Haldane Bundles: A Dataset for Learning to Predict the Chern Number of Line Bundles on the Torus

Cody Tipton[1,*], **Elizabeth Coda**[2], **Davis Brown**[2], **Alyson Bittner**[2], **Jung Lee**[2], **Grayson Jorgenson**[2], **Tegan Emerson**[2,3,4], **Henry Kvinge**[1,2]
[1]Department of Mathematics, University of Washington
[2]Pacific Northwest National Laboratory
[3]Department of Mathematics, Colorado State University
[4]Department of Mathematical Sciences, University of Texas
cat1184@uw.edu, {first}.{last}@pnnl.gov

Abstract

Characteristic classes, which are abstract topological invariants associated with vector bundles, have become an important notion in modern physics with surprising real-world consequences. As a representative example, the incredible properties of topological insulators, which are insulators in their bulk but conductors on their surface, can be completely characterized by a specific characteristic class associated with their electronic band structure, the first Chern class. Given their importance to next generation computing and the computational challenge of calculating them using first-principles approaches, there is a need to develop machine learning approaches to predict the characteristic classes associated with a material system. To aid in this program we introduce the *Haldane bundle dataset*, which consists of synthetically generated complex line bundles on the 2-torus. We envision this dataset, which is not as challenging as noisy and sparsely measured real-world datasets but (as we show) still difficult for off-the-shelf architectures, to be a testing ground for architectures that incorporate the rich topological and geometric priors underlying characteristic classes.

1 Introduction

As a family of fundamental topological invariants of vector bundles, characteristic classes have long been a central tool in differential topology [16]. As is often the case, it was only after they had become an established tool in mathematics that their importance to physics was recognized in, for example, Chern-Simons theory [22, 6]. Another surprising appearance of characteristic classes is in topological materials, where they can be associated with the electronic band structure of a crystal lattice. Remarkably, non-trivial topology in this setting often has dramatic physical consequences. For example, topological insulators are topologically protected insulators in their interior but conductors on their surface (or edge) [11].

Because materials with topological properties are of intense interest in applications (e.g., next generation computing [13] and advanced sensors [14]), substantial effort has gone into finding them. While this hunt has seen some successes, first-principles approaches to calculating the topological characteristics of a material and then experimentally validating these predictions remain time-, labor-, and resource-intensive. There has thus been interest in leveraging machine learning as a cheap way to predict the topological properties of a material. Most approaches thus far have relied on off-the-shelf architectures and generic data-processing procedures that do not take advantage of the unique data-type that is a vector bundle. However, the problem of predicting the characteristic classes

[*]This work was performed during an internship at Pacific Northwest National Laboratory.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

of a material system is well-aligned with current research directions in the field of geometric deep learning (e.g., invariance to Lie group actions).

We believe that part of the challenge of developing methods of learning from vector bundles arises from the fact that there are no benchmark datasets where the bundle structure is actually exposed. Indeed, the features in large topological materials databases [1, 15] consist solely in the geometric structure and constituent atoms in the crystal lattice with associated atomic characteristics. While topological labels to these datasets ultimately depend on an underlying vector bundle structure, this is several layers beneath the features that are given. To create a dataset where researchers can develop architectures for learning on vector bundles, we introduce the Haldane Bundle dataset, a dataset of complex line bundles on the 2-torus with labeled first Chern number. Our process of generating valid random line bundles, which is mathematically non-trivial, is inspired by the Haldane model [8], an influential toy model for the anomalous quantum Hall effect. Our contributions in this paper include the following.

- A description of a method of sampling random line bundles on the 2-torus and our approach to efficiently calculating the Chern number of these line bundles.
- Summary statistics of the resulting Haldane Bundles dataset as well as off-the-shelf model performance.
- A discussion of some of the geometric and topological properties that a model designed for this task should have.

We hope that our work will begin the process of building a connection between the geometric deep learning community and scientists studying topological materials. Code to generate our datasets can be found at `https://github.com/shadtome/haldane-bundles`.

2 Related Work

There has been significant interest in using machine learning approaches for materials design and discovery [7]. In the realm of topological materials, several works have looked at predicting the topological properties of a material. For example, [3] and [18] both use tree-based methods that take as input a mix of categorical and numeric features like atomic properties. On the other hand, [23] look at the simple case of predicting topological properties of 1-dimensional insulators directly from a Hamiltonian. This amounts to calculating the winding number. An off-the-shelf convolutional neural network is used. Finally, [17] try to learn to predict the Hamiltonian which they then diagonalize to get the band structure. None of these papers learn directly from the underlying vector bundles and thus are not able to support development of models that incorporate the mathematical constructions beneath the topological properties of materials.

While vector bundles are a fundamental concept in both mathematics and physics, they have seen limited use in machine learning. Notable exceptions include [19], where a vector bundle framework is used for dimensionality reduction, [5, 4] where a more general fiber bundle framework is used to model many-to-one and many-to-many maps respective, and [12] where frames in a subbundle of the tangent bundle of data manifolds are used to better understand the learning process in computer vision models.

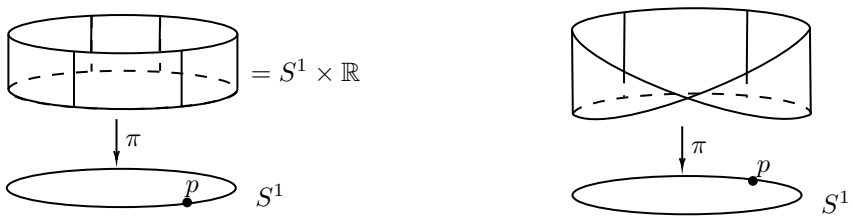
3 A Lightening Tour of Vector Bundles, Characteristic Classes, and the Haldane Model

3.1 Vector bundles

Vector bundles were introduced to provide a systematic way of studying tangent vectors, differential forms, and other geometric constructions on manifolds. Roughly speaking, a continuous vector bundle is a collection of vector spaces, one attached to each point on a manifold, which vary continuously. The standard example is the tangent bundle of a smooth manifold, which captures the space of tangent vectors as they vary from point to point. More formally, a *vector bundle of rank* n on a manifold M is a space E with a surjective map (the bundle *projection*) $\pi : E \to M$ such that for any $x \in M$, there is a neighborhood $U \subseteq M$ and a homeomorphism $\phi : \pi^{-1}(U) \to U \times \mathbb{R}^n$. π further satisfies a

commutative diagram that we omit here but that can be informally interpreted as saying that π maps $U \times \mathbb{R}^n$ to U. That is, π collapses all the points in the vector space $\mathbb{R}^n$. One consequence of this definition is that for any $x \in M$, $\pi^{-1}(x) = \mathbb{R}^n$. This space is known as the *fiber* at x. Vector bundles can be defined for both fields of $\mathbf{R}$ and fields over $\mathbf{C}$ and this choice has a substantial impact on the structure and properties of the vector bundle.

Vector bundles can have non-trivial topology introduced by "twists" in the way that the vector spaces (sitting above points in the manifold) change as one moves around the manifold. Line bundles on the circle provide a visualization of this phenomenon (see Figure 1). In both the left and right example, lines (1-dimensional vector spaces) are assigned to each point on the circle S^1 (shown in the bottom of the figure). In both cases, if one only looks at the points along a small interval of $(\alpha, \beta) \subset S^1$ (thought as an open subset of angles between α and β) along with their associated lines, the resulting space is topologically equivalent to $(\alpha, \beta) \times \mathbb{R}$. That is, the vector bundle is trivial locally. But looking at all of S^1 one sees that in the example on the right (which is a Möbius band), there is a twist such that if we start at a non-zero point on a fiber and then travel around the circle and come back to the same fiber, we will find ourself at a different point in that fiber. This shows that it is impossible to define a continuous function from the whole circle to the Möbius bundle that does not have any zeros.



(a) The tangent bundle over the circle

(b) Möbius bundle over the circle

Figure 1: There are only two non-isomorphic line bundles over the circle. The left is trivial, the right is not.

3.2 Characteristic classes

The example of the cylinder and the Möbius band in Figure 1 suggests that the extent to which vector bundles twist may be a useful global statistic that can be used to describe a bundle. This and similar notions of vector bundle shape are formalized in the concept of a characteristic class. Characteristic classes are topological invariants that capture global statistics of a vector bundle. They appear in a number of guises and are a fundamental tool in algebraic topology, algebraic geometry, differential geometry, and mathematical physics [16]. For example, the Stiefel-Whitney class is a characteristic class of real vector bundles that takes values in $\mathbf{Z}/2\mathbf{Z}$ and detects whether a vector bundle is orientable. Hence, the cylinder (Figure 1, left) has trivial Stiefel-Whitney class since it is orientable and the Möbius band has Stiefel-Whiteny class $\overline{1}$ since it is not orientable. The complex vector bundles in the Haldane Bundles dataset are labeled by a different characteristic called the first Chern class $c_1^{\#}$ which takes values in $\mathbf{Z}$ and detects how far a bundle is twisted from being trivial.

Even though they are a topological and not a geometric construction, the Chern class of a vector bundle can be calculated via an integral (over the manifold) of a specific differential form c_1 called the curvature form. In particular, if L is a line bundle, then

$$c_1^{\#}(L) = \int_M c_1(L).$$

One method of calculating the Chern number requires one to find a connection of the line bundle and compute its curvature. For more information about this see, [2].

Finally, it is important to note that a vector bundle is always trivial if it can be defined consistently with a global set of coordinates. This explains why in Section 4.1, we define line bundles in local patches which we then glue together. If we did not, every vector bundle we defined would be trivial (equivalent to $M \times \mathbf{R}$).

3.3 The Haldane Model

Our Haldane Bundles dataset is inspired by the Haldane model [8], which is a 2-dimensional toy model of a Chern insulator (a type of topological insulator whose properties arise from its electronic band structure having non-trivial Chern number). This simplified quantum system is defined over a honeycomb lattice (reminiscent of graphene). Due to the periodic nature of the lattice and Bloch's Theorem, the momentum space for this system can be viewed as sitting on the 2-torus, $\mathbf{T}^2$. Thus, the Hamiltonian is a function of points on $\mathbf{T}^2$, $H : \mathbf{T}^2 \to \text{Mat}(\mathbf{C})_{2,2}$. In particular, for $p \in \mathbf{T}^2$,

$$H(p) = \begin{bmatrix} G(p) & \overline{F}(p) \\ F(p) & -G(p) \end{bmatrix} \tag{1}$$

where $F(p) = t_1 \sum_{j=1}^{3} e^{a_i \cdot p}$, $G(p) = M + 2t_2 \sum_{j=1}^{3} \sin(b_i \cdot p)$, a_i are the vectors forming the honeycomb lattice, and the b_i are the second neighbor hopping vectors. One can build a $\mathbf{C}$-line bundle on $\mathbf{T}^2$ associated to H by looking at one of the eigenvectors of H. Note that besides p, several other parameters determine H including M, t_1, and t_2. In the case where $|t_2/M| < \frac{1}{3\sqrt{3}}$, the Chern number is zero. On the other hand, when $|t_2/M| > \frac{1}{3\sqrt{3}}$, we obtain non-zero Chern numbers. In the latter case we get a model of a topological insulator.

The Haldane model is an attractive starting point for development of methods of learning to predict Chern numbers from line bundles because different line bundles can be explicitly generated by varying M, t_1, and t_2 and based on these choices we know immediately what the Chern number is. Unfortunately, the resulting dataset is too simple for development of generalizable models. Indeed, because of the regularity in the resulting eigenvectors, a model can easily memorize predictive features without any generalization ability. This motivates our introduction of the Haldane Bundles dataset.

4 The Haldane Bundle Dataset

In this section, we describe how we construct the class of line bundles on the 2-torus that are the datapoints of the Haldane Bundles dataset. These line bundles have a range of nice computational and theoretical properties, are easily parameterized since they are fully determined by a pair of smooth functions G and F on $\mathbf{T}^2$, yet provide sufficient variation to create a challenging machine learning task.

4.1 Constructing Lots of Line Bundles

We begin by describing a generic and abstract algorithm for constructing line bundles on any smooth manifold M and then specify to $\mathbf{T}^2$. Pick a collection of smooth maps $\psi_\alpha : V_\alpha \to \mathbf{C}^{n+1} \setminus \{0\}$ for $\alpha \in A$ where A is some index set such that $\{V_\alpha\}_{\alpha \in A}$ covers the space M. The $\{\psi_\alpha\}_{\alpha \in A}$ induce smooth maps $\{\Psi_\alpha\}_{\alpha \in A}$ from each V_α to complex projective space $\mathbf{CP}^n$, a smooth $2n$-dimensional manifold where each point corresponds to a line in $\mathbf{C}^{n+1}$. This connection is defined by $\Psi_\alpha(p) = [\psi_\alpha(p)]$ where $[\psi_\alpha(p)]$ means the unique line traveling through the point $\psi_\alpha(p)$ and the origin.

The functions $\{\Psi_\alpha\}_{\alpha \in A}$ each independently assign lines to the points in their respective domains $\{V_\alpha\}_{\alpha \in A}$ of M. To define a single consistent assignment on M, we need to be able to glue the $\{\Psi_\alpha\}_{\alpha \in A}$ together. When $\{\Psi_\alpha\}_{\alpha \in A}$ are sufficiently consistent, we can do this by introducing some gluing maps $\tau_{\alpha,\beta} : V_\alpha \cap V_\beta \to \mathbf{C}^*$ such that $\tau_{\alpha,\beta}(p)\psi_\beta(p) = \psi_\alpha(p)$ for all $p \in V_\alpha \cap V_\beta$ and for all $\alpha, \beta \in A$. These gluing maps ensure that we can consistently translate between Ψ_α to get a well-defined line bundle where pairs of Ψ_α are both defined. Taken together, this construction gives a line bundle that is well-defined on all of M. Note that the existence of $\tau_{\alpha,\beta}$ is impossible if either $\psi_\beta(p)$ or $\psi_\beta(p)$ is zero and the other is not (this will motivate the construction in the next paragraph).

We construct our Haldane Bundles following this method. Let $G : M \to \mathbf{R}$, $F : M \to \mathbf{C}$ be smooth maps with no common zeros, and define $R, R^\dagger : M \to \mathbf{R}$ by $R(p) = G(p) + \sqrt{G(p)^2 + F(p)\overline{F(p)}}$ and $R^\dagger(p) = G(p) - \sqrt{G(p)^2 + F(p)\overline{F(p)}}$. Let $\psi : V \to \mathbf{C}^2 \setminus \{0\}$, $\psi^\dagger : V^\dagger \to \mathbf{C}^2 \setminus \{0\}$ be

defined by

$$\psi(p) = \begin{bmatrix} R(p) \\ F(p) \end{bmatrix} \qquad \psi^\dagger(p) = \begin{bmatrix} -\overline{F(p)} \\ R^\dagger(p) \end{bmatrix} \tag{2}$$

where V and $V^\dagger$ are the open subsets of M where ψ and $\psi^\dagger$ are not zero, respectively. Note that $V \cup V^\dagger = M$, since F and G have no common zeros. The smooth function $\tau : V \cap V^\dagger \to \mathbf{C}^*$ defined as $\tau = \frac{R^\dagger}{F}$ is a gluing function for the pair of maps ψ and $\psi^\dagger$ with $\tau(p)\psi(p) = \psi^\dagger(p)$ making the induced map $\Psi : M \to \mathbf{CP}^1$, where $\Psi|_V = [\psi]$ and $\Psi|_{V^\dagger} = [\psi^\dagger]$, a well-defined line bundle on M (note that F does not vanish on $V \cap V^\dagger$ by construction so τ is well-defined).

We call the pair of functions G and F a *Haldane pair*, and we will call the corresponding line bundle $L(G, F)$, the *Haldane bundle* with respect to G and F.

Theorem 1. *For any Haldane pair (G, F) on a 2-dimensional smooth manifold M, the Chern number for $L(G, F)$ is computed through the following integral*

$$c_1^{\#}(L(G,F)) = \frac{i}{2\pi}\left(\int_V d(\delta\overline{F}) \wedge d(\delta F) + \int_{V^\dagger \setminus (\overline{V \cap V^\dagger})} d(\delta^\dagger F) \wedge d(\delta^\dagger \overline{F})\right)$$

where $\delta = \frac{1}{\sqrt{2R(R-G)}}$ and $\delta^\dagger = \frac{1}{\sqrt{2R^\dagger(R^\dagger - G)}}$.

We provide a proof of Theorem 1 in Section B.1 of the Appendix.

4.2 Over the Torus

We can realize $\mathbf{T}^2$ as the product space $S^1 \times S^1$ and hence to find a function defined on $\mathbf{T}^2$ it suffices to find a function of two variables that is doubly periodic. We therefore use Fourier polynomials to construct our Haldane pair. We call a smooth function $f : \mathbf{T}^2 \to \mathbf{C}$ a *complex-valued Fourier polynomial* if for all $p \in \mathbf{T}^2$ it takes the form

$$f(p) = \frac{1}{2\pi} \sum_{k \in \mathbf{Z}^2} c_k e^{2\pi i (k \cdot p)}$$

where all but a finite number of the $c_k \in \mathbf{C}$ are zero and $(k \cdot p)$ is the ordinary dot product. Similarly, we call a function $g : \mathbf{T}^2 \to \mathbf{R}$ a real-valued Fourier polynomial if for all $p \in \mathbf{T}^2$ we have

$$g(p) = \frac{1}{2\pi} \sum_{k \in \mathbf{Z}^2} (a_k \cos(2\pi(k \cdot p)) + b_k \sin(2\pi(k \cdot p)))$$

where all but a finite number of the $a_k \in \mathbf{R}$ and $b_k \in \mathbf{R}$ are zero.

Because the torus is equivalent to the product $S^1 \times S^1$, we can represent real-valued functions f on the torus as $n \times n$ 2-dimensional arrays (generalized images) where $f(i, j) = f(i + n, j + n)$. Note that the condition that makes this different from standard images is that the array "wraps around" on both axes, so that the $(n + 1)$th column (respectively row) is equal to the 1st column (resp. row). This imposes a constraint on the boundary of "images" representing features on the torus.

We can approximate the Chern number on the torus using the formula 1 by partitioning the torus into small enough squares and approximating the integrand on these small patches. This is possible, since in local coordinates the integrand in the integral is just a sum and product of values of $F, \delta, \delta^\dagger$ and partial derivatives of these. Since our functions are Fourier polynomials, we can compute these easily on a GPU by using discrete convolutions with the coefficients. For more information on this see B.2 in the Appendix.

The distribution of Chern numbers that one gets depends on the degree of Fourier polynomials that are used. These distributions are visualized in Figure 2 for some small degrees. The time it took us to compute Chern number as a function of Fourier polynomial degree and resolution (of the grid on the torus) on a single GPU (Nvidia Tesla p100 @ 16GB) is shown in Figure 3.

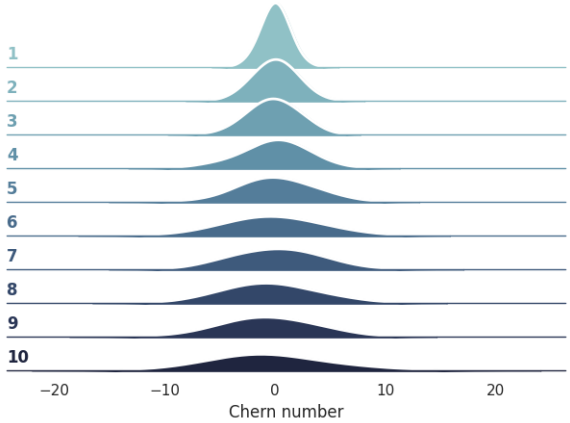


Figure 2: The distribution of Chern numbers for different degrees of Fourier polynomials. The vertical axis is the maximum degree of the polynomials.

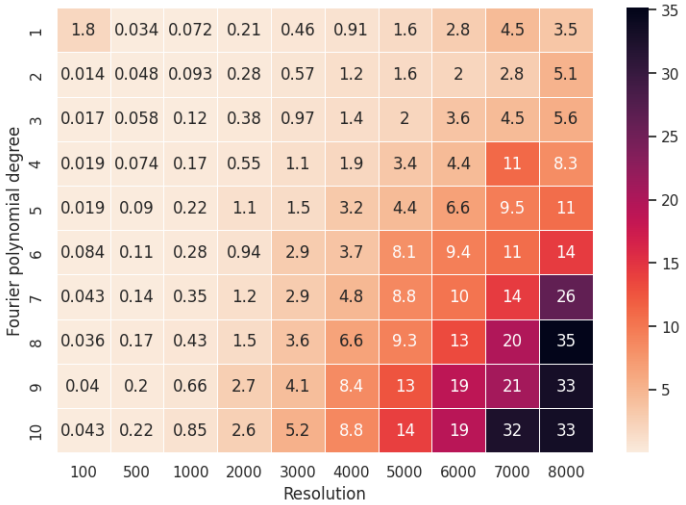


Figure 3: Time to compute the Chern number depending on the max degree of the polynomials and the partition size of the torus, in seconds.

4.3 Is the Haldane Model Enough for an Interesting Machine Learning Task?

Before describing the Haldane Bundles dataset described in Section 4, it is worth asking if all this is necessary. Indeed, might we generate a dataset sufficient for developing a machine learning capability to predict characteristic classes by simulating the simpler Haldane model (described in Section 3.3)? To test this we first fixed the form of G and F in the Hamiltonian in Equation 1 and varied the parameters M, t_1, and t_2 to obtain 10k distinct Hamiltonians. For each Hamiltonian, we sampled the corresponding eigenvector line bundles uniformly over $\mathbf{T}^2$ at a 32×32 resolution and compute the Chern number (based on the choice of M, t_1, and t_2). Because the line bundles are complex-valued, but high-performing off-the-shelf architectures are real-valued, we convert the complex eigenvector line bundles into separate real and imaginary channels.

On five random $80/20$ train/test splits for this dataset, a ResNet9 [9] achieves $99.8 \pm 0.2\%$ test accuracy, averaged across all runs. As derived in [21] from the Brouwer degree formula, the Chern number for the Haldane model can be determined by the sign of $G(p)$ in Equation 1 at the zeros of F. The zeros, or Dirac points, are highlighted in the top row of Figure 4, demonstrating a clear pattern that distinguishes between the different Chern numbers. A simple VanillaGrad saliency map [20] on the trained model shows that the model exploits this pattern to make its classification. That these points are not present/predictive for arbitrary line bundles yet off-the-shelf models perform with high

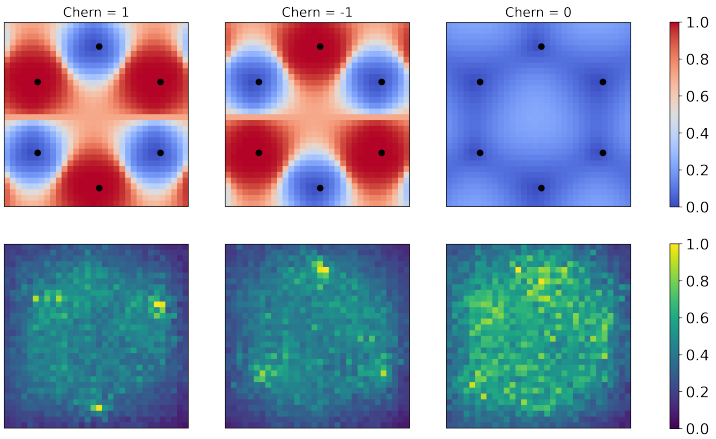


Figure 4: **Top:** A representative example of one channel of the eigenvector line bundle for the Haldane model for each of the three possible Chern numbers. The plots depict the real part of the first component of the eigenvector. The highlighted points are the zeros of F. **Bottom:** The saliency maps of a trained model, averaged over 100 examples of each class. Observe that for the nontrivial Chern classes, the highlighted regions are centered around the Dirac points.

accuracy suggests that this dataset has too many easily learnable incidental correlations that do not really get at the heart of what Chern numbers are.

4.4 Generating and Evaluating Machine Learning Models on the Haldane Bundle Dataset

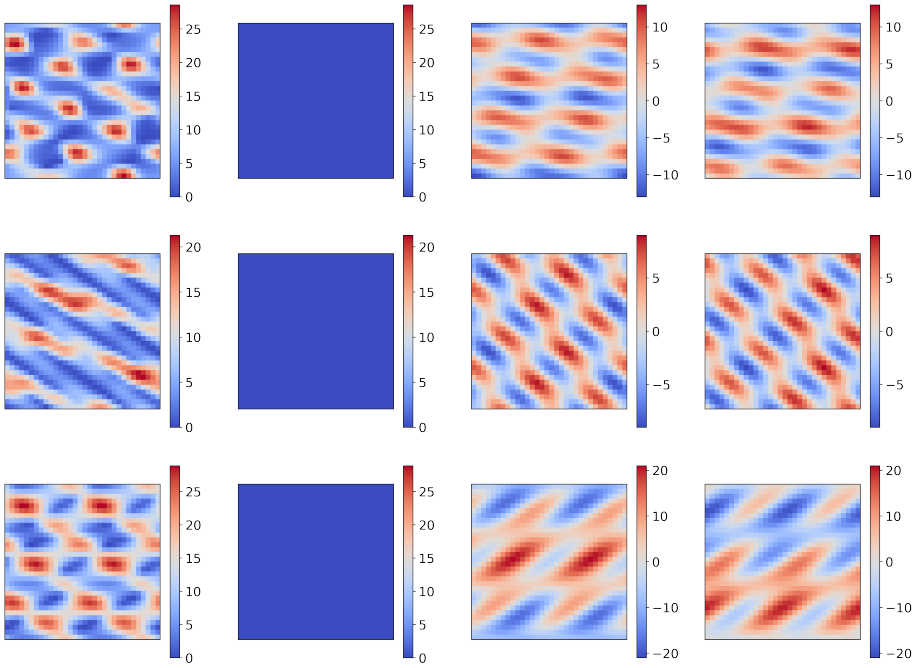


Figure 5: Three examples from the Haldane Bundle dataset with Chern numbers $1, 0$, and -2. From left to right the plots depict $\mathrm{Re}(R(p)), \mathrm{Im}(R(p)), \mathrm{Re}(F(p))$, and $\mathrm{Im}(F(p))$ sampled over $\mathbf{T}^2$ (where Re and Im are the real and imaginary part of a complex number.

To create a more challenging machine learning task, we use the framework from Section 4.1 to generate 50k Fourier polynomial Haldane pairs (G, F) by randomly sampling $a_k, b_k \in \mathbf{R}$ and $c_k \in \mathbf{C}$ for $k \in \mathbf{Z}^2$ such that $|k_1| \leq 4$ and $|k_2| \leq 4$. We also calculated the Chern number of each

Haldane pair. We downsample the generated dataset to obtain a class balanced dataset with 2.5k examples of each Chern number $\in \{-2, -1, 0, 1, 2\}$. In contrast to the dataset in the previous section where we fixed the form of G and F and varied parameters to obtain the dataset, here we generate a different form of G and F for each example. As a result, the zeros of F are not spatially fixed within the representation and there is not a trivial pattern of zeros for the model to exploit. Some examples from this dataset are visualized in Figure 5. Details for the actual calculation can be found in Section B.2 of the Appendix.

We train a Vision Transformer (ViT) [10], a ResNet9, and a ResNet9 with standard convolutions replaced by circular convolutions that wrap around to account for the fact that the line bundle is over the torus (see Section 4.2 for a discussion of this aspect of the data representation). We report model performance averaged over five random $80/20$ train/test splits in Table 1. The best test accuracy any of these standard off-the-shelf models achieved is 31.07%, illustrating the more challenging nature of this dataset and providing evidence that this is a task where some research into novel architectures is warranted.

Table 1: Benchmarking the Haldane Bundle Dataset on various off-the-shelf models. All accuracies are averaged over five random train/test splits and model initializations. Models are trained on normalized line bundles. 95% confidence intervals are included for each.

Architecture	Train Accuracy (%)	Test Accuracy (%)
ViT	100.0 ± 0.0	29.1 ± 1.16
ResNet	99.99 ± 0.02	26.57 ± 1.26
ResNet-C	100.0 ± 0.0	26.81 ± 1.03
Null Classifier	20.66 ± 0.07	21.36 ± 0.19

5 Topological and Geometric Priors Intrinsic to This Dataset

We want to end by highlighting some of the interesting geometric and topological features intrinsic to the challenge of trying to predict characteristic classes that may be attractive to the geometric deep learning community and other researchers that are interested in incorporating non-trivial mathematical ideas into deep learning frameworks.

Complex-valued deep learning: The applications that motivated this research (topological materials) and other physics applications where characteristic classes are important all work over the complex numbers. While there is work in complex-valued deep learning, the vast majority of research (including almost all of the science of deep learning) is over $\mathbf{R}$. On the other hand, Chern class prediction is fundamentally connected to the complex numbers. While our baselines incorporated complex numbers in a naive way, it would be interesting to understand whether deeper integration of complex numbers into the networks would improve performance.

The Geometry of the Underlying Manifold: Characteristic classes are calculated over manifolds. In this case, that manifold is the 2-torus whose symmetries can be incorporated into a CNN with only minor modification. On the other hand, there may be cases where characteristic classes need to be calculated over more complicated spaces. One can imagine such a setting being an ideal application for recently developed methods from geometric deep learning that can capture the structure of non-Euclidean spaces.

$U(1)$**-invariance:** The data stored over any point in a vector bundle is a vector space. An n-dimensional $\mathbf{F}$-vector space is invariant to actions of the group $GL_n(\mathbf{F})$. That is, if one applies $g \in GL_n(\mathbf{F})$ to V, one gets other elements of V. This symmetry becomes especially important in applications, like data science, where a vector space V needs to be represented concretely by, for example, a basis $b_1, \ldots, b_n$. In the line bundle case of the Haldane Bundles dataset, the line l is represented by a non-zero vector v that l passes through it. It is understood by the mathematician that for any non-zero $a \in \mathbf{F}$, cv represents the same line, but this is likely not true for the model. It may be interesting to think about whether specialized augmentation or equivariant architectures might help build a more robust model.

Global properties from local characteristics: Topological characteristics are intrinsically global descriptors of a space. That is, they can rarely be consistently inferred if one only examines part of a

space. While at its best, deep learning is powerful precisely because it can extract high-level features by aggregating low-level ones, it is also known to catastrophically fail by focusing on spurious correlations. Developing methods of nudging a model away from single feature correlations is critical in the problem of predicting characteristic classes and could be usefully applied in a host of other problems.

6 Conclusion

In this paper we introduced the dataset Haldane Bundles for the purpose of developing better machine learning techniques of predicting the characteristic classes of a vector bundle. While our ultimate motivation comes from the importance of characteristic classes in a range of applications (most importantly, topological materials), we also aim to show that this is an interesting problem from the perspective of incorporating sophisticated mathematical ideas into deep learning.

Acknowledgments and Disclosure of Funding

This research was supported by the Mathematics for Artificial Reasoning in Science (MARS) initiative at Pacific Northwest National Laboratory. It was conducted under the Laboratory Directed Research and Development (LDRD) Program at at Pacific Northwest National Laboratory (PNNL), a multiprogram National Laboratory operated by Battelle Memorial Institute for the U.S. Department of Energy under Contract DE-AC05-76RL01830.

References

[1] Barry Bradlyn, Luis Elcoro, Jennifer Cano, Maia G Vergniory, Zhijun Wang, Claudia Felser, Mois I Aroyo, and B Andrei Bernevig. Topological quantum chemistry. *Nature*, 547(7663):298–305, 2017.

[2] Shiing-shen Chern. *Complex Manifolds with out Potential Theory*. Springer, New York, 1979.

[3] Nikolas Claussen, B Andrei Bernevig, and Nicolas Regnault. Detection of topological materials with machine learning. *Physical Review B*, 101(24):245117, 2020.

[4] Elizabeth Coda, Nico Courts, Colby Wight, Loc Truong, WoongJo Choi, Charles Godfrey, Tegan Emerson, Keerti Kappagantula, and Henry Kvinge. Fiber bundle morphisms as a framework for modeling many-to-many maps. In *Topological, Algebraic and Geometric Learning Workshops 2022*, pages 79–85. PMLR, 2022.

[5] Nico Courts and Henry Kvinge. Bundle networks: Fiber bundles, local trivializations, and a generative approach to exploring many-to-one maps. *arXiv preprint arXiv:2110.06983*, 2021.

[6] Daniel S Freed. Classical chern-simons theory, part 1. *arXiv preprint hep-th/9206021*, 1992.

[7] JE Gubernatis and T Lookman. Machine learning in materials design and discovery: Examples from the present and suggestions for the future. *Physical Review Materials*, 2(12):120301, 2018.

[8] F Duncan M Haldane. Model for a quantum hall effect without landau levels: Condensed-matter realization of the" parity anomaly". *Physical review letters*, 61(18):2015, 1988.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 06 2016.

[10] Alexander Kolesnikov, Alexey Dosovitskiy, Dirk Weissenborn, Georg Heigold, Jakob Uszkoreit, Lucas Beyer, Matthias Minderer, Mostafa Dehghani, Neil Houlsby, Sylvain Gelly, Thomas Unterthiner, and Xiaohua Zhai. An image is worth 16x16 words: Transformers for image recognition at scale. 2021.

[11] Panagiotis Kotetes. *Topological insulators*. Morgan & Claypool Publishers, 2019.

[12] Henry Kvinge, Grayson Jorgenson, Davis Brown, Charles Godfrey, and Tegan Emerson. Neural frames: A tool for studying the tangent bundles underlying image datasets and how deep learning models process them. *arXiv preprint arXiv:2211.10558*, 2022.

[13] Henry F Legg, Matthias Rößler, Felix Münning, Dingxun Fan, Oliver Breunig, Andrea Bliesener, Gertjan Lippertz, Anjana Uday, AA Taskin, Daniel Loss, et al. Giant magnetochiral anisotropy from quantum-confined surface states of topological insulator nanowires. *Nature Nanotechnology*, 17(7):696–700, 2022.

[14] Bin Liu, Wuyuan Xie, Han Li, Yanrong Wang, Daoping Cai, Dandan Wang, Lingling Wang, Yuan Liu, Qiuhong Li, and Taihong Wang. Surrounding sensitive electronic properties of bi2te3 nanoplates—potential sensing applications of topological insulators. *Scientific Reports*, 4(1):4639, 2014.

[15] Antimo Marrazzo, Marco Gibertini, Davide Campi, Nicolas Mounet, and Nicola Marzari. Relative abundance of z 2 topological order in exfoliable two-dimensional insulators. *Nano letters*, 19(12):8431–8440, 2019.

[16] John Willard Milnor and James D Stasheff. *Characteristic classes*. Number 76. Princeton university press, 1974.

[17] Vittorio Peano, Florian Sapper, and Florian Marquardt. Rapid exploration of topological band structures using deep learning. *Physical Review X*, 11(2):021052, 2021.

[18] Gabriel R Schleder, Bruno Focassio, and Adalberto Fazzio. Machine learning for materials discovery: Two-dimensional topological insulators. *Applied Physics Reviews*, 8(3), 2021.

[19] Luis Scoccola and Jose A Perea. Fibered: Fiberwise dimensionality reduction of topologically complex data with vector bundles. In *39th International Symposium on Computational Geometry (SoCG 2023)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2023.

[20] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*, 2014.

[21] Doru Sticlet, Frederic Pié chon, Jean-Noël Fuchs, Pavel Kalugin, and Pascal Simon. Geometrical engineering of a two-band chern insulator in two dimensions with arbitrary topological index. *Physical Review B*, 85(16), apr 2012.

[22] Edward Witten. Quantum field theory and the jones polynomial. *Communications in Mathematical Physics*, 121(3):351–399, 1989.

[23] Pengfei Zhang, Huitao Shen, and Hui Zhai. Machine learning topological invariants with neural networks. *Physical review letters*, 120(6):066401, 2018.

A Further Constructions of Haldane Bundles

Whenever one obtains a class of line bundles, one can start taking tensor products to obtain even more varied line bundles. Furthermore, the Chern number of a tensor product of line bundles is just the sum of the Chern numbers of each piece. This gives us a way to calculate the Chern number of the tensor product of Haldane bundles, for which we already have a way to calculate.

First, we will generalize a Haldane pair, to a sequence to represent the individual tensor products.

Definition 1. *A Haldane sequence is a sequence of Haldane pairs* $Y = ((G_1, F_1), \ldots, (G_n, F_n))$*, and the corresponding Line bundle is*

$$L(Y) = L(G_1, F_1) \otimes \cdots \otimes L(G_n, F_n).$$

By theorem 1 and the fact that Chern number is additive over tensor products, we can calculate the Chern number for any Haldane sequence $Y = ((G_1, F_1), \ldots, (G_n, F_n))$ as

$$c_1^{\#}(Y) = \sum_{j=1}^{n} c_1^{\#}(L(G_j, F_j)).$$

These type of line bundles we do not use in our data sets, but we do include them in our code. They give a way to get larger Chern number by using smaller degrees. But, to represent the data for models to learn on they can be more complicated as they have more components since these line bundles are represented by the smooth maps $\Psi : M \to \mathbf{CP}^{2^n-1}$ for $n \geq 1$.

B Proofs

B.1 Proof of Theorem 1

For this section, we will give a short construction of the curvature form used in computing the Chern number and give a proof to theorem 1 .

Definition 2. *A* pre-line bundle *of degree n is a triple $X = (V_\alpha, \psi_\alpha, \tau_{\alpha,\beta})_{\alpha,\beta\in A}$ on some arbitrary index set A consisting of the following properties:*

- *the maps $\psi_\alpha : V_\alpha \to \mathbf{C}^{n+1} \setminus \{0\}$ and $\tau_{\alpha,\beta} : V_\alpha \cap V_\beta \to \mathbf{C}^*$ are smooth for all $\alpha, \beta \in A$,*
- *for each $\alpha, \beta \in A$, $\psi_\alpha(p) = \tau_{\alpha,\beta}(p)\psi_\beta(p)$ for all $p \in V_\alpha \cap V_\beta$ and all $\alpha, \beta \in A$,*
- *and $(V_\alpha)_{\alpha\in A}$ covers M.*

As in section 4.1, if $X = (V_\alpha, \psi_\alpha, \tau_{\alpha,\beta})$ is a pre-line bundle, we can build a smooth map $\Psi_X : M \to \mathbf{CP}^n$ with $\Psi(p) = [\psi_\alpha(p)]$ for $p \in V_\alpha$. Let L_n be the tautological line bundle on $\mathbf{CP}^n$ with

$$L_n = \{(p, v) \ : \ p \in \mathbf{CP}^n, \ \ v \in l_p\}$$

where l_p is the 1-dimensional vector space spanned by a representative of p. Then the corresponding line bundle is defined as $L(X) := \Phi_X^*(L_n)$.

To build our curvature, we first need to build a connection on $L(X)$, and we need to make sure our maps $\psi_\alpha, \tau_{\alpha,\beta}$ are normalized. We say that a pre-line bundle $X = (V_\alpha, \psi_\alpha, \tau_{\alpha,\beta})$ of degree n is normalized if $|\psi_\alpha(p)| = 1$ for all $p \in V_\alpha$ and $|\tau_{\alpha,\beta}(p)| = 1$ in the complex plane for all $p \in V_\alpha \cap V_\beta$ and for all $\alpha, \beta \in A$.

lemma 2. *Let $X = (V_\alpha, \Psi_\alpha, \tau_{\alpha,\beta})_{\alpha\in A}$ be a complex line bundle of degree n. For each V_α, there is a local frame $\Sigma_\alpha : V_\alpha \to L(X)$ such that on $V_\alpha \cap V_\beta$ we have the following transformation law*

$$\Sigma_\alpha = \tau_{\alpha,\beta}\Sigma_\beta$$

for all $\alpha, \beta \in A$.

Proof. Let $\Psi_X : M \to \mathbf{CP}^n$ be the induced smooth map from the pre-line bundle X. For each $\alpha \in A$, define

$$\Sigma_\alpha(p) = (p, \Psi(p), (\psi_\alpha^0(p), \ldots, \psi_\alpha^n(p)))$$

for $p \in V_\alpha$, which is a smooth frame on V_α. It is easy to see that we have

$$\begin{aligned}\tau_{\alpha,\beta}\Sigma_\beta &= (p, \Psi(p), \tau_{\alpha,\beta}(\psi_\beta^0(p), \ldots, \psi_\beta^n(p)) \\ &= (p, \Psi(p), (\psi_\alpha^0(p), \ldots, \psi_\alpha^n(p))) \\ &= \Sigma_\alpha.\end{aligned}$$

This completes the proof. □

Next, we will define our connection form on each of the local sections V_α and show that they transform in the expected way for connections. For this definition, we define $\Omega_M^n(U, \mathbf{C})$ to be the space of complex differential n-forms on an open subset U of M.

Definition 3. *Let $X = (V_\alpha, \psi_\alpha, \tau_{\alpha,\beta})_{\alpha,\beta\in A}$ be a normalized complex line bundle of degree n on a smooth manifold M. For each V_α, define the 1-form $\omega_\alpha \in \Omega_M^1(V_\alpha, \mathbf{C})$ as*

$$\omega_\alpha = \sum_{r=1}^{n} \overline{\psi_\alpha^r} d\psi_\alpha^r$$

for all $\alpha \in A$.

lemma 3. *Let $X = (V_\alpha, \Psi_\alpha, \tau_{\alpha,\beta})_{\alpha,\beta\in A}$ be a normalized complex line bundle of degree n over a smooth manifold M. The collection of one forms ω_α^B on each local trivialization V_α defines a global connection $\nabla : \Gamma(L(X) \to \Gamma((T^*M)_{\mathbf{C}} \otimes_{\mathbf{C}} L(X))$ with*

$$\nabla(\Sigma_\alpha) = \omega_\alpha \otimes \Sigma_\alpha$$

on each open subset V_α.

Proof. We need to show that the collection of 1-forms ω_α transform like a connection. On $V_\alpha \cap V_\beta$, we have

$$\begin{aligned}
\omega_\alpha &= \sum_{r=1}^{n} \overline{\psi_\alpha^r} d\psi_\alpha^r \\
&= \sum_{r=1}^{n} \overline{\tau}_{\alpha,\beta} \overline{\psi_\beta^r} d(\tau_{\alpha,\beta}\psi_\beta^r) \\
&= \sum_{r=1}^{n} \overline{\tau}_{\alpha,\beta} \overline{\psi_\beta^r} \tau_{\alpha,\beta} d\psi_\beta^r + \sum_{r=1}^{n} \overline{\tau}_{\alpha,\beta} \overline{\psi_\beta^r} \psi_\alpha^r d\tau_{\alpha,\beta} \\
&= \sum_{r=1}^{n} \overline{\psi_\beta^r} d\psi_\beta^r + \sum_{r=1}^{n} (\overline{\psi_\beta^r}\psi_\beta^r) \overline{\tau}_{\alpha,\beta} d\tau_{\alpha,\beta} \\
&= \omega_\beta + \overline{\tau}_{\alpha,\beta} d\tau_{\alpha,\beta}.
\end{aligned}$$

Multiplying both sides by $\tau_{\alpha,\beta}$ will get the result. □

For each α, we can compute $\Omega_\alpha = d\omega_\alpha \in \Omega^2_M(V_\alpha, \mathbf{C})$, and these glue together to make a global curvature form Ω on M. This shows that the curvature is of the form

$$\Omega|_{V_\alpha} = \sum_{r=0}^{n} d\overline{\psi_\alpha^r} \wedge d\psi_\alpha^r$$

for each $\alpha \in A$. By Chern-Weil theory the curvature gives us a Chern class $c_1(X) = [\frac{i\Omega}{2\pi}] \in H^2_{dR}(M; \mathbf{C})$ and hence gives us the chern number

$$c_1^{\#}(X) = \frac{i}{2\pi} \int_M \Omega$$

whenever M is a 2-dimensional manifold.

Now that we have our curvature form, we can use this to give the proof of theorem 1.

proof of theorem 1. Let (G, F) be a Haldane pair with our smooth maps $\psi : V \to \mathbf{C}^2 \setminus \{0\}$ and $\psi^\dagger : V^\dagger \to \mathbf{C}^2 \setminus \{0\}$ defined as in 4.1. We need to normalize these vectors to use our connection in definition 3, and it is not too hard to show that $|\psi| = \sqrt{2R(R-G)}$ and $|\psi^\dagger| = \sqrt{2R^\dagger(R^\dagger - G)}$. We define

$$\widetilde{\psi} = \delta\psi \qquad\qquad \widetilde{\psi^\dagger} = \delta^\dagger\psi^\dagger$$

with $\delta = \frac{1}{\sqrt{2R(R-G)}}$ and $\delta^\dagger = \frac{1}{\sqrt{2R^\dagger(R^\dagger - G)}}$..

Now that we normalized our vectors, we can use these to compute our curvature. On V, we have

$$\begin{aligned}
\Omega|_V &= d(\overline{\delta R}) \wedge d(\delta R) + d(\overline{\delta F}) \wedge d(\delta F) \\
&= d(\delta\overline{F}) \wedge d(\delta F) \\
&= \delta^2 d\overline{F} \wedge F + \delta F d\overline{F} \wedge d\delta + \delta\overline{F} d\delta \wedge dF
\end{aligned}$$

using the fact that R and δ are real and the product rule for the exterior derivative. Similarly, we obtain

$$\begin{aligned}
\Omega|_{V^\dagger} &= d(\delta^\dagger F) \wedge d(\delta^\dagger \overline{F}) \\
&= (\delta^\dagger)^2 dF \wedge d\overline{F} + \delta^\dagger F d\delta^\dagger \wedge d\overline{F} + \delta^\dagger \overline{F} dF \wedge d\delta^\dagger.
\end{aligned}$$

We can break our integral apart on the two open subset V and $V^\dagger \setminus \overline{V \cap V^\dagger}$ to get the result. □

B.2 Computation of Chern number over the Torus

In the case where (G, F) is a Haldane pair over $\mathbf{T}^2$ consisting of Fourier polynomials, we can approximate the Chern number using the formula in theorem 1 in a very efficient manner on the GPU. First, we need to give a description on how we save our data of Fourier polynomials in the computer, as this will be important for computation later. Let

$$F(p) = \frac{1}{2\pi} \sum_{k \in \mathbf{Z}^2} c_k e^{2\pi i (k \cdot p)} \tag{3}$$

for $c_k \in \mathbf{C}$ where all but a finite number of them are nonzero, and let

$$G = \frac{1}{2\pi} \sum_{k \in \mathbf{Z}^2} (a_k \cos(2\pi(k \cdot p)) + b_k \sin(2\pi(k \cdot p))) \tag{4}$$

for $a_k, b_k \in \mathbf{R}$, where all but a finite number of them are zero.

Next, we need to describe some bounds to when the coefficients are zero. Let $N_-(F)$ and $N_+(F)$ be positive integers such that $N_-(F)$ is the minimal integer such that $c_k = 0$ for all $k_1 < -N_-(F)$ and $k_2 < -N_-(F)$, and define $N_+(F)$ to be the minimal positive integer such that $c_k = 0$ for all $k_1 > N_+(F)$ and $k_2 > N_+(F)$. Similarly, we can define $N_-(G)$ and $N_+(G)$ as well applied to both coefficients a_k and b_k. Let $N_F = N_-(F) + N_+(F) + 1$ and construct a matrix A_F of size $N_F \times N_F$ with entries $(A_F)_{i,j} = c_{(i-N_-(F), j-N_-(F))}$ for $1 \leq i, j \leq N_F$. Similarly, define $N_G = N_-(G) + N_+(G) + 1$ and define the matrices B_G^t of size $N_G \times N_G$ for $t = 1, 2$ with

$$(B_G^1)_{i,j} = a_{(i-N_-(G), j-N_-(G)} \qquad (B_G^2)_{i,j} = b_{(i-N_-(G), j-N_-(G)} \tag{5}$$

for $1 \leq i, j \leq N_G$. For example, if $N_-(F) = N_+(F) = d$, then A_F is of the form

$$\begin{bmatrix} c_{(-d,-d)} & \cdots & c_{(0,-d)} & \cdots & c_{(d,-d)} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ c_{(-d,0)} & \cdots & c_{(0,0)} & \cdots & c_{(d,0)} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ c_{(-d,d)} & \cdots & c_{(0,d)} & \cdots & c_{(d,d)} \end{bmatrix}. \tag{6}$$

Example 1. *Here is a simple example, that is related to the Haldane model. Suppose we have a Haldane pair (G, F) with the Fourier polynomials*

$$F(p) = t_1 e^{2\pi i(x+y)} + t_1 e^{2\pi i(-y)} + t_1 e^{2\pi i(-x)}$$
$$G(p) = M + 2t_2 \sin(-2\pi x + 2\pi y) + 2t_2 \sin(4\pi x + 2\pi y) + 2t_2 \sin(-2\pi x - 4\pi y)$$

with parameters $t_1, t_2, M \in \mathbf{R}$. Then then $N_-(F) = N_+(F) = 1$ and $N_-(G) = N(G, -) = 2$ and we would have the following matrices for the coefficients:

$$A_F = \begin{bmatrix} 0 & t_1 & 0 \\ t_1 & 0 & 0 \\ 0 & 0 & t_1 \end{bmatrix} \qquad B_G^1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & M & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \; B_G^2 = \begin{bmatrix} 0 & 2t_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2t_2 & 0 & 0 & 2t_2 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

With our representation of our Fourier polynomials through square matrices containing their coefficients, we can start describing how we calculated the Chern number using theorem 1. We proceed in the usual method by partitioning our torus in to small enough squares and evaluate our Ω on each of the sampled points of each square, multiplying by the area of that square, and then taking the sum to approximate the integral. The most expensive part of all of this is finding the values of F,G, and the partial derivatives at each of the sampled points and adding up the values. As the partition size of the torus grows, the more computationally expensive this becomes. To handle this, it is fruitful to calculate the values of F,G and the partial derivatives using the parallel capabilities of GPU's.

First, partition our torus into $1/N \times 1/N$ sized squares for some large enough $N > 1$, so that there are N^2 boxes in total, and we have sampled a point in each of these boxes, say the bottom

left corner of each of them. Let $P = [0, \ldots, i/N, \ldots, (N-1)/N]$ be a vector with N entries essentially representing either the first or second entry of a sampled point in the torus. Furthermore, let $K_F = [-N_-(F), \ldots, 0, \ldots, N_+(F)]$ and $K_G = [-N_-(G), \ldots, 0, \ldots, N_+(G)]$ be two vectors representing the possible entries for $k \in \mathbf{Z}^2$ where a_k, b_k, c_k are possibly non-zero. We will be using these vectors with certain operations to make it possible to implement the computation on the GPU.

Before we start the computation, we will review the Kronecker product, Kronecker sum of two vectors, and a new operation that combines these together which makes it possible to evaluate the sampled points on the Torus using the GPU.

Definition 4. *Let $v = [v_1, \ldots, v_n], h = [h_1, \ldots, h_m]$ two vectors with components in an arbitrary field $\mathbf{F}$. The Kronecker product of v and h is the vector $w = Kron(v \otimes h)$ of length nm with*

$$w = [v_1h_1, v_1h_2, \ldots, v_1h_m, \ldots, v_nh_1, \ldots, v_nh_m] \tag{7}$$
$$= [v_1h, \ldots, v_nh]. \tag{8}$$

The Kronecker product is essentially a ordered list of all possible multiplications of elements from v and h. Note that $Kron : \mathbf{F}^n \otimes \mathbf{F}^m \to \mathbf{F}^{nm}$ is a linear map over $\mathbf{F}$ and it is not commutative.

Next, is the Kronecker sum, which is similar to the Kronecker product, except it calculates all possible sums between two vectors and arranges it into a matrix.

Definition 5. *Let $v = [v_1, \ldots, v_n]$ and $h = [h_1, \ldots, h_m]$ be two vectors with components in an arbitrary field $\mathbf{F}$. The Kronecker sum of v and h is the matrix $C = Kron_\Sigma(v, h)$ defined as follows. Let A be a matrix of size $m \times n$ with each row is the vector v as in*

$$A = \begin{bmatrix} v \\ \vdots \\ v \end{bmatrix}. \tag{9}$$

Similarly, let B be a matrix of size $m \times n$ with each column is the vector h^T as in

$$B = \begin{bmatrix} h^T & \cdots & h^T \end{bmatrix}. \tag{10}$$

We define C to be the $m \times n$ matrix

$$C = A + B = \begin{bmatrix} v_1 + h_1 & v_2 + h_1 & \cdots & v_n + h_1 \\ v_1 + h_2 & v_2 + h_2 & \cdots & v_n + h_2 \\ \vdots & \vdots & \vdots & \vdots \\ v_1 + h_m & v_2 + h_m & \cdots & v_n + h_m \end{bmatrix}. \tag{11}$$

The Kronecker sum $Kron_\Sigma : \mathbf{F}^n \times \mathbf{F}^m \to Mat_{m,n}(\mathbf{F})$ is linear over $\mathbf{F}$ and it is also non-commutative.

With these two operations, we define a new linear map mdot : $\mathbf{F}^n \otimes \mathbf{F}^n \to Mat_{nm}(\mathbf{F})$ defined for $v \in \mathbf{F}^n$ and $h \in \mathbf{F}^m$ as

$$\text{mdot}(v \otimes h) = Kron_\Sigma(Kron(v, h), Kron(v, h)) \tag{12}$$

$$= \begin{bmatrix} v_1h_1 + v_1h_1 & \cdots & v_1h_m + v_1h_1 & \cdots & v_nh_1 + v_1h_1 & \cdots & v_nh_m + v_1h_1 \\ v_1h_1 + v_1h_2 & \cdots & v_1h_m + v_1h_2 & \cdots & v_nh_1 + v_1h_2 & \cdots & v_nh_m + v_1h_2 \\ \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ v_1h_1 + v_1h_m & \cdots & v_1h_m + v_1h_m & \cdots & v_nh_1 + v_1h_m & \cdots & v_nh_m + v_1h_m \\ \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ v_1h_1 + v_nh_1 & \cdots & v_1h_m + v_nh_1 & \cdots & v_nh_1 + v_nh_1 & \cdots & v_nh_m + v_nh_1 \\ \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ v_1h_1 + v_nh_m & \cdots & v_1h_m + v_nh_m & \cdots & v_nh_1 + v_nh_m & \cdots & v_nh_m + v_nh_m \end{bmatrix} \tag{13}$$

which essentially a dot product between all possible combinations $(v_i, v_j) \cdot (h_r, h_t)$ for $1 \leq i, j \leq n$ and $1 \leq r, t \leq m$. One very useful property of this product is that we can break this matrix into a

$n \times n$ block matrix of the form

$$mdot(v,h) = \begin{bmatrix} A_{1,1} & A_{2,1} & \cdots & A_{n,1} \\ A_{1,2} & A_{2,2} & \cdots & A_{n,2} \\ \vdots & \vdots & \cdots & \vdots \\ A_{1,n} & A_{2,n} & \cdots & A_{n,n} \end{bmatrix} \tag{14}$$

where

$$A_{i,j} = \begin{bmatrix} (v_i,v_j)\cdot(h_1,h_1) & (v_i,v_j)\cdot(h_2,h_1) & \cdots & (v_i,v_j)\cdot(h_m,h_1) \\ (v_i,v_j)\cdot(h_1,h_2) & (v_i,v_j)\cdot(h_2,h_2) & \cdots & (v_i,v_j)\cdot(h_m,h_2) \\ \vdots & \vdots & \cdots & \vdots \\ (v_i,v_j)\cdot(h_m,h_1) & (v_i,v_j)\cdot(h_m,h_2) & \cdots & (v_i,v_j)\cdot(h_m,h_m) \end{bmatrix}. \tag{15}$$

Note that this operation is non-commutative, and the ordering we do the product in the next section is very important. Furthermore, since all of these operations are operations that can be done in parallel in the computer, hence mdot is able to be computed in the GPU.

With our new operation mdot at our disposal, we can continue our computation of our Chern number on the torus. Our goal was to compute the values of F, G,and the partial derivatives on the whole torus. The first essential part of this, is to compute the matrices mdot$(P \otimes K_F)$ for F, mdot$(P \otimes K_G)$ for G, and similar products for the partial derivatives. We will focus our attention on the values of F on the torus, as it is similar for G and the partial derivatives.

By definition mdot$(P \otimes K_F)$ is a $N \times N$ block matrix $(A_{i,j})_{i,j}$ described in the definition of mdot above. Apply the exponential function component wise to get matrix

$$\exp\left(2\pi i \text{mdot}(P \otimes K_F)\right) = \begin{bmatrix} \exp 2\pi i A_{1,1} & \exp 2\pi i A_{2,1} & \cdots & \exp 2\pi i A_{n,1} \\ \exp 2\pi i A_{1,2} & \exp 2\pi i A_{2,2} & \cdots & \exp 2\pi i A_{n,2} \\ \vdots & \vdots & \cdots & \vdots \\ \exp 2\pi i A_{1,n} & \exp 2\pi i A_{2,n} & \cdots & \exp 2\pi i A_{n,n} \end{bmatrix}. \tag{16}$$

Here, the matrix $\exp 2\pi i A_{i,j}$ represents all possible values $e^{2\pi i(k\cdot p)}$ for a fixed point $p = (\frac{i-1}{N}, \frac{j-1}{N})$ and all possible $k \in \mathbf{Z}^2$ with respect to $N_+(F)$ and $N_-(F)$. Applying a discrete convolution with weight A_F, input $\exp\left(2\pi i \text{mdot}(P, K_F)\right)$, and stride N results in a $N \times N$ matrix

$$\begin{bmatrix} F(0,0) & F(\frac{1}{N},0) & \cdots & F(\frac{N-1}{N},0) \\ F(0,\frac{1}{N}) & F(\frac{1}{N},\frac{1}{N}) & \cdots & F(\frac{N-1}{N},\frac{1}{N}) \\ \vdots & \vdots & \cdots & \vdots \\ F(0,\frac{N-1}{N}) & F(\frac{1}{N},\frac{N-1}{N}) & \cdots & F(\frac{N-1}{N},\frac{N-1}{N}) \end{bmatrix} \tag{17}$$

which are the values of F on each of the sampled points of our partition.

With this and applying the same idea for G and the partial derivatives, we get our values of F,G, and the partial derivatives on the sampled points of the torus. We can use this to compute the integrands in the GPU and take the sum at the end to obtain an approximation to the integral.