac{d^2}{M^2}H\left(p, Me_0, Me_0\right) - 1\right| = |1 - \tilde{\sigma}(M)^2| < \epsilon, \quad \forall M > M_0.$$

By induction, there exists a_t such that $\|a_t - b^{z_t}\|_\infty < \min\{\epsilon, M/2\}$. Consider

$$\vec{b'} = \alpha_t\frac{1}{M^2}H(p, a_t, a_t) = \alpha_t\tilde{\sigma}(M\alpha_t\cdot p)^2 a_t * a_t.$$

Then

$$\begin{aligned} |\vec{b'} - \vec{b}| &= |\alpha_t(\tilde{\sigma}(M\alpha_t \cdot p)^2 - 1)b^{z_t} * b^{z_t} + \tilde{\sigma}(M\alpha_t \cdot p)(b^{z_t} * b^{z_t} - \alpha_t * \alpha_t)| \\ &\leq \alpha_t\epsilon|b^{z_t} * b^{z_t}| + 2\epsilon|b^{z_t}| + \epsilon^2 \\ &\leq C(\vec{b}, N)\epsilon. \end{aligned} \tag{3.7}$$

When the coefficient functions approximate constants, the last layer is essentially a one-hidden-layer fully connected network over $p_1, \cdots, p_{2N}$. Similar to the $N = 1$ case, as the width grows larger, the network can approximate any function f over $p_1, \cdots, p_{2N}$.

We now consider the Trace Normalization layer, and show that any function that could be represented by networks without TrNorm could also be represented by those with TrNorm. Without loss of generality, we only consider $\mathcal{BLN}^1_\sigma(G)$. For the sole GEBL and GEAct, since the Lie group is compact, we could scale the coefficients such that $|\mathrm{Tr}\sigma \circ \mathrm{GEBL}(G)| < \epsilon$. As a result, the output of the packed layer is essentially $\frac{1}{\epsilon}\sigma \circ \mathrm{GEBL}(G)$. Therefore, by rescaling the parameters of the dense layer again, the TrNorm layer is cancelled. This concludes the proof of the lemma. □

We may now complete the proof of Theorem 8.

Proof. (Proof of Theorem 8)
Since G is compact, we have $L^2(G) \supset L^\infty(G)$ and $\|f\|_2 \geq C\|f\|_\infty$. Therefore, by Proposition 4, for all $f \in L^2_{\text{class}}(G)$, and for any $\epsilon > 0$, there exists

$$f_n = f_n(p_1, \cdots, p_n) \in L^\infty(K)$$

such that $\|f - f_n\|_2 < 1/2\epsilon$. By Lemma 9 the function class $\mathcal{BLN}^k_{\hat{\sigma}}(G)$, consisting of neural networks with k gauge equivariant bilinear layers, can approximate any function $f(p_1, \ldots, p_k)$ arbitrarily well, provided k is large enough. We deduce that there exists $g \in \mathcal{BLN}^n_{\hat{\sigma}}(G) \subset L^\infty(G)$ such that $\|g - f_n\|_\infty < 1/2C\epsilon$. Therefore

$$\|g - f\|_2 < (C \cdot 1/2C + 1/2)\epsilon = \epsilon.$$

This concludes the proof of the main theorem. □

Similar techniques could also be applied to show the approximation property of GEConvNet. However, its expressibility is considerably weaker in practice, as is demonstrated in Chapter 4.

4

Numerical Experiments

In this chapter, we aim to empirically evaluate the effectiveness and robustness of our proposed gauge-equivariant neural network architecture (GEBLNet) in predicting Chern numbers of multiband topological insulators.

As is mentioned in Chapter 2, the major challenge that lies in this learning task is the massive symmetry group. To demonstrate this, we start by considering the simpler problem of predicting determinants of $N \times N$ real matrices A.

4.1 Learning Chern numbers using ResNets

Using the fact that $\operatorname{Tr}\log(X) = \log\det(X)$, the Chern number defined in (2.8) can be written as a sum over $\operatorname{Im}\log\det(W)$. As a warmup to predicting Chern numbers,

We construct a dataset containing $N \times N$ matrices with elements sampled from a uniform distribution on the unit interval $[0, 1]$. As a baseline, we use a naive multilayer perceptron (MLP) with residual connections $f : \mathbb{R}^{N^2} \to \mathbb{R}$, taking the matrix elements as input and predicting the determinant value.

The determinant of an $N \times N$ matrix can be expressed as an order N polynomial in the matrix elements. For example, the determinant of a 4×4 matrix A_{ij} is given by the fourth order expression.

$$\det(A) = \sum_{i,j,k,l=1}^{4} \epsilon^{ijkl} A_{1i} A_{2j} A_{3k} A_{4l}, \tag{4.1}$$

where ϵ is the totally antisymmetric Levi-Cevita tensor.

Inspired by the functional form of the determinant in equation (4.1), we consider higher order layers with structure

$$A^{\text{out}}_{ij} = \sum_{k_1,\ldots,k_{2R}} \theta_{ij}{}^{k_1 \ldots k_{2R}} A^{\text{in}}_{k_1 k_2} \ldots A^{\text{in}}_{k_{2R-1} k_{2R}}, \tag{4.2}$$

Table 4.1: Relative error δ for linear MLP and bilinear residual architectures predicting the determinant of real 4×4 matrices with uniform random elements in $[0, 1]$. Standard MLP architecture fails to learn the determinant relation.

Architecture	Layers	δ
MLP	2	1.02
	3	1.02
Bilinear	2	0.01
	3	0.01

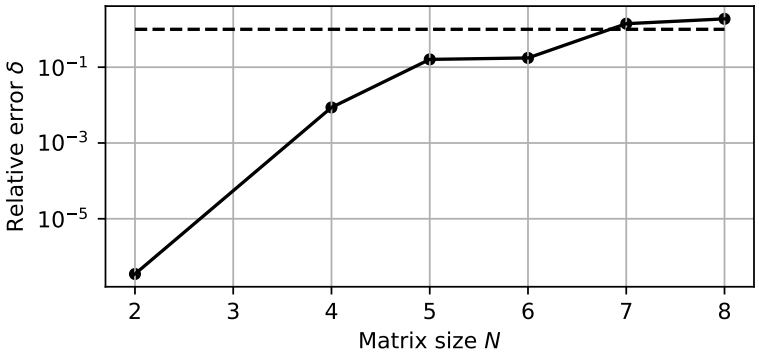

Figure 4.1: Best relative error of predicted matrix determinants for polynomial architectures as a function of increasing matrix size. The ablations include layers up to order 4. Dashed line indicates relative error of a mean predictor. Architectures considered include layers of order ≤ 4, and depth ≤ 4, containing terms of up to order 16 by composition.

where R is the number of factors of A in the layer, and $\theta_{ij}^{k_1 \ldots k_{2R}}$ are learnable parameters. An architecture containing layers with $R = 2$ will be referred to as bilinear. As the number of parameters grows quickly with order R, we use layers of order $R \leq 4$ and construct higher order terms by composition of multiple layers. For example, two layers of order $R = 3$ and $R = 2$ will contain terms of order $2, 3$ and 6 when using residual connections.

Predicted determinants $f(A)$ are evaluated against the target determinant value $\det(A)$ using absolute relative error $\delta = |f(A) - \det(A)| \,/\, |\det(A)|$. Table 4.1 shows the failure of a residual MLP architecture to learn the determinant of 4×4 real matrices with uniformly distributed elements on $[0, 1]$, whereas a bilinear architecture with layers of order 2 can achieve low relative error.

Even though these higher order layers provide an architecture that is expressive enough for determinants in small dimensions, they quickly run into issues for larger matrices. Figure 4.1 shows that for matrix sizes corresponding to band size ≥ 4, learning the determinant becomes prohibitively hard without better model priors. This leads us to resort to a equivariant approach.

4.2 Data Generation

We now formulate the data generation process of the main learning task. Throughout the training and evaluation procedure, we generate samples on the fly, with a batch size of 32.

4.2.1 General Dataset

The data generation pipeline consists of two main steps. Given a grid size $N_x \times N_y$, we first generate random link variables using the following algorithm:

1. For $\mu = x, y$, draw $A_k^\mu \sim \mathcal{N}(0,1)^{N\times N}$.

2. Perform a QR decomposition on A_k^μ, decomposing it into the product of a unitary matrix U_k^μ and a semi-definite matrix Σ_k^μ, $A_k^\mu = U_k^\mu \Sigma_k^\mu$.

3. Use U_k^μ as the link variable for site k in direction μ.

As discussed in Section 2.7, the distribution of the links generated in this way is uniform on $\mathrm{U}(N)$, i.e. the random variables U and gU are identically distributed for all $g \in \mathrm{U}(N)$.

In the next step, we compute the fluxes W_k using (2.9) and the discrete Chern number $\tilde{C}$ using (2.8). Ultimately, this yields a dataset of data-value pairs $((U_k^\mu, W_k^\gamma), \tilde{C})$. We generate the training samples continuously during training to avoid overfitting.

4.2.2 Diagonal Dataset

For some of our experiments, we require control over the distribution of the Chern numbers in our training data. To this end, we employ a different data generation strategy. Due to the invariance of our model under gauge transformations, training samples which lie in the same gauge orbit are equivalent in the sense that the parameter updates they induce are the same. This implies that we can select an arbitrary element along the gauge orbit for training. As discussed in Section 2.5, there is always a diagonal matrix with U(1)-valued components in the orbit. Therefore, by training on these matrices in $\mathrm{U}(1)^N$ and manipulating the distribution of the diagonal values, we can generate datasets with different distributions of Chern numbers.

To propose the exact data generation scheme, we analyze the constraints imposed by the discrete definition of the Chern number. By the definition of fluxes in (2.9), each link appears exactly twice in all plaquettes, once in itself, and once inverted. For example, U_k^x appears in itself in W_k and inverted in $W_{k-\hat{y}}$. Then we have:

$$\prod_k \det W_k = \prod_{\mu,k} \det U_k^\mu (\det U_k^\mu)^{-1} = 1$$

Specifically, since $\sum_k \mathrm{Im}(\log(\det W_k)) = \mathrm{Im}(\log(\prod \det W_k)) \mod 2\pi$, the discrete Chern number $\tilde{C}$ is an integer.

Proposition 10. *$\tilde{C} = \frac{1}{2\pi}\sum_x F_x = n \in \mathbb{Z}$.*

Then the necessary condition for a set of plaquettes to be generated from some links is:

$$\prod_k \prod_\lambda e^{i\theta_k^\lambda} = e^{i\sum_k \sum_\lambda \theta_k^\lambda} = 1, \tag{4.3}$$

On the other hand, given any W_k that is diagonal per site, suppose it is generated by diagonal links $U_k^\mu = \mathrm{diag}\{e^{i\tau_{k,\mu}^1}, \ldots, e^{i\tau_{k,\mu}^N}\}$. Then for each index λ we have the following equations:

$$\prod e^{i\tau_{k,x}^\lambda} e^{i\tau_{k+\hat{x},y}^\lambda} e^{-i\tau_{k+\hat{y},x}^\lambda} e^{-i\tau_{k,y}^\lambda} = 1, \forall k \tag{4.4}$$

This implies a necessary condition for W_k to be generated from diagonal links is that, for any λ, $\sum \theta_k^\lambda = 0$. We omit the subscript λ for now.
Recall that k is the flattened index of (i, j), which could have the possible form $k = N_{\text{site}} i + j$. If we further flatten the index (k, μ) as k for $\mu = x$, $k + N_{\text{site}}$ for $\mu = y$, then the equations become linear:

$$\tau_k + \tau_{(k+N_{\text{site}}+1) \bmod 2N_{\text{site}}} - \tau_{(k+N_x) \bmod N_{\text{site}}} - \tau_{k+N_{\text{site}}} = \theta_{\hat{x}}, \ \forall k, \tag{4.5}$$

which is essentially

$$\begin{pmatrix} 1 & & -1 & & & -1 & 1 & & & \\ & \ddots & & \ddots & & & \ddots & \ddots & & \\ -1 & & \ddots & & -1 & & & \ddots & \ddots & \\ & \ddots & & \ddots & & & & & \ddots & 1 \\ & & -1 & & 1 & 1 & & & & -1 \end{pmatrix}^\top \begin{pmatrix} \tau_0 \\ \tau_1 \\ \vdots \\ \tau_{2N_{\text{site}}-1} \end{pmatrix} = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_{N_{\text{site}}-1} \end{pmatrix} \tag{4.6}$$

The coefficient matrix has rank $N_{\text{site}} - 1$, and it is solvable iff. $\sum_k \theta_k = 0$, and that is exactly what the necessary condition specifies. Therefore, the fluxes W_k can be generated from diagonal U_k^μ if and only if

$$\forall \lambda, \ \prod_k e^{i\theta_k^\gamma} = 1. \tag{4.7}$$

This determines a submanifold M' in $M = \{m \in U(1)^{N \times N_{\text{site}}} : m \text{ satisfies (4.3)}\}$ with codimension $N - 1$. With the natural metric on $U(N)^{N_{\text{site}}} \supset M$, defined as $d(g, h) = \|\psi_k^\lambda\|_2$, where ψ_k^λ are phase angles of eigenvalues of gh^{-1}, M' is a $\pi\sqrt{\frac{N}{N_{\text{site}}}}$-net of M. For each channel λ, suppose $\sum_k \theta_k^\lambda = \phi_\lambda$, $\phi_\lambda \in [-\pi, \pi)$. Let the new θ be $\tilde{\theta}_k^\gamma = \theta_k^\lambda + -\phi_k / N_{\text{site}}$. Then

$$d(W, \tilde{W}) \le \sqrt{\sum_{k,\lambda} \left(\frac{1}{N_{\text{site}}}\right)^2 \phi_k^2} \le \pi\sqrt{\frac{N}{N_{\text{site}}}}.$$

As the number of sites gets larger (the grid gets more refined), the net gets denser. We can further extend the sufficient condition by considering the permutations, since the permutation matrices are also unitary and their actions on fluxes are adjoint.

We now propose the diagonal data generation scheme:

1. Generate label $F_k \in [-\pi, \pi)$, such that $\sum F_k = 2\pi n$.
2. For every k but the last one, generate $(\phi_k)_x$ such that $\sum_\lambda \phi_k^\lambda = F_k$.
3. For every k but the last one, let W_k be $\mathrm{diag}\{e^{i\theta_k^1}, \ldots, e^{i\theta_k^N}\}$.
4. Let the last $W_{\hat{k}}$ be $\prod_{k \neq \hat{k}} W_k^{-1}$.

The last product will not cause confusion since diagonal matrix multiplication is commutative.

The process could also be reversed: generate the fluxes first, then find a solution to (4.6) to get the links. This way, we could operate directly on the distribution of eigenvalues, thus customizing the data generation process. Furthermore, the diagonal dataset reduces the computation cost significantly for training.

4.3 Model Comparison

In this section, we compare the model performance in the benchmark training task over a grid size of 5×5 and $N = 4$ filled bands.

4.3.1 Baseline Equivariant Models

Since Chern number is a gauge invariant quantity, the naive strategy to learn it is through qauge invariant features. Following from the discussions in Section 2.4, we could either extract eigenvalues straightforwardly, or the traces of powers $\mathrm{Tr}W^n$. This leads to the two baseline models: DeepSpec and TrMLP, introduced in Section 3.1.

DeepSpec is composed of two parts: the element-wise embedding function ϕ and the set-level processing function. We set both the embedding function and the processing function to be MLPs, and test the combinations of different embedding functions and set-level functions under the benchmark training task, as shown in Table 4.2.

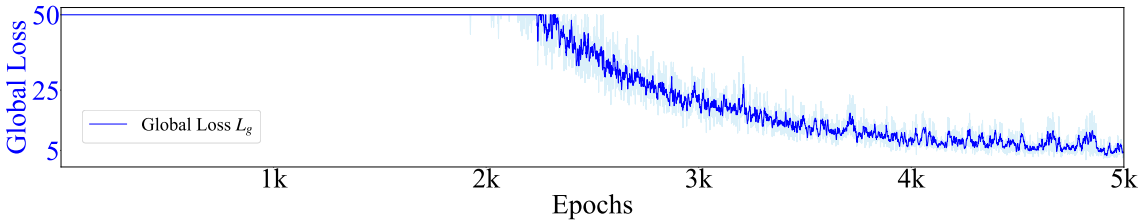

Figure 4.2: Global loss L_g of model D1-2, clipped at 50. During the initial steps of the training, the global loss maintained at a very high level (around 10000), and only managed to decrease tsubstantially after 2500 epochs.

ID	Channel Size of ρ	Channel Size of ϕ	Accuracy
D1-1	$1 \to 32 \to 16 \to 8$	$1 \to 16 \to 8$	9.6%
D1-2	$1 \to 32 \to 16 \to 8$	$1 \to 32 \to 16 \to 8$	7.8%
D1-3	$1 \to 32 \to 16 \to 8$	$1 \to 64 \to 32 \to 16$	10.9%
D2-1	$1 \to 64 \to 32 \to 16$	$1 \to 16 \to 8$	5.5%
D2-2	$1 \to 64 \to 32 \to 16$	$1 \to 32 \to 16 \to 8$	11.1%
D2-3	$1 \to 64 \to 32 \to 16$	$1 \to 64 \to 32 \to 16$	12.0%

Table 4.2: Classification accuracy of DeepSpec variants with different channel sizes. Models are denoted as Dx-y, where x indexes the processing channel configuration and y indexes the embedding channel size. All models are trained on the dataset 4 bands on a 5×5 grid, and evaluated on the corresponding validation set.

As summarized in Table 4.2, the DeepSpec model doesn't perform well in predicting the Chern numbers. Furthermore, increasing the capacity of either subnetwork does not lead to improvement in classification accuracy. In particular, models with a larger ϕ (e.g., D1-3, D2-3) do not perform significantly better than those with smaller embeddings, suggesting that the bottleneck lies not in the expressive power of ϕ or ρ. The overall low accuracy across all configurations indicates that DeepSpec struggles to extract the relevant topological information with invariant spectral features directly. A loss curve of global loss L_g is demonstrated in Figure 4.2. This indicates a need for preserving equivariant features to learn the Chern numbers more efficiently.

TrMLP operates on traces of powers, and the maximum power to extract is a untrainable hyperparameter. We tested the performance of TrMLP with various model setups. Out of the 8 settings, only 3 of them converged, while the others all failed in the training stage due to numerical instabilities.

Since W is drawn from $U(N)$ uniformly with respect to the Haar measure, for any integer n, W^n is uniform on the Haar measure as well. Therefore, $\forall m,\ n,\ \mathrm{Tr}W^m$ and $\mathrm{Tr}W^n$ are identically distributed, and within the disk $B(0, N)$. Therefore, the increasing of maximum power and channel size is not the cause of the instability of the MLP following the feature extraction.

ID	Channel Size	Accuracy
T1	$4 \to 8 \to 8$	19.0%
T2	$8 \to 16 \to 8$	27.5%
T3	$8 \to 32 \to 16 \to 8$	93.3%

Table 4.3: Classification accuracy of GEConvNet variants with different channel sizes and kernel sizes. Models are denoted as Tx, where x indexes the channel configuration. All models are trained on the dataset 4 bands on a 5×5 grid, and evaluated on the corresponding validation set.

The 3 successful training runs, along with their evaluation accuracies, are listed in Table 4.3. There is only one run that learned the valid quantity, achieving an accuracy of 93.3%. This further verifies the instability of TrMLP, the reason behind which is potentially related to initialization, and requires further study.

4.3.2 GEConvNet

GEConvNet, equipped with GEConv layers, is theoretically more expressive than GEBLNet due to its ability to capture non-local information through gauge-equivariant convolutions. However, the following empirical results show that GEConv demonstrates weaker performance, and the accuracy decreases as the kernel size increases.

We evaluate GEConvNet on the task of predicting Chern numbers from synthetic gauge field configurations under various problem settings. Specifically, we construct GEConvNets with varying symmetry constraints and design choices, as summarized in Table 4.4.

All models are trained on synthetic datasets generated as described in Section 4.2, using configurations sampled uniformly from $\mathrm{U}(N)^{\mathrm{sites}}$ with random fillings. Training is conducted on a fixed grid size of 5×5 and band number $N = 3$, while evaluation is performed on a separate validation set drawn from the same distribution.

The classification accuracy of each model, is obtained by rounding the predicted Chern number to the nearest integer. Deeper networks (C3, C4) consistently outperform shallower ones (C1, C2) when the kernel size is 0, indicating the importance of depth in capturing the local structure relevant to Chern number prediction. However, as the kernel size increases ($k = 2, 4$), accuracy declines across all models. One possible explaination is that, since the Chern number is defined as a summation over entirely local quantities, this additional capacity to couple neighboring sites has the risk to introduce redundant degrees of freedom, ultimately hampering the performance.

Figure 4.3 explicitly compares the label of the same sample with its outputs in model C4-0 and C4-2. As is speculated, the capability to couple neighbouring sites eventually costs GEConvNets the capability to capture local quantities.

ID	Channel Size	Kernel Size	Accuracy
C1-0	$1 \to 32 \to 16$	0	44.5%
C2-0	$1 \to 64 \to 32$	0	54.2%
C3-0	$1 \to 32 \to 16 \to 16 \to 8$	0	91.7%
C4-0	$1 \to 64 \to 32 \to 16 \to 8$	0	88.7%
C1-2	$1 \to 32 \to 16$	2	23.7%
C2-2	$1 \to 64 \to 32$	2	15.4%
C3-2	$1 \to 32 \to 16 \to 16 \to 8$	2	35.7%
C4-2	$1 \to 64 \to 32 \to 16 \to 8$	2	30.1%
C1-4	$1 \to 32 \to 16$	4	22.3%
C2-4	$1 \to 64 \to 32$	4	24.8%
C3-4	$1 \to 32 \to 16 \to 16 \to 8$	4	23.9%
C4-4	$1 \to 64 \to 32 \to 16 \to 8$	4	40.3%

Table 4.4: Classification accuracy of GEConvNet variants with different channel sizes and kernel sizes. Models are denoted as Cx-k, where x indexes the channel configuration and k indicates the convolution kernel size. All models are trained on the dataset 4 bands on a 5×5 grid, and evaluated on the corresponding validation set.

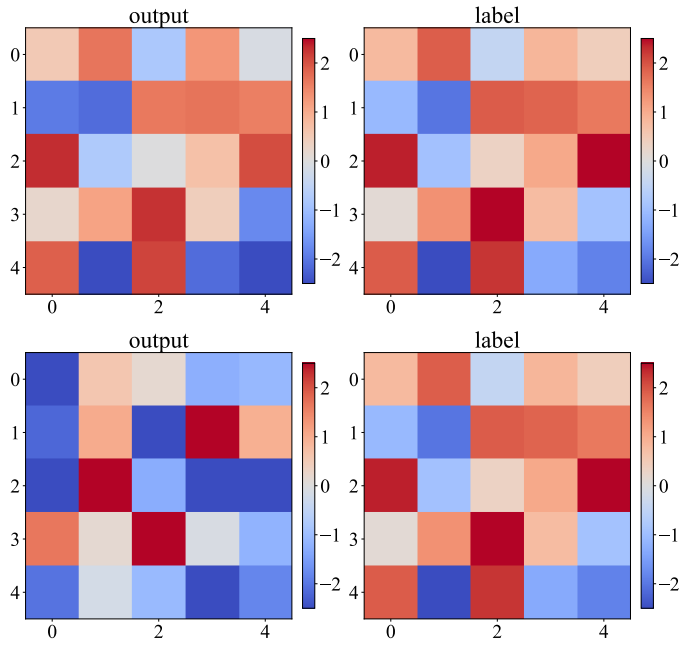

Figure 4.3: Comparison between the predicted local Chern density (left) and the ground truth (right) of the same sample. The predictions are produced by GEConvNet models C4-0 (top) and C4-2 (bottom), respectively, where C4-0 uses kernel size 0 and C4-2 uses kernel size 2.

4.3.3 GEBLNet

GEBLNet serves as our primary model to study. In contrast to baseline models and GEConvNet, it possesses two key characteristics: it operates exclusively on local quantities, and it maintains gauge-equivariant matrix features throughout the network up to the final layer. To assess its expressibility and robustness, we evaluate a range of architectural configurations under multiple random seeds. The full set of results is summarized in Table 4.5.

ID	Channel Size	Seed	Accuracy
B1-1	$1 \to 16 \to 8 \to 4$	289	92.5%
B1-2	$1 \to 16 \to 8 \to 4$	150	29.6%
B2-1	$1 \to 16 \to 8 \to 8 \to 4$	289	26.4%
B2-2	$1 \to 16 \to 8 \to 8 \to 4$	150	84.3%
B3-1	$1 \to 32 \to 16$	289	93.7%
B3-2	$1 \to 32 \to 16$	150	94.8%
B4-1	$1 \to 32 \to 16 \to 8$	289	94.2%
B4-2	$1 \to 32 \to 16 \to 8$	150	95.9%
B5-1	$1 \to 32 \to 16 \to 16 \to 8$	289	95.6%
B5-2	$1 \to 32 \to 16 \to 16 \to 8$	150	94.4%
B6-1	$1 \to 64 \to 32 \to 16$	289	94.2%
B6-2	$1 \to 64 \to 32 \to 16$	150	97.5%

Table 4.5: Classification accuracy of GEBLNet variants with different channel configurations and random seeds. Models are denoted as Bx-y, where x indexes the architecture and y indexes the seed configuration. All models are trained on synthetic data with 4 bands on a 5×5 grid.

GEBLNet shows strong expressive capacity even at small sizes. For instance, configuration B1 achieves an accuracy as high as 92.5%, indicating that the model is capable of learning meaningful topological features with minimal channel width and depth. However, this performance is highly sensitive to the choice of initialization, as evidenced by the large accuracy gap between different seeds (e.g., 92.5% vs. 29.6% for B1).

As the network width increases, both accuracy and stability improve. Larger configurations such as B3–B6 consistently achieve high accuracy across different seeds, with performance exceeding 94% and impact from choices of seeds significantly reduced. This suggests that once the model surpasses a certain capacity level, further increases in size yield minor improvements in accuracy, yet provide robustness to initialization. Compared with other models, GEBLNet achieves a significantly higher and stabler performance, and as the model complexity increases, the accuracy plateaus at a high level.

A complexity-accuracy comparison in Figure 4.4 further compared GEBLNet with other equivariant networks in terms of parameter size. The results show the superior

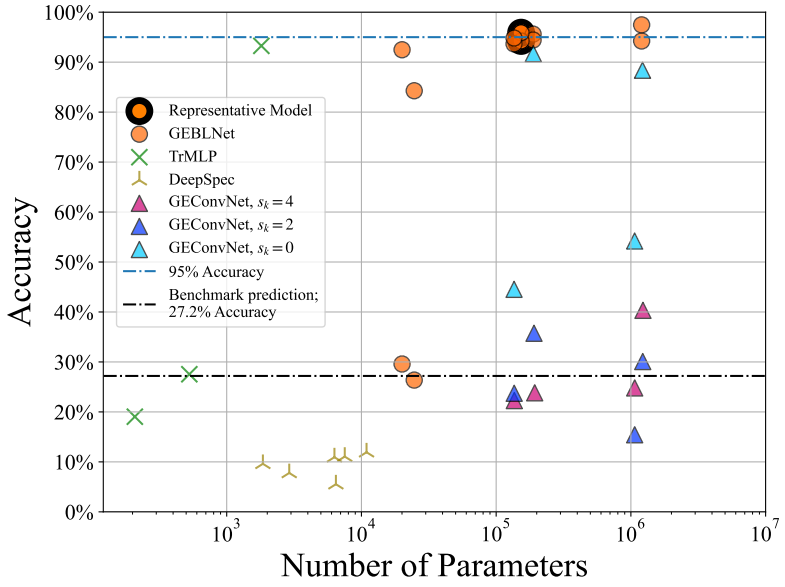

Figure 4.4: Comparison of model complexity and accuracy across different architectures. Complexity is measured by the number of learnable parameters, and s_k denotes the kernel size for GEConv layers. Each marker represents a training run with variations in models, learnable parameters, and random seeds. See detailed settings in Table 4.2, 4.3, 4.4 and 4.5.

performance and stability of GEBLNet across a range of model settings. Notably, networks with more than 10,000 parameters already achieve strong expressivity, ruling out the need for excessive capacities. Based on this observation, we adopt configuration B4 as the optimal model, marked with the orange thick-bordered circles. Meanwhile, the black dash-dotted line at 27.2% refers to the accuracy of the naive prediction that outputs zero Chern number for any sample, serving as the minimal threshold for meaningful performance.

With B4-1, we push the limit of GEBLNet by increasing the band number and complexity. The model is trained and evaluated on a fixed 5×5 grid while gradually increasing the number of bands from 4 to 8. As shown in Table 4.6, GEBLNet maintains high accuracy in the range of 4 to 7 bands, with performance only slightly degrading as the number of bands increases. A significant drop is observed at 8 bands, suggesting a level beyond which the model is no longer adequate for the task. However, as Figure 4.5 indicates, the differences between ground truths and outputs are within a reasonable range, thus could potentially be bridged with larger capacity or more refined structures. Despite the limitation the current model encounters, GEBLNet shows a strong and robust performance overall.

Table 4.6: Accuracy of GEBLNet trained and evaluated on a 5^2 grid.

Bands	4	5	6	7	8
Accuracy	95.9%	94.0%	93.8%	91.7%	52.5%

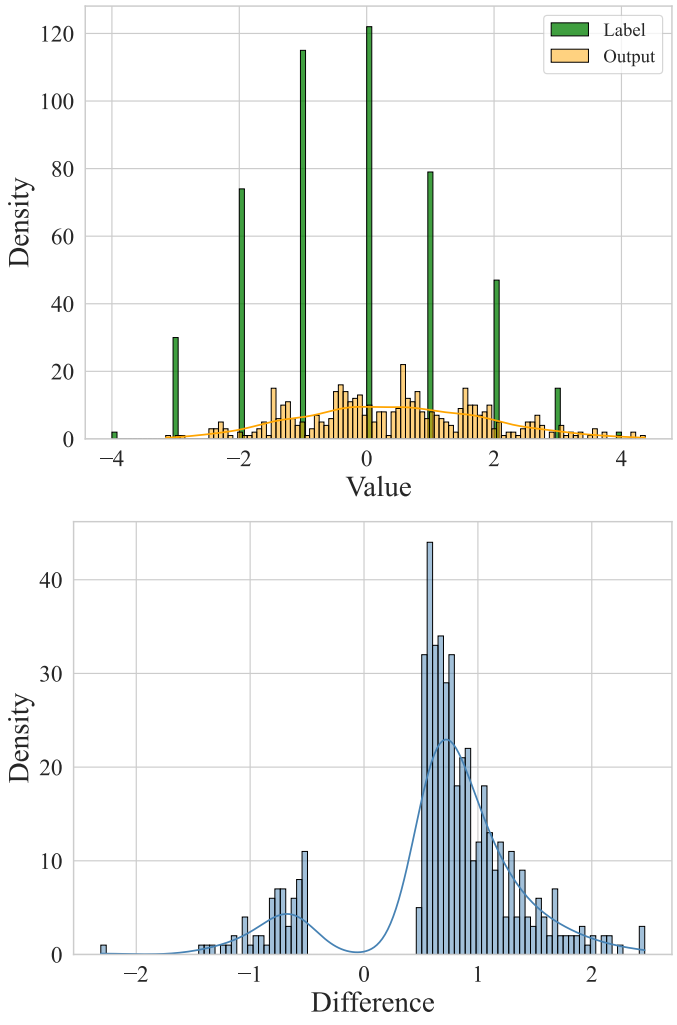

Figure 4.5: Distribution analysis of GEBLNet predictions on misclassified samples from the 8-band case. Left: comparison between the model output and the ground truth label. Right: histogram of the prediction error (output minus label).

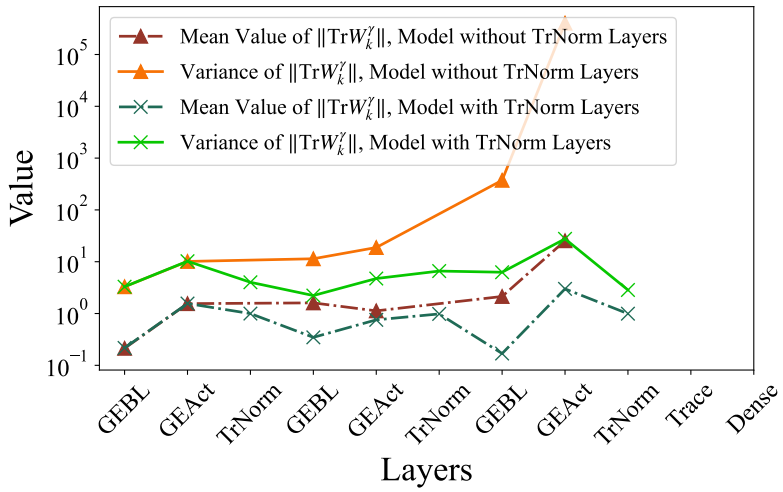

Figure 4.6: Caption

4.4 Generalization of GEBLNet

4.4.1 Training on Trivial Samples

In order to test the generalization properties of GEBLNet, we train exclusively on diagonal and topologically trivial samples. To achieve this, non-trivial samples were manually filtered out during data generation, turning L_{g} effectively into $\|f(W)\|_1$. We continue to adopt the benchmark training task over a grid size of 5×5 and $N = 4$ filled bands.

In this training scenario, we first make an adjustment to the evaluation process. Since the dataset is entirely nullified, the model is limited to learning the Chern number up to a global, sample-independent rescaling factor, i.e.

$$\exists k_f \ s.t. \ f(W) \approx R_f C$$

Here, the L_{std} serves to prevent the model from vanishing to zero everywhere, i.e. $R_f = 0$, by forcing the outputs to locally differ. Additionally, we evaluate on general samples by calculating a rescaling factor

$$R_{\mathrm{scale}} = \frac{\mathrm{mean}\{C\}}{\mathrm{mean}\{f(W)\}}$$

over a large set of non-trivial samples, and rescale the output as $R_{\mathrm{scale}} f(W)$.

A naive GEBLNet model without TrNorm layers can learn the Chern number for up to 3 bands but fails for 4 bands and above, mostly outputting zero local quantities except for a few random seeds. Specifically, out of the 10 seeds tested, only 2 of them result in a successful run. We attempt to solve this initialization instability with multiple strategies.

Seed	83	150	189	247	289
Turning Point	1393	1508	1706	1846	-

Table 4.7: Turning point of model B4 initialized with seed 83, trained on trivial data on a 5×5 grid, with 4 filled bands.

Seed	160	47	189	35
Accuracy	92.7%	94.3%	95.4%	93.8%

Table 4.8: Accuracy of model B4 initialized with seed 83, evaluated on non-trivial data on a 5×5 grid, with 4 filled bands.

TrNorm Layers We begin by analyzing the statistics of the absolute values of layer-wise outputs in GEBLNet. Specifically, we examine the B4-1 configuration trained without the TrNorm layer, which did not succeed in this training task—its output collapses within 500 epochs. As shown in Figure 4.6, we track the mean and variance of the output across layers for a representative batch. In the absence of TrNorm, the variance escalates rapidly, reaching magnitudes on the order of 10^5 after the final GEAct layer. Given this fact, a plausible explanation for the numerical instability and vanishing gradient is that such enormous growth of variance leads the model to abandon the standard deviation loss entirely, preferring to nullify the output rather than attempting to regress toward a meaningful solution.

This motivates the implementation of TrNorm. The expression given in Equation (3.6) ensures that, after each TrNorm layer, the mean value of traces across channels is normalized to 1, which is in alignment with Figure 4.6. This cancels out the accumulation of variance throughout the network and stabilizes the training. Figure 4.7 compares the loss curve of B4, initialized with seed 83, with and without TrNorm, up to the first 3000 epochs.

When TrNorm is applied, the training dynamics can be roughly divided into three stages. In the initial phase (lasting approximately 1800 epochs), the network degenerates to producing near-zero outputs, indicating an inactive status. In the second stage, the standard deviation loss L_{std} decreases while the global loss L_g increases, yet the total loss $L_g + L_{\text{std}}$ remains nearly constant, suggesting a transition into the desired learning process. Eventually, in the third stage, both L_g and the average standard deviation stabilize at ideal levels, and the model converges to a meaningful solution. These observations suggest that a key challenge lies in shortening the initial inert phase, and enabling the network to begin efficiently learning the target quantity earlier in training.

The effect of the TrNorm layer is tested under five different random seeds, as summarized in Table 4.7. To precisely identify the transition out of the initial inert phase (Phase I), we define the turning point as the first epoch at which the standard deviation loss satisfies $L_{\text{std}} < 0.49$. Among the five runs, four successfully converged to

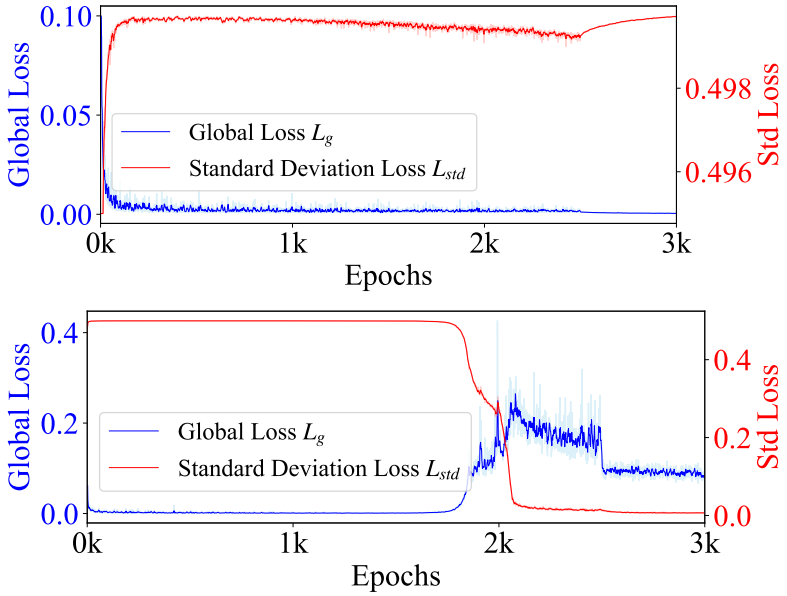

Figure 4.7: Global loss and standard deviation loss curve of GEBLNet, with configurations identical to B4, trained on a diagonal, trivial dataset, to learn the Chern number on a 5^2 grid, with 4 filled bands. Up: B4 without TrNorm. Bottom: B4 with TrNorm (B4-1).

valid solutions, with their turning points occurring between epoch 1300 and epoch 1900. This suggests a generally reliable convergence with TrNorm and a consistent duration of Phase I. Furthermore, we evaluate model B4 and study the local outputs of 1024 samples after rescaling, and compare the scaling factor calculated per batch. Figure 4.8 indicates the consistency of the rescaling factor across various samples, suggesting that the model succeeds in learning local quantities. This is further validated by evaluation on general datasets, see Table 4.8.

Perturbation Term To further improve the robustness of GEBLNet when trained on trivial datasets, we introduce a residual-like modification that enforces non-zero outputs. Specifically, for the final dense layer f acting on $(\mathrm{Tr}W_k^1, \ldots, \mathrm{Tr}W_k^n)$, we add a nonzero perturbation component $g(x)$, such that the final output takes the form $f(x) + g(x)$. Since f is linear, we require g to be nonlinear and positively bounded from zero. In practice, we define $g(x)$ as

$$g_\omega(\mathrm{Tr}W_k^1, \ldots, \mathrm{Tr}W_k^n) = \delta(1 + \exp(\omega)) \left(\sum_i (\mathrm{Tr}W_k^i)^2 \right),$$

where ω is a learnable scalar parameter and δ is a small hyperparameter (set to 0.01).

The expectation of g has a positive lower bound with respect to the hyperparameter chosen. Letting $(\mathrm{Tr}W_k^i)\lambda$ be denoted as $x\lambda^{k,i}$, where λ indexes samples, we note that $\mathrm{Var}(x_\lambda^{k,i})$ corresponds to the output variance from the final TrNorm layer, as in Figure 4.6). Denote the generic random variable from which $x_\lambda^{k,i}$ is drawn as X.

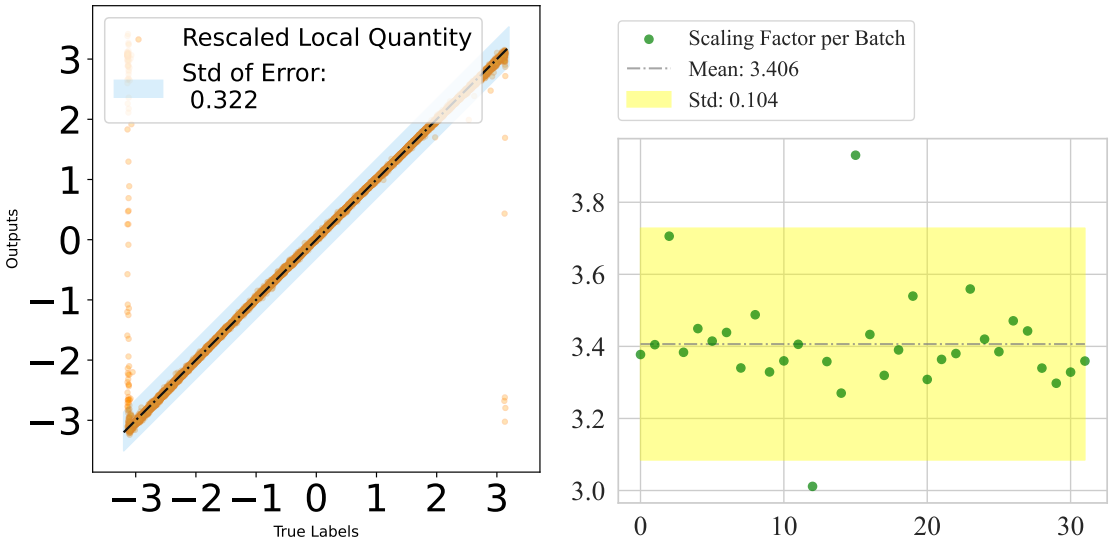

Figure 4.8: Left: comparison of rescaled local outputs with local true values. Points closer to the reference line $y = x$ indicate higher accuracy in capturing local quantities. Right: scaling factors per batch compared with that calculated over all 1024 samples.

Then the expected value of $g(X)$ satisfies:

$$E(g(X)) = \delta(1 + \exp(\omega))N_{\text{ch}}E(X^2) \tag{4.8}$$

$$= \delta(1 + \exp(\omega))N_{\text{ch}}\frac{1}{N_{samples}}\left[\sum_{\lambda}[E(X_\lambda)^2 + \text{Var}(X_\lambda)]\right] \tag{4.9}$$

$$\geq \delta(1 + \exp(\omega))N_{\text{ch}}\left[1 + \frac{1}{N_{samples}}\sum_{\lambda}\text{Var}(X_\lambda)\right] \tag{4.10}$$

$$\geq \delta(1 + \exp(\omega))N_{\text{ch}}. \tag{4.11}$$

The inequality (4.10) follows from the normalization property of the TrNorm layer, which enforces $E(X_\lambda)^2 = 1$, $\forall\lambda$.

After introducing this perturbation mechanism, the previously failing model B4 with seed 289 successfully learned to predict on trivial samples. However, further experiments with additional seeds revealed instability: in some runs, models that had exited Phase I reverted back to a collapsed state, as shown in Figure 4.9. This indicates that while the perturbation improves robustness in some cases, its behavior is not yet fully reliable. More systematic studies and refinements of the perturbation term are needed to stabilize its performance.

Equivariance Relaxation We also experimented with an equivariance relaxation technique proposed by [43], which aims to improve model training and prevent degenerate outputs by temporarily relaxing the strict equivariance constraints. In our implementation, we applied this idea to the GEBL layers by injecting an additional

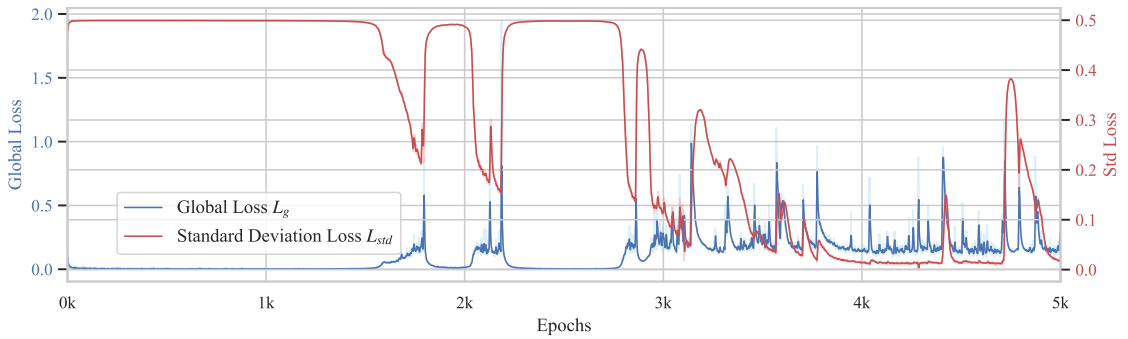

Figure 4.9: Global loss and standard deviation loss curve of GEBLNet, with configurations identical to B4 and a perturbation term, trained on a diagonal, trivial dataset, to learn the Chern number on a 5^2 grid, with 4 filled bands. During the training dynamics, the model fluctuated multiple times between valid and collapsed states, leading to an unstable performance.

non-equivariant term of the form

$$h_\delta(W)_k^\gamma = \sum_{\mu,\nu} \alpha_{\gamma\mu\nu} W_k^\mu \cdot W_k^\nu \,,$$

where the multiplication is applied pointwise, and $\alpha_{\gamma\mu\nu}$ are learnable coefficients. This non-equivariant component is scaled by the current epoch $\delta(t_{\text{epoch}})$, and gradually increases and then decreases over the course of training:

$$\delta(t_{epoch}) = \begin{cases} \delta_0 t_{epoch}, & t_{epoch} \in [1,\ 2500] \\ \delta_0(5000 - t_{epoch}), & t_{epoch} \in [2500,\ 5000] \end{cases}$$

with $\delta_0 = \frac{0.01}{5000} = 2 \times 10^{-6}$. Despite the theoretical motivation, we found that this method did not lead to successful learning when applied to trivial input data. The model failed to converge and give nonzero outputs. However, the idea remains a promising direction for future works, and may benefit from further tuning of both the schedule and the magnitude of δ_0.

Data Manipulation The general data sampling scheme leads to a uniform distribution with respect to the determinants, as shown in Section 2.7. On the other hand, with the diagonal data generation scheme introduced in Section 4.2, we are able to directly manipulate the distribution of local quantities. Specifically, the phases of the determinants of flux matrices. We attempted to wield this capability to guide the model toward learning local quantities by training it on biased datasets. However, this approach did not lead to successful learning outcomes.

Nevertheless, the scheme remains valuable for other purposes. For example, Figure 4.8 illustrates that the model tends to be inaccurate near the discontinuity at $-\pi$ and π, which may be attributed to the inherent continuity of the model. This observation suggests a potential direction: by concentrating training data around these discontinuities, it may be possible to improve the model's precision in these critical regions. A more detailed investigation is needed to explore this possibility.

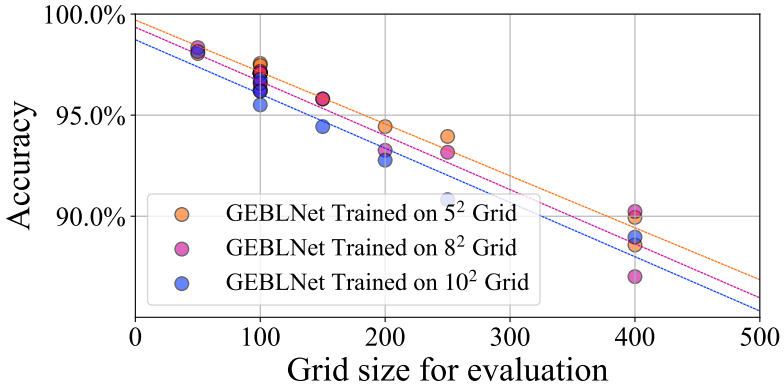

Figure 4.10: Comparison of model accuracy across different grid sizes. Each run, represented by markers of the same color, has identical configurations (B4), but is trained on grids of a different size. Each line represents a linear regression on the corresponding run.

4.4.2 Evaluation on Larger Grid

Due to its strictly local architecture, GEBLNet naturally supports inputs of arbitrary grid size, making it possible to learn on a rather small grid and predict Chern numbers over larger grids. We evaluate this generalization ability by training B4 on 5×5 grids and testing them on larger grids, such as 10×10, 15×10, and 20×20. As shown in Figure 4.10, the model demonstrates strong generalization performance: the accuracy decreases only moderately as grid size increases, and the drop appears approximately linear with respect to the number of grid sites. This is likely due to the accumulation of small local errors across the extended spatial domain.

To investigate whether this decrease in accuracy is caused by accumulated numerical errors, we also trained B4 on 8×8 and 10×10 grids using the same architecture. The resulting model did not outperform the 5×5-trained one when evaluated on the same test set. Furthermore, their rates of accuracy decay are nearly identical, suggesting similar local errors. This indicates that the observed performance degradation is not a result of insufficient training grid sizes, but rather reflects an inherent limitation in the model's capacity. In other words, the architecture itself lacks sufficient depth and width to accurately capture the global quantity.

This observation points to a future direction to explore: enhancing GEBLNet's expressivity with either larger model capacities or novel structures. Possible extensions include incorporating residual connections or Transformer-like structures that retain equivariance while enabling refined analysis of local quantities. Such architectural enhancements may allow the model to maintain high precision on significantly larger domains.

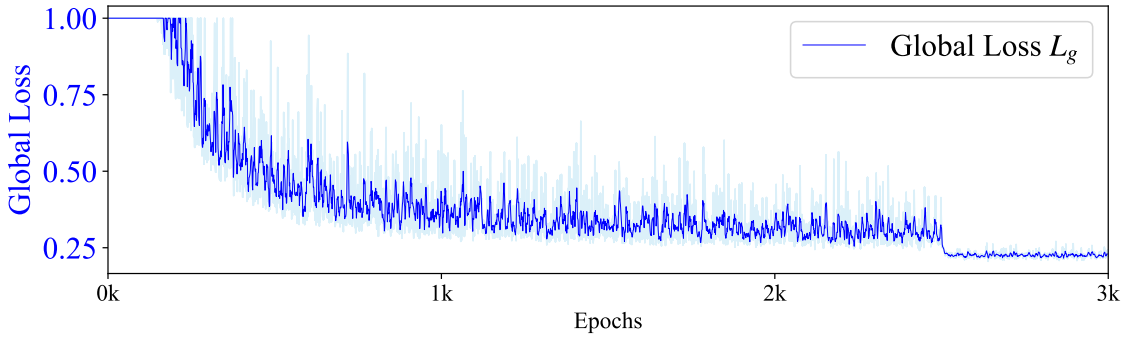

Figure 4.11: Global validation loss curve of B4, trained on a 3^4 grid, with 3 filled bands to learn the second order Chern number $\tilde{C}_2$.

4.5 Learning Higher Dimensional Chern numbers

We further extend the task to 4-dimensional grids, where the definition of the second order Chern number becomes significantly more complex than in the 2D case (see Section 2.4). In 4D, each lattice site is associated with $C_4^2 = 6$ independent Wilson loops, corresponding to all distinct plane directions. This increased combinatorial complexity not only enlarges the input space but also introduces more intricate geometric and topological dependencies across directions.

Moreover, unlike in the 2D case where the discretized Chern number is always an integer, the 4D discretization may yield non-integer values because of approximation errors in higher-order terms. As a result, standard classification accuracy is no longer an appropriate evaluation metric.

Instead, we adopt the mean absolute error (MAE) of the predicted global quantity, which corresponds to the global loss L_g. As shown in Figure 4.11, our model achieves an MAE of approximately 0.25, indicating that the predictions are well within acceptable bounds—often within rounding error of the true topological value.

In addition to global performance, we also investigate the model's ability to recover the correct local quantities. Figure 4.12 visualizes a comparison between the rescaled local outputs and the ground truth local contributions. The strong pointwise agreement between the predicted and true local densities further validates that the model is not only learning the correct global behavior but is also successfully identifying the underlying local structure that composes the higher-dimensional topological invariant. This illustrates the model's capacity to generalize to more complex topological settings beyond the 2D case.

An alternative strategy to learn Chern numbers accurately is to train on samples with special spatial features, and then to generalize to general samples. In particular, consider a product insulator whose Brillouin zone factorizes as

$$\mathrm{BZ} = \mathrm{BZ}_1 \times \mathrm{BZ}_2,$$

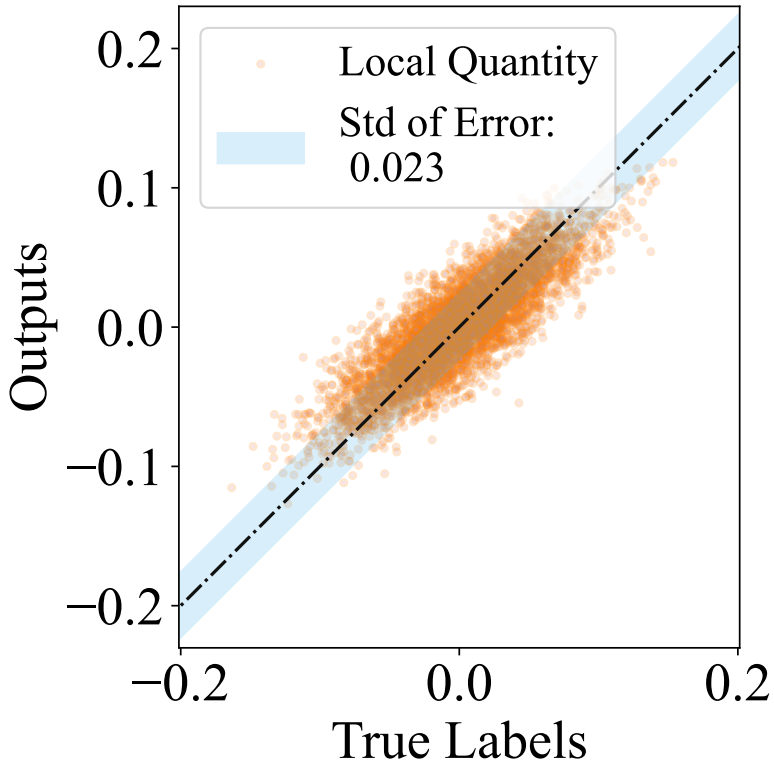

Figure 4.12: Comparison of rescaled local outputs with local true values.

with corresponding Bloch Hamiltonian

$$H(k_1, k_2) = H_1(k_1) \otimes H_2(k_2).$$

Its Berry connection then splits as

$$A_\mu(k_1, k_2) = \begin{cases} A^1_\mu(k_1) \otimes I_{d_2}, & \mu \text{ along } k_1, \\ I_{d_1} \otimes A^2_\mu(k_2), & \mu \text{ along } k_2, \end{cases}$$

and the second-order Chern number factorizes as

$$C = -2C_1C_2.$$

To generate such product-manifold samples for N bands, we randomly decompose $N = N_1 \times N_2$ and split the four momentum dimensions into two pairs. We then sample link variables (and compute Wilson loops) separately on each factor, assemble the full data via tensor product, and calculate the second-order Chern number accordingly, up to permutation over axes.

Training GEBLNet in the B4 configuration on this product dataset yields an accuracy of approximately 80% on the same product samples. However, when evaluated on general (non-product) configurations, performance degrades significantly. It remains an open question whether this failure reflects a capacity limitation of the current GEBLNet architecture or that of the discretized approximation for the second-order Chern number. A deeper investigation into both aspects is therefore required.

5

Conclusion

In this thesis, we introduced GEBLNet, a novel gauge-equivariant neural network architecture specifically designed to predict topological invariants, particularly Chern numbers, in multiband topological insulators.

Contributions

Our key contribution lies in two aspects. Theoretically, we propose a universal approximation theorem that rigorously guarantees GEBLNet's capacity to represent arbitrary first-order Chern numbers, given sufficiently expressive layers and appropriate training data. More generally, this model could learn any gauge-invariant function over compact matrix groups. Empirically, extensive experiments validated the theoretical result, demonstrating consistently high accuracy and strong generalization across diverse lattice sizes and band configurations.

In terms of model architecture, we constructed a novel trace normalization layer TrNorm. This stabilizes the training by canceling out the rapidly exploding outputs throughout layers, enabling the model to predict unseen data regimes by learning on only topologically trivial samples.

Furthermore, we extended the framework to higher-dimensional topological invariants, specifically, second-order Chern numbers on four-dimensional lattices. Experimental evidence showed that our architecture can learn these more complex invariants within reasonable approximation errors.

Limitations and Future Directions. Despite these promising results, several important limitations remain.

On the theoretical side, our universal approximation theorem currently applies only to first-order Chern numbers, leaving higher-order learning tasks without rigorous support. Moreover, the discrete numerical scheme employed for second-order

Chern numbers is not an exact integer-valued construction; developing an accurate and gauge-invariant discretization for higher-order invariants remains an open challenge.

Architecturally, our investigations were restricted to straightforward feed-forward stacks of gauge-equivariant bilinear (GEBL) layers. We have yet to explore potentially beneficial structural enhancements, such as residual connections, recurrent architectures, or transformer-based modules, which could improve training stability, efficiency, and model expressivity.

Finally, although we demonstrated the success of gauge-equivariant learning for multiband topological insulators, the applicability of our method to other Lie group equivariant problems remains untested.

Addressing these limitations will further enhance the capabilities and scope of our gauge equivariant neural networks.

Bibliography

[1] Michael M. Bronstein et al. *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*. Apr. 2021. arXiv: 2104.13478 [cs, stat]. (Visited on 04/29/2021).

[2] Jan E. Gerken et al. "Geometric Deep Learning and Equivariant Neural Networks". In: *Artificial Intelligence Review* (June 2023). ISSN: 1573-7462. DOI: 10.1007/s10462-023-10502-7. arXiv: 2105.13926. (Visited on 06/04/2023).

[3] Erik J. Bekkers et al. "Roto-Translation Covariant Convolutional Networks for Medical Image Analysis". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Ed. by Alejandro F. Frangi et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 440–448. ISBN: 978-3-030-00928-1. DOI: 10.1007/978-3-030-00928-1_50.

[4] Alexander Bogatskiy et al. "Lorentz Group Equivariant Neural Network for Particle Physics". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 992–1002. arXiv: 2006.04780.

[5] Alexandre Duval et al. *A Hitchhiker's Guide to Geometric GNNs for 3D Atomic Systems*. Dec. 2023. DOI: 10.48550/arXiv.2312.07511. arXiv: 2312.07511 [cs, q-bio]. (Visited on 12/13/2023).

[6] Giuseppe Carleo et al. "Machine learning and the physical sciences". In: *Reviews of Modern Physics* 91.4 (Dec. 2019). ISSN: 1539-0756. DOI: 10.1103/revmodphys.91.045002. URL: http://dx.doi.org/10.1103/RevModPhys.91.045002.

[7] Juan Carrasquilla. "Machine learning for quantum matter". In: *Advances in Physics: X* 5.1 (Jan. 2020), p. 1797528. ISSN: 2374-6149. DOI: 10.1080/23746149.2020.1797528. URL: http://dx.doi.org/10.1080/23746149.2020.1797528.

[8] Mario Krenn et al. "Artificial intelligence and machine learning for quantum technologies". In: *Phys. Rev. A* 107 (1 Jan. 2023), p. 010101. DOI: 10.1103/PhysRevA.107.010101. URL: https://link.aps.org/doi/10.1103/PhysRevA.107.010101.

[9] Anna Dawid et al. *Modern applications of machine learning in quantum sciences*. 2023. arXiv: 2204.04198 [quant-ph]. URL: https://arxiv.org/abs/2204.04198.

[10] Joel E Moore. "The birth of topological insulators". In: *Nature* 464.7286 (2010), pp. 194–198. DOI: 10.1038/nature08916.

[11] M. Z. Hasan and C. L. Kane. "Colloquium: Topological insulators". In: *Rev. Mod. Phys.* 82 (4 Nov. 2010), pp. 3045–3067. DOI: 10.1103/RevModPhys.82.3045. URL: https://link.aps.org/doi/10.1103/RevModPhys.82.3045.

[12] Qing Lin He et al. "Topological spintronics and magnetoelectronics". In: *Nature materials* 21.1 (2022), pp. 15–23. DOI: 10.1038/s41563-021-01138-5.

[13] Ling Lu, John D. Joannopoulos, and Marin Soljačić. "Topological photonics". In: *Nature Photonics* 8.11 (Nov. 2014), pp. 821–829. ISSN: 1749-4893. DOI: 10.1038/nphoton.2014.248. URL: https://doi.org/10.1038/nphoton.2014.248.

[14] Kyung-Hwan Jin et al. "Topological quantum devices: a review". In: *Nanoscale* 15.31 (July 2023). DOI: 10.1039/d3nr01288c.

[15] Paolo Zanardi and Mario Rasetti. "Holonomic quantum computation". In: *Physics Letters A* 264.2-3 (1999), pp. 94–99.

[16] Chetan Nayak et al. "Non-Abelian anyons and topological quantum computation". In: *Rev. Mod. Phys.* 80 (3 Sept. 2008), pp. 1083–1159. DOI: 10.1103/RevModPhys.80.1083. URL: https://link.aps.org/doi/10.1103/RevModPhys.80.1083.

[17] Carlo W. J. Beenakker. "Search for Majorana fermions in superconductors". In: *Annual Review of Condensed Matter Physics* 4 (2013), pp. 113–136. DOI: 10.1146/annurev-conmatphys-030212-184337. URL: https://doi.org/10.1146/annurev-conmatphys-030212-184337.

[18] Juan Carrasquilla and Roger G. Melko. "Machine learning phases of matter". In: *Nature Physics* 13.5 (Feb. 2017), pp. 431–434. ISSN: 1745-2481. DOI: 10.1038/nphys4035. URL: http://dx.doi.org/10.1038/nphys4035.

[19] Evert P. L. van Nieuwenburg, Ye-Hua Liu, and Sebastian D. Huber. "Learning phase transitions by confusion". In: *Nature Physics* 13.5 (Feb. 2017), pp. 435–439. ISSN: 1745-2481. DOI: 10.1038/nphys4037. URL: http://dx.doi.org/10.1038/nphys4037.

[20] Yanming Che et al. "Topological quantum phase transitions retrieved through unsupervised machine learning". In: *Phys. Rev. B* 102 (13 Oct. 2020), p. 134213. DOI: 10.1103/PhysRevB.102.134213. URL: https://link.aps.org/doi/10.1103/PhysRevB.102.134213.

[21] Mathias S. Scheurer and Robert-Jan Slager. "Unsupervised Machine Learning and Band Topology". In: *Phys. Rev. Lett.* 124 (22 June 2020), p. 226401. DOI: 10.1103/PhysRevLett.124.226401. URL: https://link.aps.org/doi/10.1103/PhysRevLett.124.226401.

[22] Pengfei Zhang, Huitao Shen, and Hui Zhai. "Machine Learning Topological Invariants with Neural Networks". In: *Physical Review Letters* 120.6 (Feb. 2018). ISSN: 1079-7114. DOI: 10.1103/physrevlett.120.066401. URL: http://dx.doi.org/10.1103/PhysRevLett.120.066401.

[23] Ning Sun et al. "Deep learning topological invariants of band insulators". In: *Phys. Rev. B* 98 (8 Aug. 2018), p. 085402. DOI: 10.1103/PhysRevB.98.085402. URL: https://link.aps.org/doi/10.1103/PhysRevB.98.085402.

[24] Oleksandr Balabanov and Mats Granath. "Unsupervised learning using topological data augmentation". In: *Phys. Rev. Res.* 2 (1 Mar. 2020), p. 013354. DOI: 10.1103/PhysRevResearch.2.013354. URL: https://link.aps.org/doi/10.1103/PhysRevResearch.2.013354.

[25] Oleksandr Balabanov and Mats Granath. "Unsupervised interpretable learning of topological indices invariant under permutations of atomic bands". In: *Machine Learning: Science and Technology* 2.2 (2020), p. 025008. DOI: 10.1088/2632-2153/abcc43.

[26] Marc Finzi, Max Welling, and Andrew Gordon Gordon Wilson. "A Practical Method for Constructing Equivariant Multilayer Perceptrons for Arbitrary Matrix Groups". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 3318–3328. URL: https://proceedings.mlr.press/v139/finzi21a.html.

[27] Rui Wang, Robin Walters, and Rose Yu. "Approximately Equivariant Networks for Imperfectly Symmetric Dynamics". In: *CoRR* abs/2201.11969 (2022). arXiv: 2201.11969. URL: https://arxiv.org/abs/2201.11969.

[28] Derek Lim et al. "Expressive Sign Equivariant Networks for Spectral Geometric Learning". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023. URL: https://openreview.net/forum?id=UWd4ysACo4.

[29] Hannah Lawrence and Mitchell Tong Harris. "Learning Polynomial Problems with SL(2, R)-Equivariance". In: *International Conference on Learning Representations*. 2023. URL: https://api.semanticscholar.org/CorpusID:265608700.

[30] Maurice Weiler et al. *Equivariant and Coordinate Independent Convolutional Networks. A Gauge Field Theory of Neural Networks*. 2023. DOI: 10.1142/14143.

[31] Taco S. Cohen et al. "Gauge Equivariant Convolutional Networks and the Icosahedral CNN". In: *arXiv:1902.04615 [cs, stat]* (Feb. 2019). arXiv: 1902.04615 [cs, stat]. (Visited on 06/13/2019).

[32] Pim de Haan et al. "Gauge Equivariant Mesh CNNs: Anisotropic Convolutions on Geometric Graphs". In: *arXiv:2003.05425 [cs, stat]* (Mar. 2020). arXiv: 2003.05425 [cs, stat]. (Visited on 03/12/2020).

[33] Gurtej Kanwar et al. "Equivariant Flow-Based Sampling for Lattice Gauge Theory". In: *Physical Review Letters* 125.12 (Sept. 2020), p. 121601. DOI: 10.1103/PhysRevLett.125.121601. (Visited on 10/18/2024).

[34] Denis Boyda et al. "Sampling Using SU(N) Gauge Equivariant Flows". In: *Physical Review D* 103.7 (Apr. 2021), p. 074504. DOI: 10.1103/PhysRevD.103.074504. (Visited on 10/18/2024).

[35] Kim A. Nicoli et al. "Estimation of Thermodynamic Observables in Lattice Field Theories with Deep Generative Models". In: *Physical Review Letters* 126.3 (Jan. 2021), p. 032001. DOI: 10.1103/PhysRevLett.126.032001. (Visited on 06/14/2023).

[36] Simone Bacchio et al. "Learning Trivializing Gradient Flows for Lattice Gauge Theories". In: *Physical Review D* 107.5 (Mar. 2023), p. L051504. DOI: 10.1103/PhysRevD.107.L051504. (Visited on 10/19/2024).

[37] Ryan Abbott et al. “Sampling QCD Field Configurations with Gauge-Equivariant Flow Models”. In: *Proceedings of The 39th International Symposium on Lattice Field Theory — PoS(LATTICE2022)*. Vol. 430. SISSA Medialab, Apr. 2023, p. 036. DOI: 10.22323/1.430.0036. arXiv: 2208.03832. (Visited on 01/16/2025).

[38] Di Luo et al. “Gauge Equivariant Neural Networks for Quantum Lattice Gauge Theories”. In: *Physical Review Letters* 127.27 (Dec. 2021), p. 276402. DOI: 10.1103/PhysRevLett.127.276402. arXiv: 2012.05232. (Visited on 01/16/2025).

[39] Matteo Favoni et al. “Lattice Gauge Equivariant Convolutional Neural Networks”. In: *Physical Review Letters* 128.3 (Jan. 2022), p. 032003. DOI: 10.1103/PhysRevLett.128.032003. arXiv: 2012.12901. (Visited on 03/29/2023).

[40] Brian C. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. 2nd. GTM 222. Theorem 12.20. Springer, 2015.

[41] Takahiro Fukui, Yasuhiro Hatsugai, and Hiroshi Suzuki. “Chern numbers in discretized Brillouin zone: efficient method of computing (spin) Hall conductances”. In: *Journal of the Physical Society of Japan* 74.6 (2005), pp. 1674–1677. DOI: 10.1143/JPSJ.74.1674.

[42] G. W. Stewart. “The Efficient Generation of Random Orthogonal Matrices with an Application to Condition Estimators”. In: *SIAM Journal on Numerical Analysis* 17.3 (1980), pp. 403–409. ISSN: 00361429. URL: http://www.jstor.org/stable/2156882 (visited on 05/16/2025).

[43] Stefanos Pertigkiozoglou et al. “Improving Equivariant Model Training via Constraint Relaxation”. In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 2024. URL: https://openreview.net/forum?id=tWkL7k1u5v.

DEPARTMENT OF MATHEMATICAL SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY