arXiv:2506.08618v1 [cs.LG] 10 Jun 2025

HSG-12M: A Large-Scale Spatial Multigraph Dataset

Xianquan Yan[1,2] **Hakan Akgün**[3] **Kenji Kawaguchi**[2] **N. Duane Loh**[1,4,5*] **Ching Hua Lee**[1†]
Department of {[1]Physics, [2]Computer Science, [4]Biological Sciences}, National University of Singapore
[3]Department of Physics, Bilkent University, Turkey
[5]NUS Centre for Bioimaging Sciences, National University of Singapore

Abstract

Existing graph benchmarks assume non-spatial, simple edges, collapsing physically distinct paths into a single link. We introduce HSG-12M, the first large-scale dataset of **spatial multigraphs**—graphs embedded in a metric space where multiple geometrically distinct trajectories between two nodes are retained as separate edges. HSG-12M contains 11.6 million static and 5.1 million dynamic *Hamiltonian spectral graphs* across 1401 characteristic-polynomial classes, derived from 177 TB of spectral potential data. Each graph encodes the full geometry of a 1-D crystal's energy spectrum on the complex plane, producing diverse, physics-grounded topologies that transcend conventional node-coordinate datasets. To enable future extensions, we release `Poly2Graph`[3]: a high-performance, open-source pipeline that maps arbitrary 1-D crystal Hamiltonians to spectral graphs. Benchmarks with popular GNNs expose new challenges in learning from multi-edge geometry at scale. Beyond its practical utility, we show that spectral graphs serve as universal topological fingerprints of polynomials, vectors, and matrices, forging a new algebra-to-graph link. HSG-12M lays the groundwork for geometry-aware graph learning and new opportunities of data-driven scientific discovery in condensed matter physics and beyond.

1 Introduction

Graph representation learning [1–5] has emerged as a powerful paradigm for modeling structured data across disciplines. While large-scale, high-quality datasets have driven significant progress in this field [6–42], a critical limitation persists: virtually all public benchmarks treat data as *simple* graphs, allowing at most one edge between any node pair. Even when source data contains multi-edges, these are typically aggregated into a single weighted edge, discarding crucial geometric information.

In contrast, many real-world networks are fundamentally spatial multigraphs, i.e. graphs embedded in a metric space, where entities may connect through multiple distinct geometrically meaningful paths [39–59]. Such **spatial graphs** or **geometric networks** [60, 61] naturally arise in urban street networks [43–46, 52–54], biological neural networks [39–42, 55], protein structures [47, 48, 50], and beyond [56–59]. When the properties of interest include both connectivity *topology* and connection *geometry*, collapsing intrinsically distinct multi-edges results in critical information loss. Despite their ubiquity, to the best our knowledge, no large-scale spatial multigraph dataset—nor any more generic multigraph dataset—has been available to benchmark graph representation learning methods.

Simultaneously, the integration of AI into scientific research is transforming how complex physical systems are understood [12, 18–33, 39–42, 62–80]. However, this transformation is often hindered by a shortage of high-quality, domain-specific datasets, particularly in physical sciences. Recent

[*] `duaneloh@nus.edu.sg`

[†] `phylch@nus.edu.sg`

[3] https://github.com/sarinstein-yan/Poly2Graph

Preprint.

breakthroughs in protein folding [76, 75], materials discovery [78, 79], and many-body physics [19, 64, 69] underscore how well-curated scientific datasets can unlock AI's full potential, enabling discoveries that would otherwise remain inaccessible.

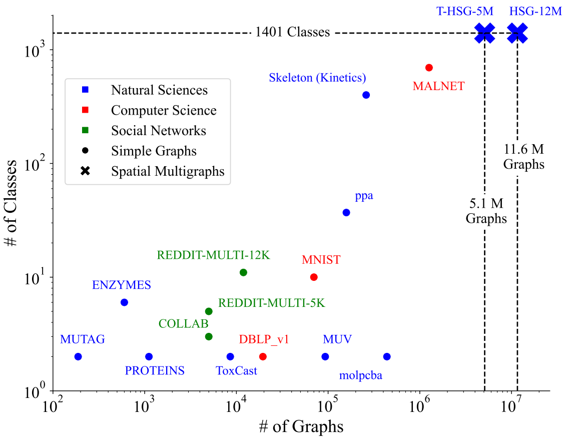

Figure 1: Number of graphs v.s. number of classes in HSG-12M compared to other graph-classification datasets. HSG-12M is the only large-scale *multigraph* (i.e. unlike *simple* graph that only allows one edge between any node pair) dataset, with exceptional class diversity even exceeds all other simple graph datasets. T-HSG-5M holds temporal spatial multigraphs. Table A3 lists comprehensive comparison.

In this work, we address these gaps with **HSG-12M** (Hamiltonian Spectral Graphs, 12 Million): the first large-scale dataset of spatial multigraphs, grounded in non-Hermitian quantum physics. Each graph is a so-called *spectral graph* [81–84] derived from the energy band structure of a one-dimensional crystal Hamiltonian, encoding the complex energy spectrum's full geometry[4]. In condensed matter physics, energy band structure is a fundamental concept, key to understanding insulators, conductors, phase transitions, electron dynamics, and system symmetries.

Recent advances have shown that the energy spectrum of one-dimensional crystals under open boundary conditions[5] forms arcs and loops on the complex energy plane. These spectral loci can be naturally represented as spatial graphs embedded in the two-dimensional $\mathbb{C}$-plane. Moreover, these *spectral graphs* serve as fingerprints with far more intricate structures than conventional topological signatures for electronic behavior (e.g., $\mathbb{Z}/\mathbb{Z}_2$ invariants, Chern number [85]). Figure 2&A4 show examples of these graphs, featuring a kaleidoscope of nontrivial edge geometries and multiplicities that form diverse patterns beyond existing graph datasets.

Despite their theoretical significance, spectral graph extraction has traditionally relied on manual plotting and visual inspection—an approach limited to toy examples and small-scale investigations. In the absence of any automated workflow or large curated dataset, its systematic studies have remained out of reach.

To overcome the reliance on manual inspection, we developed `Poly2Graph`: an open-source pipeline that combines algebraic geometry, non-Bloch band theory, and morphological image processing to fully automate spectral graph extraction. By delivering unprecedented speed and memory efficiency, `Poly2Graph` enabled us to distill 177 TB of spectral potential data into 12M spatial multigraph representations (256 GB), spanning 1401 characteristic polynomial classes. We additionally provide 5.1M *temporal spatial graphs* capturing continuous deformations of spectral graphs, establishing the first large-scale temporal (dynamic) spatial graph dataset for graph-level tasks.

In summary, this work introduces a large-scale spatial multigraph dataset and methodology at the intersection of non-Hermitian quantum physics and graph representation learning. Our key contributions include:

1. **Large Scale & Exceptional Class Diversity.** 11.6 million static and 5.1 million dynamic spatial multigraphs spanning 1401 classes, distilled from 177 TB of spectral potential data. HSG-12M is the first large-scale multigraph dataset for graph-level tasks (Figure 1&A6) with class diversity exceeding all simple graph datasets.
2. **Novel Graph Type & New Challenges.** Spatial multigraphs simultaneously capture connection topology with edge multiplicity preserved and geometry of multiedges & nodes in the embedding

[4] In mathematical terms, the energy spectrum refers to the set of eigenvalues of the Hamiltonian matrix. Within this work, energy band structure can be considered the same as energy spectrum.

[5] To be precise, it is 1-D crystal (lattice) Hamiltonian, under open boundary conditions (OBC), in the thermodynamic limit (i.e. large-size limit, the length of the lattice → ∞).

space. This first large-scale collection introduces new challenges for developing geometry-aware graph learning algorithms capable of handling edge multiplicity.

3. **New Domain, Physics-grounded, Universal Relevance.** Spectral graphs are firmly grounded in theories of non-Hermitian quantum physics, introducing an abundant database from an entirely new domain. Physically, spectral graph encapsulates information about quantum state dynamics and topology, Hamiltonian symmetry class, response strength, quantum sensing capability, and more. Thus our database paves the way for accelerating discovery of exotic phases, enabling rational design of materials with desired quantum properties.

   Additionally, we identify *Hamiltonian spectral graph* as a new class of topological object deserving attention in its own right—in section 6 we show that vectors, matrices, and polynomials, be they real or complex, admit spectral graphs as their topological fingerprint, bridging graph and ubiquitous algebra objects.

4. **Open-source, High-performance Generator.** We release Poly2Graph that can map arbitrary 1D Hamiltonians to spectral graphs, providing the first automated tool to study spectral graphs with high speed and efficiency[6]. Poly2Graph not only enables us to produce HSG-12M, but also empowers researchers to generate custom spectral graph datasets, vastly expanding the possibilities for future study.

2 Related Work

Graph Representation Learning, Datasets, and Benchmarks. Graph learning has seen a rapid rise in recent years, driven by advances in graph neural networks (GNNs) [1–5] and proliferation of datasets and benchmarks [6–14]. HSG-12M addresses critical gaps in existing benchmarks by introducing not only the first large-scale spatial multigraph dataset[7], but also one of the largest known graph machine learning datasets and natural science-based datasets. This work sets a new standard in terms of scale and class diversity.

Graph Learning in Multigraphs. In contrast to *simple* graphs, *multi*graphs permit multiple edges between the same pair of nodes. Apart from a handful of exploration on multigraph learning algorithms [87, 88], progress has been hampered by the absence of large-scale data sources.

Consequently, in many practical settings, multiple edges are typically collapsed into a single edge—often sacrificing valuable information. This simplification may be acceptable when edge-level details can be represented as aggregated attributes, as is often the case in heterogeneous graphs [89, 90], multi-modular models [41, 91], or multiplex networks [92].

However, in spatial multigraphs [60], where edges carry rich geometric information such as distances, directions, or physical observable information, such aggregation results in significant information loss. This critical issue has remained underexplored due to the lack of datasets where edge aggregation is inherently infeasible. HSG-12M addresses this gap by providing the first benchmark where capturing both multi-edge relationships and edge geometry is essential.

Graph Learning in Spatial Graphs. A spatial (or geometric) graph is a network in which nodes and edges are spatial entities living in a metric space [60, 93]. Such networks emerge naturally in domains where spatial embedding is fundamental to structure and function: urban, transportation, and communication networks are shaped by physical distances and road geometries [52, 53, 43–46, 54]; biological systems like neural and vascular networks are constrained by surrounding tissue geometry [55–57]; and river networks evolve through interactions of gravity and topography [58, 59]. In all these cases, spatial graph structure encodes essential information that cannot be inferred from connectivity alone or reconstructed from non-spatial data.

Despite growing recognition of spatial information in Spatial and Geo AI [94–98], its importance remains underappreciated in graph learning. Currently, no benchmark exists with sufficiently rich geometric structure to exhibit intricate spatial patterns, let alone one of **spatial multigraphs**. As a result, despite significant efforts to develop algorithms for spatial graphs [61, 93, 99–101], the field has lacked a standardized, large-scale testbed.

[6] It achieves $10^6\times$ speedup and 20-40× memory efficiency to compute spectral potential data—the computation bottleneck—compared to the best available code, which is acquired from Ref. [81]. The comparison only applies to the computation bottleneck, as Ref. [81] does not automate the graph extraction.

[7] To our knowledge, this is also the first *large-scale multigraph* dataset–*large-scale* conforms to OGB criteria [86].

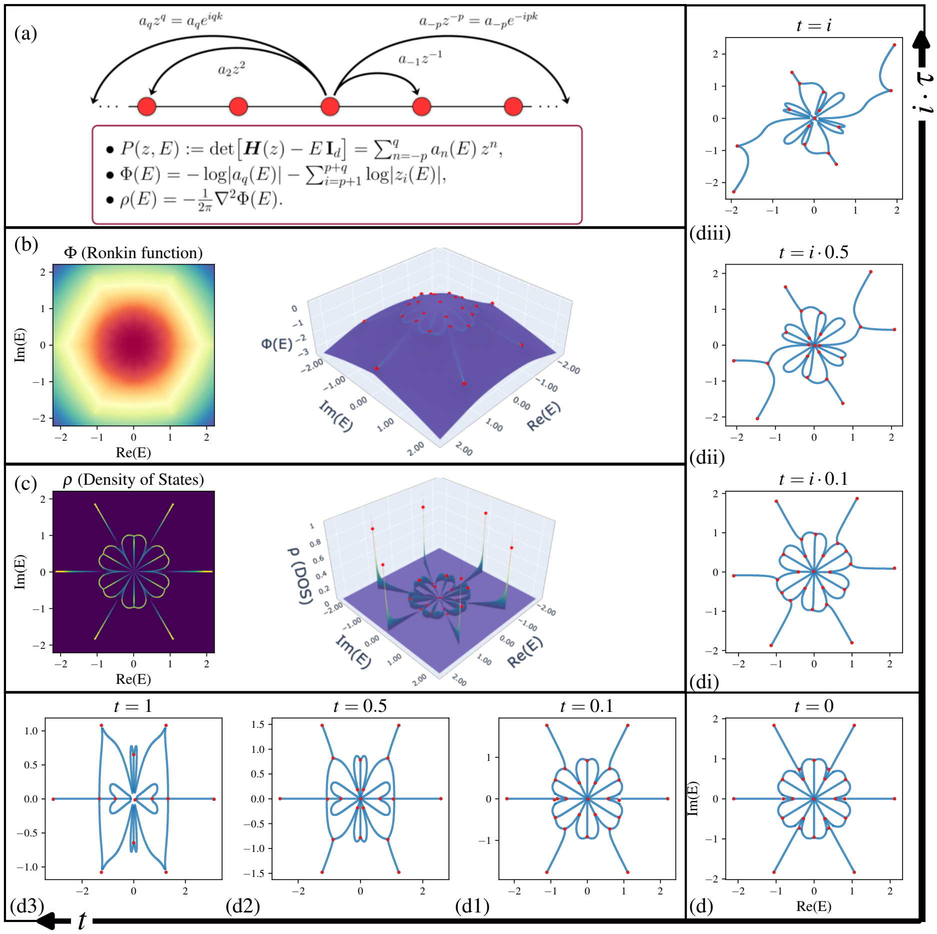

Figure 2: **Poly2Graph pipeline.** (a) Starting from a 1-D crystal Hamiltonian $H(z)$ in momentum space—or, equivalently, its *characteristic polynomial* $P(z, E) = \det[\boldsymbol{H}(z) - E\boldsymbol{I}]$. The crystal's open-boundary spectrum solely depends on $P(z, E)$. (b) The *spectral potential* $\Phi(E)$ (Ronkin function) is computed from the roots of $P(z, E) = 0$, following recent advances in non-Bloch band theory [81, 83, 84]. (c) The density of states $\rho(E)$ is obtained as the Laplacian of $\Phi(E)$. (d) The spectral graph extracted from $\rho(E)$ via a morphological computer-vision pipeline. Varying the coefficients of $P(z, E)$ produces diverse graph morphologies in the real domain (d1)-(d3) and imaginary domain (di)-(diii).

3 Poly2Graph: Automating Spectral Graph Extraction

Poly2Graph is the first *end-to-end*, high-performance pipeline that converts an arbitrary one-dimensional crystal Hamiltonian into its *spectral graph* representation. It operationalises the mathematical construction reviewed in appendix A by integrating non-Bloch band theory, algebraic geometry, and morphological image processing.

Full algorithmic details are deferred to appendix B. Here we highlight the design choices that make Poly2Graph *six orders of magnitude faster* and *20-40× more memory-efficient* than the best available code[6], thereby enabling the construction of HSG-12M.

From Hamiltonians to Characteristic Polynomials. Poly2Graph initializes with either a Bloch Hamiltonian matrix $H(z)$ or its characteristic polynomial. For a s-band tight-binding crystal chain, the Bloch Hamiltonian reads

$$\boldsymbol{H}(z) = \sum_{j=-p}^{q} \boldsymbol{T}_j\, z^j, \qquad z = e^{ik},\ k \in [-\pi, \pi),\ \boldsymbol{T}_j \in \mathbb{C}^{s \times s}, \tag{1}$$

Its open-boundary spectrum solely depends on the roots of the Laurent *characteristic polynomial*:

$$P(z,E) := \det\left[\boldsymbol{H}(z) - E\,\mathbf{I}_s\right] = \sum_{n=-p}^{q} a_n(E)\,z^n. \tag{2}$$

We choose an energy window $\Omega \subset \mathbb{C}$ in the complex energy plane that encloses the entire spectral graph $\mathcal{G}$. By default, Poly2Graph estimates Ω by diagonalising a small real-space Hamiltonian with $L = 40$ unit cells, though users may optionally specify a custom region and resolution. The resultant region Ω is discretized into a grid of complex energy values. In HSG-12M, we used a default resolution of 256 (initial) × 4 (adaptive enhancement) = 1024 points along each axis.

For each sample energy $E \in \Omega$, we solve the roots $\{z_i(E)\}$ of $P(z,E) = 0$ (treating E as constant) and then sort them by magnitude $|z_1(E)| \leq |z_2(E)| \leq \cdots \leq |z_{p+q}(E)|$. This is the computational bottleneck in naive approaches—solving roots of a large batch of high-degree polynomial for every grid point is extremely expensive. To tame this bottleneck, we implement a custom, optimized root-solver based on Frobenius companion matrices and parallel eigen-solvers with auto-backend detection for optional GPU acceleration, cutting wall-time from days to seconds.

Spectral Potential & Density-of-States (as 2D Images). With the roots $\{z_i(E)\}$ computed, we leverage non-Bloch band theory [81, 83, 84] and reliably compute the *spectral potential*[8] as:

$$\Phi(E) = -\log|a_q(E)| - \sum_{i=p+1}^{p+q} \log\big|z_i(E)\big|\,, \tag{3}$$

where $a_q(E)$ is the leading coefficient of the characteristic polynomial. The Laplacian of this potential yields the *Density of States* (DOS):

$$\rho(E) = -\frac{1}{2\pi}\nabla^2\Phi(E). \tag{4}$$

where $\nabla^2 = \partial^2_{\mathrm{Re}E} + \partial^2_{\mathrm{Im}E}$. Physically, $\rho(E)$ quantifies the number of eigenstates per unit area at energy E in the complex plane; hence, the spectral graph manifests where $\rho(E) > 0$ (Figure 2c). Geometrically, since DOS is defined as the second derivative, i.e. curvature, the spectral graph corresponds to the ridges of the spectral potential landscape (Figure 2b).

In addition, we exploit inherent symmetries in special polynomials. For example, the complex conjugate root theorem guarantees that if $P(z,E)$ has purely real coefficients, its spectral graph is symmetric about the real axis; similarly, purely imaginary coefficients produce symmetry about the imaginary axis. By calculating only the relevant half-plane and mirroring the results, we reduce computation time by up to 50% for qualifying polynomials.

Image-to-Graph Routine. To extract the spectral graph from the DOS image, we binarize the DOS and apply skeletonization to obtain a one-pixel-wide graph skeleton.

However, we face a resolution-computation tradeoff: insufficient resolution results in lost topological features (small loops, adjacent nodes, etc), while uniform high-resolution calculation across the entire energy window Ω is prohibitively expensive, especially since the spectral graph typically occupies only a small fraction of this area.

We resolve this challenge with a two-stage adaptive resolution approach:

1. *Coarse identification*: We first compute the DOS on a moderately-resolved grid (256×256), threshold to binarize the image, and perform morphological dilation with a 2×2 disk. This generates a conservative binary mask that envelops the spectral graph while excluding approximately 95-99% of non-contributive regions.
2. *Refined calculation*: Within only the masked region, we subdivide each pixel into an $m \times m$ grid (default $m = 4$), recalculating the spectral potential and DOS at this higher resolution. This targeted approach achieves an effective resolution of 1024×1024 while computing just 1-5% of the grid points.

The high-resolution DOS is then re-binarized and subjected to iterative morphological thinning operations [102] until a one-pixel-wide skeleton remains, preserving topological features ready to be distilled into a graph representation.

[8] The spectral potential is also known as the Ronkin function, an algebro-geometric property of $P(z,E)$ [84]

For the final graph extraction, we analyze this skeleton to identify three point types: (1) junction nodes where three or more paths intersect, (2) leaf nodes where paths terminate, and (3) edge points along continuous segments. The output is an `NetworkX MultiGraph` object. Crucially, each edge stores its complete geometric information as an ordered sequence of $(\mathrm{Re}(E), \mathrm{Im}(E))$ coordinates, preserving not just connectivity but the exact shape of each spectral curve.

Quality Assurance and Limitations. We validated Poly2Graph on hundreds of characteristic polynomials, by visually checking that the spectral graph from Poly2Graph agrees with the energy spectrum from exact diagonalization. In rare complicated cases, numerical instabilities can still arise close to the junction nodes whose surrounding edges have extremely low DOS (see appendix B.5). Poly2Graph will attempt to mitigate such cases by merging nearby nodes and contracting edges shorter than a predefined tolerance.

Open-Source Release and Broader Impact. Poly2Graph is released under the MIT licence at `https://github.com/sarinstein-yan/Poly2Graph` and can be installed via `$ pip install poly2graph`. We attach a tutorial in appendix F. Poly2Graph establishes a turn-key mechanism for translating linear operators into machine-learning-ready graphs, bridging condensed matter physics and graph representation learning. The same principle extends to any vector, matrix, and univariate/bivariate polynomial, opening an "algebra-as-graph" perspective (section 6, appendix E).

4 HSG-12M Dataset Description

The speed and memory efficiency of Poly2Graph make large-scale spatial multigraph research practical for the first time. Figure 1&A6 illustrate the scale of HSG-12M, showing #graphs vs. #classes and #graphs vs. total #nodes relative to other graph classification datasets. To our knowledge, HSG-12M is not only the largest dataset by number of graphs and classes but also the only large-scale spatial multigraph dataset available; moreover, each graph class corresponds a particular physical model in condensed matter physics.

In Table C we provide a comprehensive comparison with existing graph datasets and benchmarks. Most prior popular graph-classification datasets are non-spatial, simple graphs. A few are spatial, e.g., some superpixels and molecular graphs have node coordinates in 2D / 3D, but their edges remain an abstract connection defined by adjacency. HSG-12M uniquely provides *spatial multigraphs*, where the intricate geometric structure of multi-edges carries essential information that cannot be simplified without loss. The most relevant resource, OpenStreetMap [44] is much smaller, less diverse, and lacks associated ML tasks in comparison.

Furthermore, while temporal graph datasets exist [103], they typically focus on node/edge-level tasks or involve small numbers of graphs and classes. Our T-HSG-5M represents the first large-scale collection of dynamic spatial graphs, capturing the continuous evolution of spectral graphs over Hamiltonian parameters.

Data Format and Accessibility. To maximize accessibility and flexibility, we release HSG-12M under a permissive **CC BY 4.0** license. The dataset is publicly available via `Dataverse` [104]. Users can download the full dataset or select specific subsets using the code provided at `https://github.com/sarinstein-yan/HSG-12M`.

The dataset comprises 1401 separate Python `npz` files, each containing graphs from one class with relevant metadata. Raw files use `NetworkX MultiGraph` format, preserving full node and edge geometry:

Node attributes: complex coordinates, spectral potential, and density of states.

Edge attributes: edge length (also serving as weight), coordinate sequences along the edge, average spectral potential and average DOS over the edge.

We provide this descriptive format because representation learning on spatial multigraphs remains nascent, with no agreed-upon standard for representing continuous edge geometry. Rather than imposing a particular featurization, we encourage researchers to explore various approaches, e.g., treating edge curves as sequences, computing summary features like curvature, or developing novel and more sophisticated neural network-based representations. Moreover, the attribute-rich format here aids interpretability and is relevant to researchers interested in the underlying physics.

That said, for convenience, we propose our own featurization scheme and include a conversion script that transforms raw data into PyTorch Geometric (PyG) datasets with stratified train/validation/test splits (8:1:1) for graph classification benchmarking. Particularly, to manage the inhomogeneity of edge coordinates and make the spectral graphs compatible with standard GNN input, our reference conversion samples fixed numbers of equidistant points along each edge.

Table 1: Key statistics of the four HSG benchmark datasets. #Graphs: number of graphs; #Classes: number of classes; Ratio: the #Graphs of the largest class / #Graphs of the smallest class; Temporal: whether the graphs are temporal. All other five datasets are derived from HSG-12M; thus all datasets are **spatial** and irreducibly **multigraph**. `HSG-topology` contains non-isomorphic graphs in each class and is the only *imbalanced* dataset; `T-HSG-5M` is the *temporal* spectral graph collection; the rest four teal-colored datasets are balanced, static datasets.

Name	#Graphs	#Classes	Ratio	Temporal
`HSG-one-band`	198,744	24	1.0	-
`HSG-two-band`	2,277,275	275	1.0	-
`HSG-three-band`	9,125,662	1102	1.0	-
`HSG-topology`	1,812,325	1401	660.2	-
`T-HSG-5M`	5,099,640	1401	1.0	✓
`HSG-12M`	11,601,681	1401	1.0	-

Dataset Construction. Graphs are grouped by different Hamiltonian families (i.e. characteristic polynomial classes) as detailed in appendix A. We systematically sample polynomial classes while respecting mathematical symmetries to avoid spurious abundance. For instance, if a polynomial exhibits z-reciprocity—i.e. $P(z) = z^{p+q}P(1/z)$—this reciprocal transformation physically means flipping the crystal chain from left to right, which leaves the spectrum unchanged and yields the same spectral graph.

Specifically, we start from a base polynomial with a fixed hopping range $p + q$ and number of bands s (i.e. s-band Hamiltonian):

$$\hat{P}(z, E) = -E^s + z^{-p} + z^q \; . \tag{5}$$

We then set the degree of $E^k : k \in \{0, 1, \dots, s-1\}$ for each $z^i : i \in \{-p+1, \dots, q-1\}$. Subsequently, we assign two free coefficients (a, b) to two chosen monomials $z^j : j \in \{-p+1, \dots, -1, 1, \dots, q-1\}$—excluding z^0, since varying the constant term only raise or lower the entire spectral potential landscape, no effect exerted on the spectral graph.

For example, a two-band polynomial with $p = 3$ and $q = 3$ may take the form:

$$\hat{P}(z, E) = -E^2 + z^{-3} + \left(a\, z^{-1} + b\, E\, z + E\, z^2\right) + z^3, \quad a, b \in \mathbb{C} \; . \tag{6}$$

Under such a sampling scheme, we iterate over all combinations for one-band to three-band polynomials, with hopping ranges varied from four to six. After removing duplicates, we collect 24 one-band classes, 275 two-band classes, and 1102 three-band classes, amounting to a total of 1401 unique classes.

Finally, we vary the two free coefficients from $-10 - 5i$ to $10 + 5i$ respectively, with 13 real and 7 imaginary values, yielding $(13 \times 7)^2 = 8281$ samples per class.

Dataset Variants. We provide six datasets tailored to different research needs.

`HSG-one-band`: Small-to-medium scale, the collection of all one-band polynomials, balanced subset with 198,744 graphs across 24 classes. These graphs in this subset display simpler patterns ideal for rapid prototyping and algorithm validation.

`HSG-two-band` and `HSG-three-band`: Medium-to-large scale, the collection of all two-band and three-band polynomials respectively, balanced datasets with increasing complexity, containing 2.3M and 9.1M graphs across 275 and 1,102 classes, respectively.

`HSG-12M`: The complete dataset spanning all 1,401 classes with balanced sampling, totaling 11.6M static graphs, designed for large-scale challenge.

`HSG-topology`: An imbalanced subset preserving only *topologically* distinct (i.e. non-isomorphic) graphs within each class. This filtered dataset removes isomorphic duplicates, resulting in highly skewed class distributions (max class size ratio 660.2), useful for analyzing spectral graph topology diversity and benchmarking graph algorithms on imbalanced datasets.

Table 2: Graph-level classification results on the three HSG dataset variants. Test metrics shown as mean$_{\pm \text{std}}$ over three random seeds; best result in **Bold**. This minimal benchmarking indicates substantial headroom for improvement on complicated spatial multigraphs with high class diversity and imbalance.

Dataset	Metric	Model			
		GCN	GIN	GAT	GraphSAGE
HSG-one-band	Top-1 Acc.	$.532_{\pm .006}$	$.249_{\pm .060}$	$.301_{\pm .061}$	$\mathbf{.694}_{\pm .010}$
	Top-10 Acc.	$.995_{\pm .001}$	$.939_{\pm .026}$	$.958_{\pm .028}$	$\mathbf{.999}_{\pm .001}$
	Macro F_1	$.512_{\pm .001}$	$.183_{\pm .058}$	$.270_{\pm .069}$	$\mathbf{.679}_{\pm .015}$
HSG-two-band	Top-1 Acc.	$.445_{\pm .010}$	$.344_{\pm .012}$	$.076_{\pm .005}$	$\mathbf{.672}_{\pm .011}$
	Top-10 Acc.	$.922_{\pm .004}$	$.869_{\pm .007}$	$.420_{\pm .020}$	$\mathbf{.989}_{\pm .002}$
	Macro F_1	$.429_{\pm .009}$	$.322_{\pm .011}$	$.057_{\pm .003}$	$\mathbf{.663}_{\pm .010}$
HSG-topology	Top-1 Acc.	$.137_{\pm .014}$	$.002_{\pm .001}$	$.015_{\pm .003}$	$\mathbf{.465}_{\pm .013}$
	Top-10 Acc.	$.513_{\pm .019}$	$.018_{\pm .005}$	$.109_{\pm .010}$	$\mathbf{.893}_{\pm .008}$
	Macro F_1	$.077_{\pm .008}$	$.000_{\pm .000}$	$.003_{\pm .001}$	$\mathbf{.366}_{\pm .012}$

`T-HSG-5M`: Our temporal multigraph collection capturing continuous spectral graph evolution. As shown in figure 2d, varying either the real or imaginary part of a coefficient in the characteristic polynomial continuously morphs the *geometry* of the spectral graph; at certain transition points, one can observe the graph *topology* changes discontinuously. For each class, we collect all sequences of the variation in real (or imaginary) parts of one free coefficient, adding up to 5.1M temporal graphs across 1401 classes. T-HSG-5M is suitable for evaluating temporal graph-level tasks such as temporal extrapolation and classification on early sequences. Functionality to select any desired sequence or subset is provided in the same dataset repository.

5 Benchmarking Results

To assess the capabilities of existing graph learning methods on the new challenges introduced by our HSG datasets, particularly their spatial nature, edge multiplicities, class imbalance, and scale, we benchmark popular GNNs on the `HSG-one-band`, `HSG-two-band`, `HSG-topology`[9] and discuss the implications of these initial baselines.

Baseline Models. We trained four popular graph neural networks (GNNs)—GCN [105], GIN [106], GAT [107], and GraphSAGE [108]. All models have four message-passing layers, with hidden sizes ranging from 128 / 256 / 512 for three subsets respectively. The readout of node-level convolutions are aggregated by global sum pooling. The resulting graph-level embeddings are then passed through a two-layer MLP to produce class logits. A dropout layer with probability 0.1 follows every learnable layer. See appendix D for data preprocessing and all hyperparameter specifics.

Experiment Setup. All experiments were performed on an NVIDIA A5000 GPU. For each dataset, we generated three stratified splits (80% train, 10% validation, 10% test) using different random seeds. Models were trained by AdamW [109] (AMSGrad variant) optimizer, with no weight decay. We employ a cosine annealing learning rate scheduler with warm restarts [110]. In table 2, we report test performance averaged over the three seeds.

Evaluation Metrics. Given the high class diversity, we report Top-1 and Top-10 accuracy—relevant for scenarios where multiple plausible answers are acceptable. We also report the Macro-averaged F_1 score which weights every class equally and exposes performance on minority classes. Peak GPU memory utilization, training throughput are reported in appendix D.

Results and Analysis. The graph-level classification results are presented in Table 2. Several observations emerge from these results:

GraphSAGE excels with limited budgets. Across all datasets GraphSAGE learns the fastest and achieves the best Top-1 and Macro-F_1 scores, confirming its strength on large-scale datasets with a fixed training budget. With substantially longer training schedules, the ranking may change, as the other models are more expressive to accommodate the complexity of the dataset.

[9] We only benchmark on the three small-to-medium scale datasets. The `HSG-three-band`, `HSG-12M`, `T-HSG-5M` are out of our reach due to limited resources.

The imbalanced `HSG-topology` split is challenging. Removing isomorphic duplicates exposes severe class imbalance (max class size ratio ≈ 660) and deprives the model of continuously varying spatial information, causing a dramatic drop in performance.

Performance degrades with task difficulty. Test metrics drop from the `one-band` subset (balanced 24 classes) to the `topology` subset (imbalanced 1401 classes), as expected. Nevertheless, most of the time Top-10 accuracy is good, indicating that models still recover a short list of plausible classes.

Take-away. Standard off-the-shelf GNNs cope reasonably well with the simplest subset but leave substantial room on improvement on complicated spatial multigraphs with high class diversity and imbalance. The presented numbers therefore serve as *minimal* baselines; we invite the community to design geometry-aware encoders, multi-edge message passing schemes, and temporal models to close the substantial accuracy gap.

6 Discussion

Universal Relevance of Spectral Graphs. While HSG-12M is rooted in non-Hermitian band theory, its reach extends well beyond condensed-matter physics.

1. Any **bivariate Laurent polynomial** $P(z, E)$ has a spectral graph.
2. Any **univariate polynomial** $h(z)$ can be viewed as a one-band Bloch Hamiltonian; and any **vector** can be treated as a symmetrised coefficient list of a univariate polynomial.
3. Any **matrix** can be decomposed into a product of one-band Hamiltonian matrices [111], and thus in general has a *multiset* of spectral graphs (detailed in appendix E)

Hence polynomials, vectors, and matrices all admit spectral graphs as their topological fingerprints. This establishes a universal bridge between algebraic objects and graphs, inviting graph-based methods to problems in linear algebra.

Benchmarking and algorithmic opportunities. HSG-12M fills three key gaps at once: (i) it is the first *large-scale multigraph* dataset, (ii) it is the first *spatial multigraph* resource, retaining edge multiplicity with rich continuous geometry, and (iii) it provides both static and *dynamic* multigraph sequences. These traits open a suite of tasks that are under-served by current methods: multi-edge featurization, geometry-aware message passing, spatio-temporal prediction, imbalance-robust learning, etc. Beyond supervised learning, the dataset is large enough and expandable with Poly2Graph to support topology-conditioned generation and pre-training foundation models for rational inverse-design of materials.

Limitations and future work. Our extraction pipeline struggles when the hopping range or band number becomes large, because extremely low densities of states make the graph skeleton fragile, occasionally fragmenting a connected component (as shown in the bottom row in figure A4). We term this phenomenon *component fragmentation* and note that it is an intrinsic limitation of the spectral graph per se (see appendix B.5). The focus of this work is on the dataset and its generator; our benchmark serves merely as a minimal baseline due to limited resources. We invite the community to perform comprehensive, large-scale, and carefully designed benchmarking to unleash the full potential of the presented resource.

7 Conclusion

We present **HSG-12M**—11.6M static and 5.1M dynamic **spatial multigraphs** drawn from the energy spectrum of one-dimensional crystal under open boundary condition—and `Poly2Graph`, an open-source pipeline that makes their extraction faster and lighter than prior methods[6]. HSG-12M collects physics-grounded data, offering the first large-scale benchmark with irreducible edge multiplicity and geometry. The construction generalizes to arbitrary polynomials, vectors, and matrices. Benchmarking results indicate that popular GNNs struggle with geometry-aware message passing, edge multiplicity, leaving substantial room for methodological advances. We release `Poly2Graph` and HSG-12M under permissive licences and invite the community to build on this resource for new models, tasks, and insights across machine learning and condensed-matter physics.

Acknowledgments and Disclosure of Funding

This research is partially supported by the National Research Foundation Singapore under the AI Singapore Programme (AISG Award No: AISG2-TC-2023-010-SGIL) and the Singapore Ministry of Education Academic Research Fund Tier 1 (Award No: T1 251RES2207).

References

[1] William L. Hamilton. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159, 2020.

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

[3] Yao Ma and Jiliang Tang. *Deep Learning on Graphs*. Cambridge University Press, 2021.

[4] Lingfei Wu, Peng Cui, Jian Pei, Liang Zhao, and Le Song. Graph neural networks. In *Graph Neural Networks: Foundations, Frontiers, and Applications*, pages 27–37. Springer, 2022.

[5] Gabriele Corso, Hannes Stark, Stefanie Jegelka, Tommi Jaakkola, and Regina Barzilay. Graph neural networks. *Nature Reviews Methods Primers*, 4:17, 2024.

[6] Smita Ranveer and Swapnaja Hiray. Comparative analysis of feature extraction methods of malware detection. *International Journal of Computer Applications*, 120:1–7, June 2015.

[7] Scott Freitas, Yuxiao Dong, Joshua Neil, and Duen Horng Chau. A large-scale database for graph representation learning, 2021.

[8] Cai Chen and Yusu Wang. A simple yet effective baseline for non-attributed graph classification. In *International Conference on Learning Representations (ICLR)*, 2019.

[9] Federico Errica, Marco Podda, Davide Bacciu, and Alessio Micheli. A fair comparison of graph neural networks for graph classification. *arXiv preprint arXiv:1912.09893*, 2019.

[10] Till Schulz and Pascal Welke. On the necessity of graph kernel baselines. In *ECML-PKDD GEM Workshop*, 2019.

[11] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

[12] Karsten M. Borgwardt, Cheng Soon Ong, Stefan Schönauer, S.V.N. Vishwanathan, Alexander J. Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics (Oxford, England)*, 21(Suppl 1):i47–i56, 2005.

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

[14] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

[15] Shurui Gui, Xiner Li, Limei Wang, and Shuiwang Ji. GOOD: A graph out-of-distribution benchmark. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022) Datasets and Benchmarks Track*, 2022.

[16] Vijay Dwivedi, Francesco M Bianchi, Feng Yan, and et al. Long range graph benchmark. *arXiv preprint arXiv:2007.02839*, 2020.

[17] Scott Freitas, Yuxiao Dong, Joshua Neil, and Duen Horng Chau. A large-scale database for graph representation learning, 2021.

[18] Anna Varbella, Kenza Amara, Blazhe Gjorgiev, and Giovanni Sansavini. PowerGraph: A power grid benchmark dataset for graph neural networks, 2024.

[19] Sheng Yang, Fengge Wu, and Junsuo Zhao. MBDS: A multi-body dynamics simulation dataset for graph networks simulators. October 2024.

[20] L. Ruddigkeit, R. van Deursen, L. C. Blum, and J.-L. Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *Journal of Chemical Information and Modeling*, 52:2864–2875, 2012.

[21] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1(1):1–7, 2014.

[22] J. J. Irwin, T. Sterling, M. M. Mysinger, E. S. Bolstad, and R. G. Coleman. ZINC: A free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling*, 52(7):1757–1768, 2012.

[23] W. Jin, K. Yang, R. Barzilay, and T. Jaakkola. Learning multimodal graph-to-graph translation for molecular optimization, 2018.

[24] Daniil Polykovskiy et al. Molecular sets (MOSES): A benchmarking platform for molecular generation models. *Frontiers in Pharmacology*, 11, 2020.

[25] David Mendez et al. ChEMBL: Towards direct deposition of bioassay data. *Nucleic Acids Research*, 47(D1):D930–D940, 2019.

[26] X. Guo, L. Zhao, C. Nowzari, S. Rafatirad, H. Homayoun, and S. M. Dinakarrao. Deep multi-attributed graph translation with node-edge co-evolution. In *Proceedings of the 19th International Conference on Data Mining (ICDM)*, 2019.

[27] Justin Smith, Roman Zubatyuk, Benjamin Nebgen, Nicholas Lubbers, Kipton Barros, Adrian Roitberg, Olexandr Isayev, and Sergei Tretiak. The ANI-1ccx and ANI-1x data sets, coupled-cluster and density functional theory properties for molecules. *Scientific Data*, 7:134, May 2020.

[28] Bharath Ramsundar, Steven Kearnes, Patrick Riley, Dale Webster, David Konerding, and Vijay Pande. Massively multitask networks for drug discovery. *arXiv preprint arXiv:1502.02072*, 2015.

[29] Sebastian G Rohrer and Knut Baumann. Maximum unbiased validation (MUV) data sets for virtual screening based on PubChem bioactivity data. *Journal of Chemical Information and Modeling*, 49(2):169–184, 2009.

[30] Xifeng Yan, Hong Cheng, Jiawei Han, and Philip S. Yu. Mining significant graph patterns by leap search. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 433–444, 2008.

[31] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: A benchmark for molecular machine learning. *Chemical Science*, 9(2):513–530, 2018.

[32] Nikil Wale, Ian A Watson, and George Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*, 14(3):347–375, 2008.

[33] Hannu Toivonen, Ashwin Srinivasan, Ross D King, Stefan Kramer, and Christoph Helma. Statistical evaluation of the predictive toxicology challenge 2000–2001. *Bioinformatics (Oxford, England)*, 19(10):1183–1193, 2003.

[34] Kaspar Riesen and Horst Bunke. IAM graph database repository for graph based pattern recognition and machine learning. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 287–297. Springer, 2008.

[35] Marion Neumann, Plinio Moreno, Laura Antanas, Roman Garnett, and Kristian Kersting. Graph kernels for object category prediction in task-dependent robot grasping. In *Online Proceedings of the Eleventh Workshop on Mining and Learning with Graphs*, pages 0–6, 2013.

[36] Paul D Dobson and Andrew J Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of Molecular Biology*, 330(4):771–783, 2003.

[37] Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar. An API oriented open-source python framework for unsupervised learning on graphs. In *Proceedings of CIKM*, 2020.

[38] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 177–187, 2005.

[39] M. W. Weiner, P. S. Aisen, Jr. Jack, C. R., W. J. Jagust, J. Q. Trojanowski, L. Shaw, et al. The Alzheimer's disease neuroimaging initiative: Progress report and future plans. *Alzheimer's & Dementia*, 6(3):202–211, 2010.

[40] A. Di Martino, C.-G. Yan, Q. Li, E. Denio, F. X. Castellanos, K. Alaerts, et al. The autism brain imaging data exchange: Towards a large-scale evaluation of the intrinsic brain architecture in autism. *Molecular Psychiatry*, 19(6):659–667, 2014.

[41] Anwar Said, Roza G. Bayrak, Tyler Derr, Mudassir Shabbir, Daniel Moyer, Catie Chang, and Xenofon Koutsoukos. NeuroGraph: Benchmarks for graph machine learning in brain connectomics, 2024.

[42] Sara Larivière, Casey Paquola, Bo-yong Park, Jessica Royer, Yezhou Wang, Oualid Benkarim, Reinder Vos de Wael, Sofie L Valk, Sophia I Thomopoulos, Matthias Kirschner, Lindsay B Lewis, Alan C Evans, Sanjay M Sisodiya, Carrie R McDonald, Paul M Thompson, and Boris C Bernhardt. The ENIGMA Toolbox: Multiscale neural contextualization of multisite neuroimaging datasets. *Nature Methods*, 18(7):698–700, 2021.

[43] Rainer Kujala, Christoffer Weckström, Richard Darst, Milos Mladenovic, and Jari Saramäki. A collection of public transport network data sets for 25 cities. *Scientific Data*, 5:180089, May 2018.

[44] Geoff Boeing. Street network models and measures for every u.s. city, county, urbanized area, census tract, and zillow-defined neighborhood. *Urban Science*, 3(28), 2019.

[45] Manlio De Domenico, Albert Solé-Ribalta, Sergio Gómez, and Alex Arenas. Navigability of interconnected networks under random failures. *Proceedings of the National Academy of Sciences*, 111(23):8351–8356, 2014.

[46] Yaoli Wang, Di Zhu, Ganmin Yin, Zhou Huang, and Yu Liu. A unified spatial multigraph analysis for public transport performance. *Scientific Reports*, 10:9573, 2020.

[47] Namrata Anand and Po-Ssu Huang. Generative modeling for protein structures. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7505–7516, 2018.

[48] Xiaojie Guo, Sivani Tadepalli, Liang Zhao, and Amarda Shehu. Generating tertiary protein structures via an interpretative variational autoencoder. *arXiv preprint*, arXiv:2004.07119, 2020.

[49] Farras Abdelnour, Michael Dayan, and Orrin Devinsky. Functional brain connectivity is predictable from anatomic network's laplacian eigenstructure. *NeuroImage*, 172:728–739, 2018.

[50] Taseef Rahman, Yuanqi Du, Liang Zhao, and Amarda Shehu. Generative adversarial learning of protein tertiary structures. *Molecules*, 26(5):1209, 2021.

[51] Dou Huang, Xuan Song, and Zipei Fan. A variational autoencoder based generative model of urban human mobility. In *Proceedings of the IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 425–430. IEEE, 2019.

[52] Jérôme Buhl, Jacques Gautrais, Nicholas Reeves, Ricard V. Solé, Sergi Valverde, Pascal Kuntz, and Guy Theraulaz. Topological patterns in street networks of self-organized urban settlements. *The European Physical Journal B*, 49:513–522, 2006.

[53] Alessio Cardillo, Salvatore Scellato, Vito Latora, and Sergio Porta. Structural properties of planar graphs of urban street patterns. *Physical Review E*, 73(6):066107, 2006.

[54] Dou Huang, Xuan Song, and Zipei Fan. A variational autoencoder based generative model of urban human mobility. In *Proceedings of the IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 425–430. IEEE, 2019.

[55] Geoffrey B. West and James H. Brown. The origin of allometric scaling laws in biology from genomes to ecosystems: Towards a quantitative unifying theory of biological structure and organization. *Journal of Experimental Biology*, 208:1575–1592, 2003.

[56] Adrian Runions, Anne M. Fuhrer, Peter Federl, Brendan Lane, Anne-Gaëlle Rolland-Lagan, and Przemyslaw Prusinkiewicz. Modeling and visualization of leaf venation patterns. *ACM Transactions on Graphics (TOG)*, 24(3):702–711, 2005.

[57] Edward Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10:186–198, 2009.

[58] Guido Caldarelli. *Scale-Free Networks*. Oxford University Press, Oxford, 2007.

[59] Ignacio Rodriguez-Iturbe and Andrea Rinaldo. *Fractal River Basins: Chance and Self-Organization*. Cambridge University Press, Cambridge, 1997.

[60] Marc Barthélemy. Spatial networks. *Physics Reports*, 499(1-3):1–101, 2011.

[61] Xiaojie Guo, Yuanqi Du, and Liang Zhao. Deep generative models for spatial networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '21, pages 198–208, Virtual Event, Singapore, August 14–18 2021. ACM.

[62] Jing Wei, Xuan Chu, Xiang-Yu Sun, Kun Xu, Hui-Xiong Deng, Jigen Chen, Zhongming Wei, and Ming Lei. Machine learning in materials science. *InfoMat*, 1(3):338–358, 2019.

[63] Karianne Bergen, Paul Johnson, Maarten de Hoop, and Gregory Beroza. Machine learning for data-driven discovery in solid Earth geoscience. *Science*, 363:eaau0323, March 2019.

[64] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4):045002, December 2019.

[65] Gabriel R Schleder, Antonio C M Padilha, Carlos Mera Acosta, Marcio Costa, and Adalberto Fazzio. From DFT to machine learning: Recent approaches to materials science–a review. *Journal of Physics: Materials*, 2(3):032001, May 2019.

[66] Keith Butler, Daniel Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh. Machine learning for molecular and materials science. *Nature*, 559, July 2018.

[67] Hanxun Jin, Enrui Zhang, and Horacio D. Espinosa. Recent advances and applications of machine learning in experimental solid mechanics: A review. *Applied Mechanics Reviews*, 75(6):061001, 2023.

[68] Zhanzhao Li, Jinyoung Yoon, Rui Zhang, Farshad Rajabipour, Wil III, Ismaila Dabo, and Aleksandra Radlińska. Machine learning in concrete science: Applications, challenges, and best practices. *npj Computational Materials*, 8:127, June 2022.

[69] Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. Neural-network quantum state tomography. *Nature Physics*, 14, May 2018.

[70] Ruben Ohana, Michael McCabe, Lucas Thibaut Meyer, Rudy Morel, Fruzsina Julia Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Stuart B. Dalziel, Drummond Buschman Fielding, Daniel Fortunato, Jared A. Goldberg, Keiya Hirashima, Yan-Fei Jiang, Rich Kerswell, Suryanarayana Maddu, Jonah M. Miller, Payel Mukhopadhyay, Stefan S. Nixon, Jeff Shen, Romain Watteaux, Bruno Régaldo-Saint Blancard, François Rozet, Liam Holden Parker, Miles Cranmer, and Shirley Ho. The well: A large-scale collection of diverse physics simulations for machine learning. In *The Thirty-Eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[71] Mike Walmsley, Chris Lintott, Tobias Géron, Sandor Kruk, Coleman Krawczyk, Kyle W Willett, Steven Bamford, Lee S Kelvin, Lucy Fortson, Yarin Gal, William Keel, Karen L Masters, Vihang Mehta, Brooke D Simmons, Rebecca Smethurst, Lewis Smith, Elisabeth M Baeten, and Christine Macmillan. Galaxy Zoo DECaLS: Detailed visual morphology measurements from volunteers and deep learning for 314 000 galaxies. *Monthly Notices of the Royal Astronomical Society*, 509(3):3966–3988, September 2021.

[72] Raphael Townshend, Martin Vögele, Patricia Suriana, Alex Derry, Alexander Powers, Yianni Laloudakis, Sidhika Balachandar, Bowen Jing, Brandon Anderson, Stephan Eismann, Risi Kondor, Russ Altman, and Ron Dror. ATOM3D: Tasks on molecules in three dimensions. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

[73] H Liu, P Chen, X Zhai, KG Huo, S Zhou, L Han, and G Fan. PPB-affinity: Protein-protein binding affinity dataset for AI-based protein drug discovery. *Scientific data*, 11(1):1316, December 2024.

[74] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N. Kinch, R. Dustin Schaeffer, Carla Millán, Heewook Park, Cole Adams, Craig R. Glassman, Andy DeGiovanni, Jose H. Pereira, Andria V. Rodrigues, Alberdina A. van Dijk, Andrea C. Ebrecht, D. J. Opperman, Thomas Sagmeister, Christoph Buhlheller, Tea Pavkov-Keller, Manoj K. Rathinaswamy, Ujjval Dalwadi, Christopher K. Yip, John E. Burke, K. Christopher Garcia, Nick V. Grishin, Paul D. Adams, Randy J. Read, and David Baker. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

[75] Mihaly Varadi, Damian Bertoni, Paulyna Magana, Urmila Paramval, Ivanna Pidruchna, Malarvizhi Radhakrishnan, Maxim Tsenkov, Sreenath Nair, Milot Mirdita, Jingi Yeo, Oleg Kovalevskiy, Kathryn Tunyasuvunakool, Agata Laydon, Augustin Žídek, Hamish Tomlinson, Dhavanthi Hariharan, Josh Abrahamson, Tim Green, John Jumper, Ewan Birney, Martin Steinegger, Demis Hassabis, and Sameer Velankar. AlphaFold Protein Structure Database in 2024: Providing structure coverage for over 214 million protein sequences. *Nucleic Acids Research*, 52(D1):D368–D375, 2024.

[76] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Koray Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Charlie Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stig Nikolov, Rishub Jain, Jonas Adler, Tom Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Marta Pacholska, Thomas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[77] Lowik Chanussot, Abhishek Das, Siddharth Goyal, Thibaut Lavril, Muhammed Shuaibi, Morgane Riviere, Kevin Tran, Javier Heras-Domingo, Caleb Ho, Weihua Hu, Aini Palizhati, Anuroop Sriram, Brandon Wood, Junwoong Yoon, Devi Parikh, C. Lawrence Zitnick, and Zachary Ulissi. Open catalyst 2020 (oc20) dataset and community challenges. *ACS Catalysis*, 11(10):6059–6072, May 2021.

[78] A. Merchant, S. Batzner, Samuel S. Schoenholz, Muratahan Aykol, Gabin Cheon, and Ekin D. Cubuk. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85, 2023.

[79] Huan Li, Hao Zheng, Ting Yue, Zhi Xie, Shanshan Yu, Jun Zhou, Tarun Kapri, Yiming Wang, Zhi Cao, Hong Zhao, Akerke Kemelbay, Jie He, Guojing Zhang, Paul F. Pieters, Eric A. Dailing, James R. Cappiello, Miquel Salmeron, Xiaodan Gu, Ting Xu, Peng Wu, Yuzhang Li, Karl B. Sharpless, and Yi Liu. Machine learning-accelerated discovery of heat-resistant polysulfates for electrostatic energy storage. *Nature Energy*, 10(1):90–100, 2025.

[80] Thomas N. Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, pages 2688–2697, 2018.

[81] Tommy Tai and Ching Hua Lee. Zoology of non-Hermitian spectra and their graph topology. *Physical Review B*, 107(22):L220301, June 2023.

[82] Rijia Lin, Tommy Tai, Mengjie Yang, Linhu Li, and Ching Hua Lee. Topological Non-Hermitian skin effect. *Frontiers of Physics*, 18(5):53605, October 2023.

[83] Yuncheng Xiong and Haiping Hu. Graph Morphology of Non-Hermitian Bands, November 2023.

[84] Hong-Yi Wang, Fei Song, and Zhong Wang. Amoeba Formulation of Non-Bloch Band Theory in Arbitrary Dimensions. *Physical Review X*, 14(2):021011, April 2024.

[85] M. Z. Hasan and C. L. Kane. Colloquium: Topological insulators. *Reviews of Modern Physics*, 82(4):3045–3067, 2010.

[86] Weihua Hu, Matthias Fey, Hongyu Ren, Maho Nakata, Yuxiao Dong, and Jure Leskovec. OGB-LSC: A Large-Scale Challenge for Machine Learning on Graphs.

[87] Landon Butler, Alejandro Parada-Mayorga, and Alejandro Ribeiro. Convolutional learning on multigraphs. *IEEE Transactions on Signal Processing*, 71:933–946, 2023.

[88] Béni Egressy, Luc von Niederhäusern, Jovan Blanuša, Erik Altman, Roger Wattenhofer, and Kubilay Atasu. Provably powerful graph neural networks for directed multigraphs. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, 2024.

[89] Nada Chaari, Mohammed Amine Gharsallaoui, Hatice Camgöz Akdağ, and Islem Rekik. Multigraph classification using learnable integration network with application to gender fingerprinting. *Neural Networks*, 151:250–263, July 2022.

[90] Taki Youssef, Elmoukhtar Zemmouri, and Anas Bouzid. STM-GCN: A spatiotemporal multi-graph convolutional network for pedestrian trajectory prediction. *The Journal of Supercomputing*, 79(18):20923–20937, December 2023.

[91] Kexin Ding, Mu Zhou, Zichen Wang, Qiao Liu, Corey W. Arnold, Shaoting Zhang, and Dimitri N. Metaxas. Graph convolutional networks for multi-modality medical imaging: Methods, architectures, and clinical applications, 2022.

[92] Emőke-Agnes Horvat and Katharina Anna Zweig. *Multiplex Networks*, pages 1430–1434. Springer New York, New York, NY, 2018.

[93] Chukwuemeka Iddianozie and Gavin McArdle. Towards robust representations of spatial networks using graph neural networks. *Applied Sciences*, 11(15):6918, 2021.

[94] Fivos Papadimitriou. Spatial artificial intelligence.

[95] Song Gao. *Geospatial artificial intelligence (GeoAI)*, volume 10. Oxford University Press New York, 2021.

[96] Krzysztof Janowicz, Song Gao, Grant McKenzie, Yingjie Hu, and Budhendra Bhaduri. Geoai: spatially explicit artificial intelligence techniques for geographic knowledge discovery and beyond, 2020.

[97] Grace Chu, Brian Potetz, Weijun Wang, Andrew Howard, Yang Song, Fernando Brucher, Thomas Leung, and Hartwig Adam. Geo-aware networks for fine-grained recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[98] Yingjie Hu, Song Gao, Dalton Lunga, Wenwen Li, Shawn Newsam, and Budhendra Bhaduri. Geoai at acm sigspatial: progress, challenges, and future directions. *Sigspatial Special*, 11(2):5–15, 2019.

[99] Tomasz Danel, Przemysław Spurek, Jacek Tabor, et al. Spatial graph convolutional networks. *arXiv preprint arXiv:1909.05310*, 2019.

[100] John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 15820–15831, 2019.

[101] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[102] T.C. Lee, R.L. Kashyap, and C.N. Chu. Building Skeleton Models via 3-D Medial Surface Axis Thinning Algorithms. *CVGIP: Graphical Models and Image Processing*, 56(6):462–478, November 1994.

[103] Shenyang Huang, Farimah Poursafaei, Jacob Danovitch, Matthias Fey, Weihua Hu, Emanuele Rossi, Jure Leskovec, Michael Bronstein, Guillaume Rabusseau, and Reihaneh Rabbany. Temporal Graph Benchmark for Machine Learning on Temporal Graphs, September 2023.

[104] Xianquan Yan, Hakan Akgün, Kenji Kawaguchi, Duane Loh, and Ching Hua Lee. HSG-12M `https://doi.org/10.7910/DVN/PYDSSQ`, 2025.

[105] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks, February 2017.

[106] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks, February 2018.

[107] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How Powerful are Graph Neural Networks?, February 2019.

[108] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[109] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[110] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[111] Ke Ye and Lek-Heng Lim. Every matrix is a product of Toeplitz matrices. *Foundations of Computational Mathematics*, 15(1):1–25, 2015.

[112] Zhesen Yang, Kai Zhang, Chen Fang, and Jiangping Hu. Non-Hermitian bulk-boundary correspondence and auxiliary generalized Brillouin zone theory. *Physical Review Letters*, 125(22):226402, November 2020.

[113] Russell Yang, Jun Wei Tan, Tommy Tai, Jin Ming Koh, Linhu Li, Stefano Longhi, and Ching Hua Lee. Designing non-Hermitian real spectra through electrostatics. *Science Bulletin*, 67(18):1865–1873, September 2022.

[114] Anliang Wang, Xiaolong Yan, and Zhijun Wei. Imagepy: an open-source, python-based and platform-independent software package for bioimage analysis. *Bioinformatics*, 34(18):3238–3240, 2018.

[115] Juan Nunez-Iglesias, Adam J. Blanch, Oliver Looker, Matthew W. Dixon, and Leann Tilley. A new Python library to analyse skeleton images confirms malaria parasite remodelling of the red blood cell membrane skeleton. *PeerJ*, 6:e4312, February 2018.

[116] Zhenyu Yang, Ge Zhang, Jia Wu, Jian Yang, Quan Z. Sheng, Shan Xue, Chuan Zhou, Charu Aggarwal, Hao Peng, Wenbin Hu, Edwin Hancock, and Pietro Liò. State of the Art and Potentialities of Graph-level Learning, May 2023.

[117] Yuanqi Du, Shiyu Wang, Xiaojie Guo, Hengning Cao, Shujie Hu, Junji Jiang, Aishwarya Varala, Abhinav Angirekula, and Liang Zhao. GraphGT: Machine Learning Datasets for Graph Generation and Transformation.

[118] Phitchaya Mangpo Phothilimthana, Sami Abu-El-Haija, Kaidi Cao, Bahare Fatemi, Mike Burrows, Charith Mendis, and Bryan Perozzi. TpuGraphs: A Performance Prediction Dataset on Large Tensor Computational Graphs.

[119] C. Morris, F. Bause, M. Neumann, K. Kersting, and P. Mutzel. Tudataset: A collection of benchmark datasets for learning with graphs. *CoRR*, abs/2007.08663, 2020.

[120] P. D. Dobson and A. J. Doig. Distinguishing enzyme structures from non-enzymes without alignments. *J. Mol. Biol.*, pages 771–783, 2003.

[121] A. Said. Neurograph: A graph database for neuroscience. *Front. Neuroinform.*, 2020.

[122] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1010–1019, 2016.

[123] M. Zitnik, R. Sosič, M. W. Feldman, and J. Leskovec. Evolution of resilience in protein interactomes across the tree of life. *Proc. Natl. Acad. Sci. U.S.A.*, 116(10):4426–4433, 2019.

[124] W. Hu, M. Fey, M. Zitnik, ..., and J. Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Proc. NeurIPS*, pages 22118–22133, 2020.

[125] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[126] N. Kriege and P. Mutzel. Subgraph matching kernels for attributed graphs. *J. Mach. Learn. Res.*, 12:291–298, 2012.

[127] H. Altae-Tran, B. Ramsundar, A. S. Pappu, and V. Pande. Low data drug discovery with one-shot learning. *ACS Cent. Sci.*, 3(4):283–293, 2017.

[128] G. Subramanian, B. Ramsundar, V. Pande, and R. A. Denny. Computational modeling of b-secretase 1 (bace-1) inhibitors using ligand-based approaches. *J. Chem. Inf. Model.*, 56(10):1936–1949, 2016.

[129] K. M. Gayvert, N. S. Madhukar, and O. Elemento. A data-driven approach to predicting successes and failures of clinical trials. *Cell Chem. Biol.*, 23(10):1294–1301, 2016.

[130] I. F. Martins, A. L. Teixeira, L. Pinheiro, and A. O. Falcao. A bayesian approach to in silico blood–brain barrier penetration modeling. *J. Chem. Inf. Model.*, 52(6):1686–1697, 2012.

[131] Tox21 Data Challenge. Tox21 data challenge 2014. https://tripod.nih.gov/tox/challenge, 2014.

[132] A. M. Richard, R. S. Judson, K. A. Houck, and ... Toxcast chemical landscape: Paving the road to 21st century toxicology. *Chem. Res. Toxicol.*, 29(8):1225–1251, 2016.

[133] Vijay Prakash Dwivedi, Ladislav Rampášek, Mikhail Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long Range Graph Benchmark.

[134] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 990–998, 2008.

[135] V. P. Dwivedi, C. K. Joshi, T. Laurent, Y. Bengio, and X. Bresson. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*, 2020.

[136] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. In *Technical Report, University of Toronto*, 2009.

[137] Y. LeCun. The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/, 1998.

[138] Benjamin Hilprecht and Carsten Binnig. Zero-shot cost models for out-of-the-box learned cost prediction, 2022.

[139] Lianmin Zheng, Ruochen Liu, Junru Shao, Tianqi Chen, Joseph Gonzalez, Ion Stoica, and Ameer Haj-Ali. Tenset: A large-scale program performance dataset for learned tensor compilers. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

[140] Hosagrahar V Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Papakonstantinou, Jignesh M Patel, Raghu Ramakrishnan, and Cyrus Shahabi. Big data and its technical challenges. *Communications of the ACM*, 57(7):86–94, 2014.

[141] Chao Chen. Freeway performance measurement system (pems). Technical report, University of California, Berkeley, 2002.

A Mathematical Background - *Hamiltonian Spectral Graph*

A.1 The Hamiltonian and Energy Spectrum in 1D Tight-Binding Systems

In physical sciences, it is customary to represent and study a system through its Hamiltonian matrix. The **energy spectrum**, which refers to the set of eigenvalues of this matrix, reveals the energy band structure—a central object of study in condensed matter physics. Let us consider a generic 1D tight-binding Hamiltonian:

$$\boldsymbol{H} = \sum_{x,j} T_j \hat{c}_x^\dagger \hat{c}_{x+j} \tag{7}$$

where x and j index unit cells and hopping lengths, respectively; $\hat{c}_x$ is the annihilation operator for the x-th unit cell. Each term T_j represents the transition amplitude of a particle hopping from site $x+j$ to x (as $\hat{c}_{x+j}$ annihilates at $x+j$ and $\hat{c}_x^\dagger$ creates at x). The L^2 norm of the amplitude, $|T_j|^2$, equals the corresponding transition probability. These hopping terms can generically be complex or even matrix-valued to account for multi-band systems.

The matrix representation of this Hamiltonian in real space, $\boldsymbol{H}_{\text{real}}$, for which $(\boldsymbol{H}_{\text{real}})_{x,x'} = T_{x'-x}$, is a Toeplitz matrix—a matrix in which each descending diagonal from left to right is constant:

$$\boldsymbol{H}_{\text{real}} = \begin{pmatrix} T_0 & T_1 & T_2 & & \cdots & & 0 \\ T_{-1} & T_0 & T_1 & T_2 & & & \\ T_{-2} & T_{-1} & T_0 & T_1 & \ddots & & \vdots \\ & T_{-2} & T_{-1} & T_0 & \ddots & & \\ \vdots & & \ddots & \ddots & \ddots & & T_2 \\ & & & & & T_0 & T_1 \\ 0 & & \cdots & & T_{-2} & T_{-1} & T_0 \end{pmatrix} \tag{8}$$

If there are L sites in total, $\boldsymbol{H}_{\text{real}} \in \mathbb{C}^{L\times L}$. In general, $T_j \neq T_{-j}^*$ (where T_{-j}^* is the complex conjugate of T_{-j}), which breaks the Hermiticity of the Hamiltonian, i.e., $\boldsymbol{H}^\dagger := (\boldsymbol{H}^*)^T \neq \boldsymbol{H}$. Consequently, the eigenvalues can take on complex values. The energy spectrum is obtained by diagonalizing $\boldsymbol{H}_{\text{real}}$.

A.2 Hamiltonian Spectral Graph: Emergent Topology in the Thermodynamic Limit

For non-Hermitian systems, the energy eigenvalues form intricate patterns in the complex plane. The **spectral graph** $\mathcal{G}$ emerges from the energy spectra under open boundary conditions (OBC) in the **thermodynamic limit** (i.e., as the system size $L \to \infty$). In this limit, the discrete energies become continuous, and their loci trace out a planar graph on the complex plane [81, 83]. Figure A3 illustrates this emergence: the OBC energy spectra for finite system sizes $L = 50$ and $L = 150$ for a non-Hermitian lattice (whose characteristic polynomial is $P(z, E) = -z^{-2} - E - z + z^4$) clearly approach a 3-Cayley tree as L increases. Figure A3c shows the corresponding **density of states** (DOS) in the $L \to \infty$ limit.

The spectral graphs of different lattices exhibit a kaleidoscope of geometries, including arcs, loops, and more exotic shapes resembling stars, kites, braids, and even rockets [81, 82], as showcased in Figure A4. These structures represent an uncharted band topology, embedding hidden symmetries and graph topological transitions that lie beyond standard homotopy-based frameworks [85]. In effect,

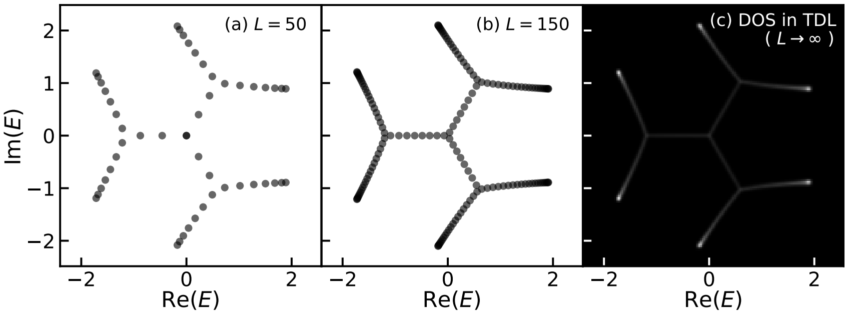

Figure A3: **The emergence of spectral graphs. (a)-(b)** show the OBC energy spectra with increasing system size $L = [50, 150]$, of the non-Hermitian lattice whose characteristic polynomial is $P(z, E) = -z^{-2} - E - z + z^4$. In the thermodynamic limit ($L \to \infty$), the spectra becomes a *band* continuum and the energy loci traces out a planar graph on the complex plain, namely the *spectral graph*. For this particular example, it is a 3-Cayley tree. **(c)** shows the corresponding density of states when $L \to \infty$.

a new class of topological invariants appears—those tied to the global geometry of the eigenvalue loci.

However, accurately diagonalizing a large non-Hermitian matrix $\boldsymbol{H}_{\text{real}}$ to obtain the OBC spectrum is notoriously hard [112], let alone for an infinite-sized matrix (i.e. an operator). This necessitates a more sophisticated theoretical approach.

A.3 Theoretical Framework: Non-Bloch Band Theory

The standard approach to analyze such systems, guided by non-Bloch band theory, begins with a Fourier transformation.

Fourier Transformation and the Bloch Hamiltonian. Fourier transforming the real-space Hamiltonian (second quantized form equation 7 or its matrix form equation 8) yields the Bloch Hamiltonian:

$$\boldsymbol{H}(z) = \sum_j \boldsymbol{T}_j z^j, \quad z := e^{ik} \tag{9}$$

which is a matrix-valued Laurent polynomial of the phase factor $z = e^{ik}$, where k is the crystal momentum.

The Characteristic Polynomial $P(z, E)$. The energy dispersion relation is found by solving the secular equation. The **characteristic polynomial** of the Hamiltonian is defined as:

$$P(z, E) = \det[\boldsymbol{H}(z) - E\,\boldsymbol{I}] = \sum_{n=-p}^{q} \left(\sum_{m=0}^{s} c_{n,m} E^m \right) z^n = \sum_{n=-p}^{q} a_n(E) z^n = 0 \tag{10}$$

Here, $a_n(E) = \sum_{m=0}^{s} c_{n,m} E^m$ are coefficients that are themselves polynomials in energy E, and s is the maximum power of E (typically the dimension of $\boldsymbol{H}(z)$ if it's a matrix). This equation is also known as the energy-momentum dispersion. The presence of z^n monomial (i.e. if $a_n(E) \neq 0$) indicates the existence of a hopping to the n-th neighbor on the left (see figure 2a).

The key insight from non-Bloch band theory is that the spectral graph can be obtained entirely from the roots $\{z_i(E)\}$ of $P(z, E) = 0$. These roots are sorted by their magnitudes: $|z_1(E)| \leq |z_2(E)| \leq \cdots \leq |z_{p+q}(E)|$.

Characteristic Polynomial Class $\mathcal{C}_P$. The algebraic form of $P(z, E)$—specifically, which monomials z^n are present with non-zero coefficient functions $a_n(E)$—plays a crucial role in determining the spectral graph. We formally define the **characteristic polynomial class** $\mathcal{C}_P$ as the union of a **binary**

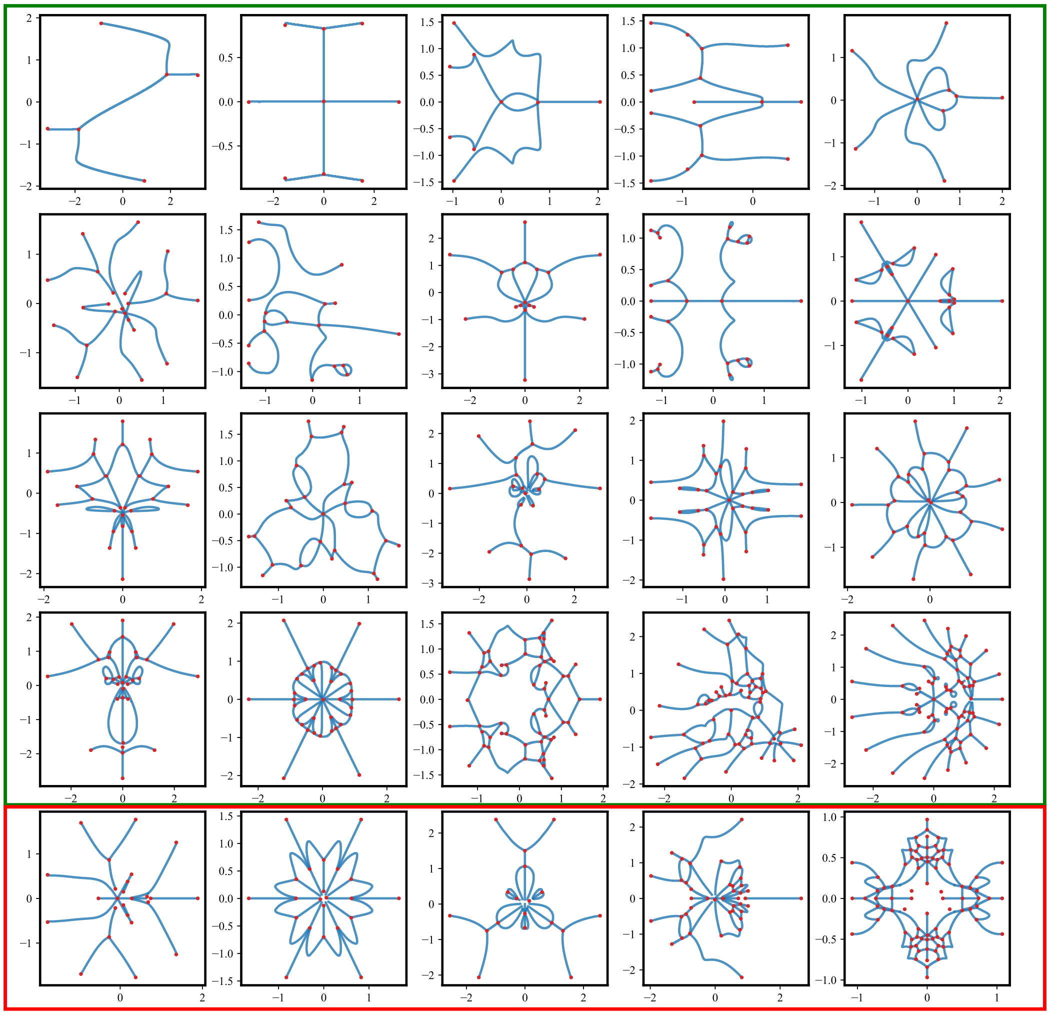

Figure A4: **A Gallery of Spectral Graphs.** The top four rows highlight the intricate structures characteristic of spectral graphs. The bottom row illustrates the distinct phenomenon we refer to as *component fragmentation* (Section 6)—some nodes in theory should be connected, however its surrounding low density of states limits accurate edge detection, causing certain nodes to be *fragmented* into disjoint nodes, often leading to fragmentation of an otherwise connected component. The phenomena often occurs for high-band and long-range hopping crystals.

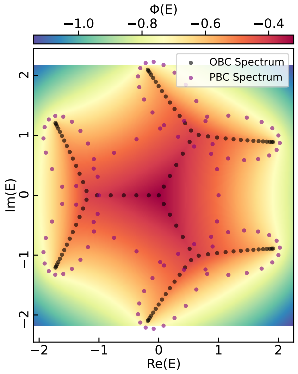

Figure A5: **Spectral Collapse & Spectral Potential.** PBC spectrum usually appears as circles and loops; changing to OBC, the spectrum *collapses* into a graph skeleton. The spectral graph resides on the *ridges* of the potential landscape, $\Phi(E)$.

coefficient vector $\boldsymbol{b}$ (derived from $P(z, E)$) and its counterpart $\boldsymbol{b}'$ (derived from $P(z^{-1}, E)$):

$$\mathcal{C}_P = \boldsymbol{b} \cup \boldsymbol{b}' \tag{11}$$

$$\boldsymbol{b} = (b_{-p}, \dots, b_q), \quad \text{where } b_n := 1 \text{ if } a_n(E) \not\equiv 0; \text{ and } b_n := 0 \text{ otherwise.} \tag{12}$$

$$\boldsymbol{b}' \text{ is the binary coefficient vector for } P(z^{-1}, E) \text{, constructed similarly to } \boldsymbol{b}. \tag{13}$$

The vector $\boldsymbol{b}$ indicates the presence ($b_n = 1$) or absence ($b_n = 0$) of the z^n term in $P(z, E)$. The spectral graph is invariant under parity transformation[10], which corresponds to $z \to z^{-1}$. If $\boldsymbol{b}$ and $\boldsymbol{b}'$ are identical (e.g., if $P(z, E)$ has coefficients $a_n(E)$ such that the set of non-zero $a_n(E)$ is symmetric upon $n \to -n$), then this palindrome alone constitutes the class $\mathcal{C}_P = \boldsymbol{b}$. We find that the characteristic polynomial class $\mathcal{C}_P$ is a key criterion for classifying spectral graphs and is thus the target for inverse classification tasks.

Example: The 1D single-band example shown in Figure A3 comes from the Bloch Hamiltonian $\boldsymbol{H}(z) = -z^{-2} - z + z^4$. Since this is a scalar Hamiltonian, its characteristic polynomial is $P(z, E) = \boldsymbol{H}(z) - E = -z^{-2} - z + z^4 - E$. The terms present correspond to z^{-2} (coefficient $a_{-2}(E) = -1$), z^0 (coefficient $a_0(E) = -E$), z^1 (coefficient $a_1(E) = -1$), and z^4 (coefficient $a_4(E) = 1$). Thus, $p = 2, q = 4$. The binary coefficient vector is $\boldsymbol{b} = (b_{-2}, b_{-1}, b_0, b_1, b_2, b_3, b_4) = (1, 0, 1, 1, 0, 0, 1)$. The parity-transformed polynomial is $P(z^{-1}, E) = -(z^{-1})^{-2} - E - (z^{-1}) + (z^{-1})^4 = -z^2 - E - z^{-1} + z^{-4}$. The terms present in $P(z^{-1}, E)$ are z^{-4}, z^{-1}, z^0, z^2. The binary vector for this polynomial (listing coefficients for powers from its $z^{-p'}$ to $z^{q'}$) is $\boldsymbol{b}' = (1, 0, 0, 1, 1, 0, 1)$ (corresponding to $a'_{-4}, a'_{-3}, \dots, a'_2$). Thus, for this example, $\mathcal{C}_P = (1, 0, 1, 1, 0, 0, 1) \cup (1, 0, 0, 1, 1, 0, 1)$.

Limitations of Standard Bloch Theory (Periodic Boundary Conditions - PBC). Under periodic boundary conditions (PBC), Bloch theory predicts that for a finite chain of N sites, the allowed momenta are $k \in \{\frac{2\pi}{N}, \frac{4\pi}{N}, \dots, \frac{2\pi N}{N}\}$. For an infinite chain ($N \to \infty$), the spectrum is obtained by letting k span the Brillouin zone (BZ), $k \in [0, 2\pi)$ (so $|z| = 1$), typically forming continuous bands (circles or loops in the complex energy plane for non-Hermitian systems).

However, for non-Hermitian systems, eigenstates under OBC often exhibit the **non-Hermitian skin effect** [82], where a macroscopic number of eigenstates localize at the boundaries. Consequently, the PBC and OBC spectra can be qualitatively different. As one evolves the system from PBC to OBC (e.g., by turning off boundary hoppings), the PBC spectrum (loops) often *collapses* inwards to form the skeletal structure of the OBC spectral graph, as depicted in Figure A5.

Non-Bloch Band Theory and the Generalized Brillouin Zone (GBZ). Since standard Bloch theory (based on translation symmetry) is inapplicable under OBC, **non-Bloch band theory** is employed. This theory introduces the concept of the **generalized Brillouin zone** (GBZ), which is a path in the complex plane of $z = e^{ik}$ (or a region in the complex k-plane) that correctly determines the continuous OBC spectrum in the thermodynamic limit. The OBC spectrum is obtained by taking z from the GBZ (or $k \in \text{GBZ} \subsetneq \mathbb{C}$) in $P(z, E) = 0$. The imaginary part of k, denoted as

[10] I.e., spatial inversion about the origin ($x \to -x$), or put differently, flipping the 1D lattice from left to right. In terms of $\boldsymbol{H}_{\text{real}}$, it is equivalent to transpose the matrix ($T_j \to T_{-j}$) which does not change eigenvalues.

$\kappa := \mathrm{Im}(k) = -\log|z|$, is also called the inverse decay length or inverse skin depth, quantifying the localization of skin modes.

The GBZ is determined by the condition that for an energy E to be part of the OBC spectrum (i.e., $E \in \mathcal{G}$), the magnitudes of the p-th and $(p+1)$-th roots of $P(z, E) = 0$ (when sorted by magnitude as $|z_1| \leq \cdots \leq |z_{p+q}|$) must be equal:

$$|z_p(E)| = |z_{p+1}(E)|, \quad E \in \mathcal{G}. \tag{14}$$

Here, p is lowest degree of z in the characteristic polynomial $P(z, E)$ (i.e. $a_{-p}(E)$ is the coefficient of the lowest power z^{-p}). In other words, the spectral graph $\mathcal{G}$ is the locus of E values that satisfy this condition.

A.4 The Shortcut to Spectral Graph via Electrostatic Analogy.

Density of States $\rho(E)$. The **density of states** (DOS) is a useful quantity to describe the continuous spectrum. It is defined as the number of eigenstates per unit area on the complex energy plane, $\rho(E) = \lim_{N\to\infty} \frac{1}{N}\sum_n \nabla^2(E - \epsilon_n)$, where ϵ_n are the eigenvalues for a system of size N. An example of DOS is shown in Figure A3c.

The Spectral Potential $\Phi(E)$. Recent developments in non-Bloch theory map the problem of finding the spectral graph and DOS to a classic 2D electrostatic problem [83, 113, 84]. If we assign an electric charge $1/N$ to each eigenvalue ϵ_n (for a system of size N), the DOS and the Coulomb potential $\Phi(E)$ at a point $E \notin \mathcal{G}$ can be written as:

$$\begin{aligned}\Phi(E) &= -\lim_{N\to\infty} \frac{1}{N} \sum_{\epsilon_n} \log|E - \epsilon_n| \\ &= -\int \rho(E') \log|E - E'|\, d^2E' \end{aligned} \tag{15}$$

$$\rho(E) = -\frac{1}{2\pi}\nabla^2 \Phi(E) \tag{16}$$

where $\nabla^2 = \partial^2_{\mathrm{Re}E} + \partial^2_{\mathrm{Im}E}$ is the Laplacian operator on the complex energy plane. The Laplacian operator extracts curvature. Geometrically, this means the loci of the spectral graph $\mathcal{G}$ reside on the *ridges* of the Coulomb potential landscape $\Phi(E)$, as suggested in figure A5 and figure 2.

Efficient Calculation of $\Phi(E)$. Leveraging Szegö strong limit theorem, the spectral potential $\Phi(E)$ in equation 15 can be reduced to a more computable form:

$$\Phi(E) = -\log|a_q(E)| + \sum_{i=p+1}^{p+q} \kappa_i(E) \tag{17}$$

where $a_q(E)$ is the coefficient of z^q (the highest power of z) in $P(z, E)$ (see equation 10), and $\kappa_i(E) = -\log|z_i(E)|$ are the inverse decay lengths associated with the q largest roots $z_i(E)$ of $P(z, E) = 0$ (these are $z_{p+1}, \ldots, z_{p+q}$ in the sorted list). Although equation 15 is strictly defined for $E \notin \mathcal{G}$, the expression in equation 17 can be analytically continued to the entire complex plane [83]. This allows the construction of the potential $\Phi(E)$ merely from knowing the characteristic polynomial $P(z, E)$, thereby obviating the need for direct diagonalization of large real-space Hamiltonians and avoiding the rapid accumulation of numerical errors associated with such diagonalizations.

B Poly2Graph Pipeline Details

Armed with the above theoretical guidance, we implement the transformations numerically, and then integrate a few computer vision techniques [102, 114, 115] to construct the spectral graph given its characteristic polynomial (or Bloch Hamiltonian). This appendix complements section 3. The core procedure of `Poly2Graph` algorithm ("Characteristic **Poly**nomial **to** Spectral **Graph**") is summarized in algorithm 1.

B.1 Initialization and Input

Poly2Graph accepts diverse input formats for the 1-D tight-binding crystal. It can initialize from a Bloch Hamiltonian $\boldsymbol{H}(z)$ or directly from its characteristic polynomial $P(z, E)$. Supported formats

Algorithm 1: `Poly2Graph`: Characteristic **Poly**nomial **to** Spectral **Graph**

```
Input: (1) $\boldsymbol{H}(z)$ or $P(z,E) := \det[\boldsymbol{H}(z) - E\,\boldsymbol{I}]$
 # ↑ Hamiltonian or its characteristic polynomial
Input: (2) Energy Domain: $\Omega \subset \mathbb{C}$ such that $\Omega \supsetneq \mathcal{G}$ (spectral graph)
Output: Spectral Graph: $\mathcal{G} \in$ networkx.MultiGraph
begin
    # Build the characteristic polynomial if only $\boldsymbol{H}(z)$ was given
    if input $\boldsymbol{H}(z)$ then
        $P(z,E) = \det[\boldsymbol{H}(z) - E\,\boldsymbol{I}] = \sum_{n=-p}^{q} a_n(E)\, z^n$

    # Stage 1:  Coarse computation over initial energy grid $\Omega$
    (Parallel) for $E \in \Omega$ do
        # Solve roots
        $\{z_i(E)\} = \mathrm{Sort}[\mathrm{Roots}(P(z,E))]$ s.t. $|z_1| \leq \cdots \leq |z_{p+q}|$
        # Compute spectral potential
        $\Phi(E) = -\log|a_q(E)| - \sum_{i=p+1}^{p+q} \log\big|z_i(E)\big|$
        # Compute Density of States (DOS)
        $\rho(E) = -\frac{1}{2\pi}\,\nabla^2\Phi(E)$
    # Identify regions of interest
    coarse_mask = dilate(binarize($\{\rho(E)\}_{E\in\Omega}$))
    # Define refined energy grid
    $\Omega'$ = get_masked_subgrid(coarse_mask)

    # Stage 2:  Refined computation within masked regions $\Omega'$
    (Parallel) for $E \in \Omega'$ do
        # Re-solve roots at higher resolution
        $\{z_i(E)\} = \mathrm{Sort}[\mathrm{Roots}(P(z,E))]$
        # Recompute spectral potential
        $\Phi'(E) = -\log|a_q(E)| - \sum_{i=p+1}^{p+q} \log\big|z_i(E)\big|$
        # Recompute DOS
        $\rho'(E) = -\frac{1}{2\pi}\,\nabla^2\Phi'(E)$
    # Combine coarse and refined DOS for full high-resolution image
    $\rho_{\text{final}}(E)$ = combine($\{\rho(E)\}_{E\in\Omega\setminus\Omega'}, \{\rho'(E)\}_{E\in\Omega'}$)
    # Binarize high-resolution DOS
    final_binarized_image = binarize($\{\rho_{\text{final}}(E)\}_{E\in\Omega}$)
    # Extract one-pixel-wide skeleton
    graph_skeleton = skeletonize(final_binarized_image)
    # Convert skeleton to graph object
    $\mathcal{G}$ = skeleton2graph(graph_skeleton)
    # Post-processing
    $\mathcal{G}$ = merge_nearby_nodes($\mathcal{G}$, tolerance)
    $\mathcal{G}$ = remove_isolated_nodes($\mathcal{G}$)
```

for $P(z,E)$ include `sympy.Matrix` (for $\boldsymbol{H}(z)$, $\boldsymbol{H}(k)$), `sympy.Poly`, or a raw string expression of the polynomial. During initialization, Poly2Graph automatically computes a full set of different representations and properties. See the tutorial appendix F, our GitHub repository, or the PyPi page for more details.

The energy domain $\Omega \subset \mathbb{C}$, which must fully contain the spectral graph $\mathcal{G}$, is also required. While users can specify a custom Ω and its discretization, by default Poly2Graph can automatically estimate a suitable region by diagonalizing a small real-space Hamiltonian (typically $L = 40$ unit cells) and applying a small padding.

B.2 Accelerated Root Finding

As detailed in section 3, solving for the roots $\{z_i(E)\}$ of $P(z, E) = 0$ for each energy E in the discretized domain Ω is the primary computational bottleneck. To achieve the reported performance gains (six orders of magnitude speedup and 20-40× memory efficiency over previous methods [81] on default settings), we employ a specialized root-finding strategy.

For a characteristic polynomial $P(z, E) = \sum_{j=-p}^{q} a_j(E) z^j$, its roots are equivalent to the eigenvalues of its Frobenius companion matrix $\mathbf{F}(E)$. For a polynomial of degree $d = p + q$, the companion matrix is a $d \times d$ matrix constructed from the coefficients:

$$\mathbf{F}(E) = \begin{pmatrix} 0 & 0 & \cdots & 0 & -a_{-p}(E)/a_q(E) \\ 1 & 0 & \cdots & 0 & -a_{-p+1}(E)/a_q(E) \\ 0 & 1 & \cdots & 0 & -a_{-p+2}(E)/a_q(E) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -a_{q-1}(E)/a_q(E) \end{pmatrix}. \tag{18}$$

This formulation holds for complex coefficients $a_j(E) \in \mathbb{C}$.

Poly2Graph constructs a batch of these companion matrices for each $E \in \Omega$, where Ω is the discretized grid of energy values. This batch is then processed by a parallelized eigensolver. The implementation automatically detects the availability of `TensorFlow` or `PyTorch` backends, leveraging them for hardware acceleration, including optional GPU support via CUDA. To optimize memory and computation, calculations are performed using single precision (float32), which has been found sufficient for high-fidelity spectral graph extraction.

B.3 Adaptive Resolution and Image Processing

The adaptive resolution strategy, outlined in section 3, is crucial for computational tractability.

1. **Coarse Identification**: The spectral potential $\Phi(E)$ (equation 3) and DOS $\rho(E)$ (equation 4) are computed on an initial, moderately resolved grid (e.g., 256 × 256). The DOS image is binarized and morphologically dilated to create a conservative mask Ω' covering potential graph regions.
2. **Refined Calculation**: Within this mask Ω', each pixel is subdivided (e.g., into a 4×4 subgrid), and $\Phi(E)$ and $\rho(E)$ are recomputed at this higher resolution.

This two-stage process achieves high effective resolution (e.g., 1024×1024) while minimizing redundant calculations in empty regions of the complex energy plane.

The resulting high-resolution DOS image is again binarized. We currently employ a *global mean threshold* ($\rho(E) > \langle \rho(E') \rangle_{E' \in \Omega}$) for binarization, as it has empirically outperformed a pool of other common global and adaptive thresholding heuristics, including Otsu, Li, Yen, Triangle, Isodata, local adaptive, and hysteresis variants for our datasets. Subsequently, iterative morphological thinning operations [102] are applied to reduce the binarized features to a one-pixel-wide skeleton, revealing the graph topology.

B.4 Graph Extraction and Post-processing

The `skeleton2graph` submodule converts the binary skeleton into a graph representation. It identifies pixels as junction nodes (three or more neighbors), leaf nodes (one neighbor), or edge points (two neighbors). The output is a `NetworkX MultiGraph` object, where each edge in particular stores its geometric path as an ordered sequence of $(\mathrm{Re}(E), \mathrm{Im}(E))$ coordinates.

To handle numerical artifacts, two post-processing steps are implemented as shown in algorithm 1:

1. `merge_nearby_nodes`: Nodes within a predefined Euclidean distance tolerance are merged. This helps consolidate fragmented junctions.
2. `remove_isolated_nodes`: Nodes with no connecting edges after the merging step are removed.

B.5 Caveats: Component Fragmentation

A notable challenge, particularly for systems with large hopping ranges or many bands, is a phenomenon we termed *component fragmentation*. As illustrated in the bottom row of figure A4, this refers to the spurious disconnection of spectral branches that should ideally form a single connected component. Fragmentation typically arises at junction nodes where the surrounding DOS is exceptionally low. In such cases, the spectral potential landscape ($\Phi(E)$) around these junctions is virtually flat, making the corresponding ridges (which correspond to edges) fall below the detection threshold of the binarization and thinning processes, due to finite floating-point precision.

While the current global mean thresholding for binarization is a robust general choice, it may struggle with complicated spectra. Ultra-low-DOS edges can be missed, leading to missing pixels in the skeleton and thus fragmentation. While more sophisticated ridge-following or adaptive local thresholding algorithms might offer improvements, they often come at a significant cost to Poly2Graph's speed and memory efficiency. Addressing this intrinsic limitation robustly remains an area for future development.

C Dataset Comparison

We present a comprehensive comparison of our dataset in terms of both structural properties and statistical metrics. Table C compiles all prominent graph datasets to the best of our knowledge. Each column is described in the caption. As illustrated, while some spatial graph datasets do exist, they generally lack rich connection geometry (RCG. Nontrivial edge patterns beyond a simple straight-line link) or support for multiple parallel edges between nodes. The dataset most similar to HSG-12M in these respects is OpenStreetMap; however, it is not designed with any ML downstream tasks, contains far fewer graphs, and is medium-scale judged by OGB criteria [86]. Moreover, although it supports non-linear edge shapes—streets connecting a pair of destinations are usually not straight-lines—the complexity of its connectivity is limited. In contrast, the edge geometries in our setting exhibit much richer geometric variation. Consequently, prior to this work, the absence of a *large-scale* multigraph learning challenge remain unaddressed.

Moreover, as shown in the table, our large-scale T-HSG-5M dataset is the only temporal dataset that includes class labels. This is particularly valuable in our setting, as different classes may exhibit distinct temporal evolution patterns. We leave the investigation of these dynamics to future research.

Additionally, as illustrated in figure 1, HSM-12M stands out as the largest classification dataset in terms of both graph count and class diversity. Although our dataset is designed for classification, it is still informative to compare it with others based on the total number of graphs and nodes. By these metrics, certain large-scale computer science datasets—such as TpuGraphs, Tenset, and MalNet—contain larger overall volumes. However, a fairer comparison emerges when we evaluate our dataset alongside those from the natural sciences, as they are constructed using similar methodologies.

To facilitate this comparison, we selected the largest datasets from the table and visualized them in figure A6. The results show that even among natural science datasets not constrained to classification tasks, ours stands out as not only competitive but also the largest in scale. These findings highlight the scope and impact of this work.

D Benchmark Details

Data Preprocessing (Featurization).

- *Node features.* Four dimensional vector: the complex coordinates of the node's position (2D), and the spectral potential value (1D) and DOS value (1D) at the node's position.
- *Edge features.* Thirteen dimensional vector: the length of the edge (1D, also serves as the weight of the edge), the average spectral potential value (1D) and average DOS value (1D) over all points on the edge, and the coordinate sequences of five equidistant points on the edge (5×2=10D).
- *Graph-level features.* None.

Common Hyperparameters. Unless stated otherwise, **all** experiments share the following configuration:

- **Hardware**: single NVIDIA RTX A5000 (24 GB).

Table A3: Overview of graph-level benchmark datasets. Each row corresponds to one dataset: **#Graphs** gives the total number of graphs; **#Classes** is the number of target labels; **#Nodes** and **#Edges** report the average number of nodes and edges per graph; **Scale** column follows the OGB Large-Scale Challenge definitions [86]: Small (S) datasets fall below the large-scale thresholds; Medium (M) datasets contain > 1 million nodes or > 10 million edges; Large (L) datasets exceed 100 million nodes or 1 billion edges.; **Attributed** indicates whether both node and edge features are present; **Spatial** denotes whether the graphs carry geometric or coordinate information (e.g. 2D, 3D, geographic coordinate system-GCS); **Temporal** flags static (S) or time-series graph data (T); **Multi** marks support for multiple edges between node pairs; and **RCG** (Rich Connection Geometry) indicates datasets whose edge geometry exhibits nontrivial patterns that go beyond simple straight-line connections. "?" entries indicate information not stated in the original paper. In addition to values extracted from the literature, some benchmark statistics were sourced from Refs. [86, 116–118].

Dataset	#Graphs	#Classes	#Nodes	#Edges	Scale	Attributed	Spatial	Temporal	Multi	RCG
Biology										
ENZYMES [12, 119]	600	6	32.6	62.1	S	–	–	S	–	–
PROTEINS [12, 119]	1.1K	2	39.1	72.8	S	✓	–	S	–	–
D&D [119, 120]	1.2K	2	284.3	715.7	S	–	–	S	–	–
ProFold [61]	76K	–	8.0	?	S	✓	3D	T	–	–
NeuroGraph [121]	23K	7	359.6	11K	M	–	–	S	–	–
Skeleton (NTU-RGB+D) [122]	56K	60	25.0	24	M	–	3D	T	–	–
ppa [123, 124]	158K	37	243.4	266.1	M	✓	–	S	–	–
Skeleton (Kinetics) [125]	260K	400	18.0	17	M	–	2D	T	–	–
Chemistry										
MUTAG [126, 119]	188	2	17.9	19.8	S	✓	–	S	–	–
SIDER [31, 127]	1.4K	2	33.6	35.4	S	✓	–	S	–	–
BACE [31, 128]	1.5K	2	34.1	36.9	S	✓	–	S	–	–
ClinTox [31, 129]	1.5K	2	26.2	27.9	S	✓	–	S	–	–
BBBP [31, 130]	2.0K	2	24.1	25.9	S	✓	–	S	–	–
Tox21 [31, 131]	7.8K	2	18.6	19.3	M	✓	–	S	–	–
ToxCast [31, 132]	8.6K	2	18.8	19.3	M	✓	–	S	–	–
Peptides-func [133]	15.5 K	–	150.9	307.3	M	✓	3D	S	–	–
Peptides-struct [133]	15.5 K	–	150.9	307.3	M	✓	3D	S	–	–
MolHIV [31, 124]	41.1K	2	25.5	27.5	M	✓	–	S	–	–
MUV [31, 29]	93.1K	2	24.2	26.3	M	✓	–	S	–	–
QM9 [21]	129K	12	18.0	18.6	M	✓	3D	S	–	–
MOSES [24]	194K	–	22	47	M	✓	3D	S	–	–
MolOpt [23]	229K	–	24	53	M	✓	3D	S	–	–
ZINC250K [22]	250K	–	23	50	M	✓	3D	S	–	–
MolPCBA [31, 124]	437.9K	2	26.0	28.1	M	✓	–	S	–	–
PCQM-Contact [133]	529.4K	–	30.1	61.1	M	✓	3D	S	–	–
ChEMBL [25]	1.8M	–	27.0	58	M	✓	3D	S	–	–
PCQM4Mv2 [86]	3.7 M	–	14.1	14.6	L	✓	3D	S	–	–
Social Networks										
IMDB-BINARY [119, 13]	1 K	2	19.8	96.5	S	–	–	S	–	–
IMDB-MULTI [119, 13]	1.5 K	3	13.0	65.9	S	–	–	S	–	–
REDDIT-BINARY [119, 13]	2 K	2	429.6	497.8	S	–	–	S	–	–
REDDIT-MULTI-5K [119, 13]	5.0 K	5	508.5	594.9	M	–	–	S	–	–
REDDIT-MULTI-12K [119, 13]	11.9 K	11	11.0	391.4	M	–	–	S	–	–
CollabNet[134]	2.3K	–	303K	207.6K	L	–	GCS	T	-	-
Computer Science										
CIFAR10 [135, 136]	60 K	10	117.6	941.1	M	✓	2D	S	–	–
MNIST [135, 137]	70 K	10	70.6	564.5	M	✓	2D	S	–	–
Database [138]	300.0 K	–	<100.0	?	M	?	–	S	–	–
MalNet [7]	1.3 M	696	15.4 K	35.2 K	L	–	–	S	–	–
TpuGraphs (Tile) [118]	12.9 M	–	40.0	?	L	?	–	S	–	–
TpuGraphs (Layout) [118]	31.1 M	–	7.7 K	?	L	?	–	S	–	–
TenSet [139]	51.6 M	–	5.0–10.0	?	L	?	–	S	–	–
Geography										
METR-LA [140]	34K	–	327.0	2.4	M	✓	GCS	T	–	–
PeMS-BAY [141]	50K	–	207	1.5	M	✓	GCS	T	–	–
OpenStreetMap [44]	110K	–	500	1.2K	M	✓	GCS	S	✓	✓
Physics										
N-body-spring [80]	3.4M	–	5.0	10	M	✓	2D	T	–	–
N-body-charged [80]	3.4M	–	25.0	3	M	✓	2D	T	–	–
T-HSG-5M	5.1M	1401	13.8	28.9	L	✓	$\mathbb{C}$-plane	T	✓	✓
HSG-12M	11.6M	1401	13.8	28.9	L	✓	$\mathbb{C}$-plane	S	✓	✓

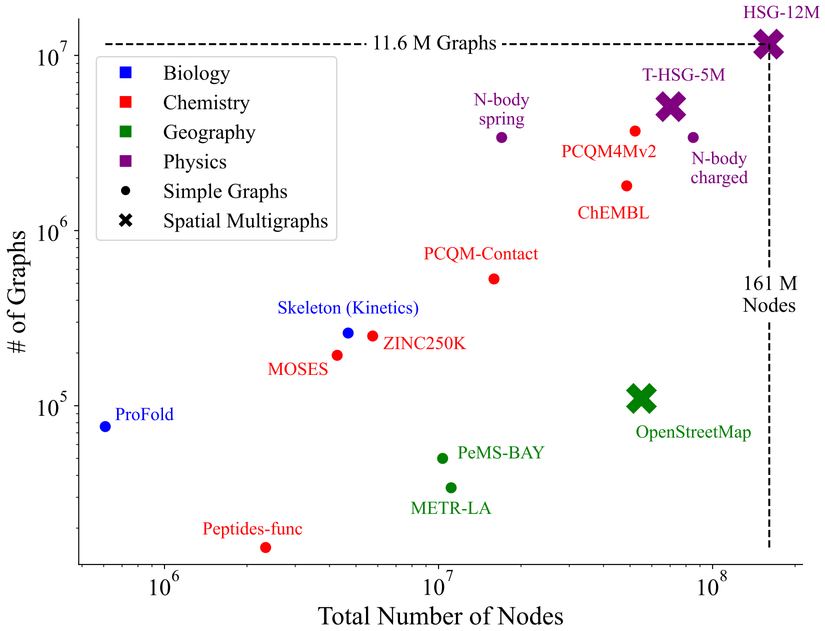

Figure A6: Number of graphs v.s. total number of nodes in HSG-12M compared to other natural science datasets. HSG-12M stands out with the highest data volume across all natural science datasets, including those not designed for classification.

- **Data splits**: three stratified splits (80 % train, 10 % validation, 10 % test) keyed by seeds `42`, `624`, and `706`.
- **Optimizer**: AdamW (AMSGrad) with zero weight decay.
- **Learning-rate schedule**: cosine annealing with warm restarts ($T_0 = 5$, $T_{\text{mult}} = 1$); initial LR 10^{-2}, floor LR 10^{-4}.
- **Architecture** for *all* four backbones (GCN, GIN, GAT, GraphSAGE): 4 message-passing layers, global-add pooling, followed by a 2-layer MLP. Dropout $= 0.1$ after every learnable layer.
- **Optimisation length**: 5 full epochs (early stopping not used).

Dataset-specific Hyperparameters. The three benchmark subsets differ only in embedding width and batch size:

Table A4: Subset-specific architectural widths and mini-batch sizes.

Subset	hidden dimension (GNN)	hidden dimension (MLP)	Batch size
`HSG-one-band`	128	128	256
`HSG-two-band`	256	256	256
`HSG-topology`	512	1500	64

Resource Usage and Throughput. Table A5 reports peak GPU memory consumption (MB/graph) and training throughput (graphs s^{-1}), each averaged over the three seeds.

Table A5: Peak GPU memory consumption (MB/graph) and training throughput (graphs s^{-1}), averaged over seeds and epochs.

Subset	**Metric**	**Backbone**			
		GCN	GraphSAGE	GAT	GIN
`HSG-one-band`	Peak MB/graph Throughput (graphs/s)	0.424 24927	0.422 25292	0.473 21344	0.431 26446
`HSG-two-band`	Peak MB/graph Throughput (graphs/s)	9.854 11823	9.865 14274	9.852 11155	9.865 15229
`HSG-topology`	Peak MB/graph Throughput (graphs/s)	157.0 1814	157.2 2217	156.9 1787	112.2 1733

E Universality of Spectral Graphs Through Toeplitz Decomposition

Claim. Any complex matrix $M \in \mathbb{C}^{N\times N}$ can be associated with a multiset of spectral graphs $\{G_i\}$ such that the collection $\{G_i\}$ represents M.

Argument. Appendix A demonstrates how a generic Toeplitz matrix, as defined in Eq.8, can naturally be interpreted as a single-band tight-binding Hamiltonian, and thus corresponds to a signature spectral graph.

This is particularly significant given the foundational role of Toeplitz matrices in linear algebra:

> *Any matrix can be expressed as a product of Toeplitz matrices* [111].

Specifically, for a generic $M^{(n\times n)}$ matrix, a decomposition into a product of $\lfloor n/2\rfloor + 1 \leq r \leq 2n+5$ Toeplitz matrices always exists. Note that the decomposition of M is not unique without further conditions on the Toeplitz factors.

Consequently, the spectral graph associated with each Toeplitz component implies that any matrix can, in principle, be represented as a *multiset* of spectral graphs. This representation can be constructed via the following steps:

1. Start with a generic matrix $M \in \mathbb{C}^{n\times n}$,
2. Decompose M into $\lfloor n/2\rfloor + 1 \leq r \leq 2n + 5$ Toeplitz matrices,
3. Apply POLY2GRAPH to each Toeplitz component to extract its corresponding spectral graph, and collect the resulting graphs into a set.

This procedure highlights the universality of our spectral graph framework and demonstrates its potential as a powerful tool for addressing diverse matrix-based challenges across disciplines. We encourage researchers to explore and apply this framework to domain-specific problems in their respective fields.

F Tutorial of `Poly2Graph` Package

`poly2graph` is a Python package for automatic *Hamiltonian spectral graph* construction. It takes in a characteristic polynomial or a Bloch Hamiltonian and returns the spectral graph.

F.1 Features

- High-performance
  - Fast construction of spectral graph from any one-dimensional models
  - Adaptive resolution to reduce floating operation cost and memory usage
  - Automatic backend for computation bottleneck. If `tensorflow` / `torch` is available, any device (e.g. `/GPU:0`, `/TPU:0`, `cuda:0`, etc.) that they support can be used for acceleration.

- Cover generic topological lattices
  - Support generic one-band and multi-band models
  - Flexible multiple input choices, be they characteristic polynomials or Bloch Hamiltonians; formats include strings, `sympy.Poly`, and `sympy.Matrix`
- Automatic and Robust
  - By default, no hyper-parameters are needed. Just input the characteristic of your model and `poly2graph` handles the rest
  - Automatic spectral boundary inference
  - Relatively robust on multiband models that are prone to "component fragmentation"
- Helper functionalities generally useful
  - `skeleton2graph` module: Convert a skeleton image to its graph representation
  - `hamiltonian` module: Conversion among different Hamiltonian representations and efficient computation of a range of properties

F.2 Installation

You can install the package via pip:

```
$ pip install poly2graph
```

or clone the repository and install it manually:

```
$ git clone https://github.com/sarinstein-yan/poly2graph.git
$ cd poly2graph
$ pip install .
```

Optionally, if TensorFlow or PyTorch is available, `poly2graph` will make use of them automatically to accelerate the computation bottleneck. Priority: `tensorflow` > `torch` > `numpy`.

This module is tested on `Python >= 3.11`. Check the installation:

```
import poly2graph as p2g
print(p2g.__version__)
```

F.3 Usage

See the Poly2Graph Tutorial JupyterNotebook for interactive examples.

`p2g.SpectralGraph` and `p2g.CharPolyClass` are the two main classes in the package.

`p2g.SpectralGraph` investigates the spectral graph topology of **a specific** given characteristic polynomial or Bloch Hamiltonian. `p2g.CharPolyClass` investigates **a class** of **parametrized** characteristic polynomials or Bloch Hamiltonians, and is optimized for generating spectral properties in parallel.

```
import numpy as np
import networkx as nx
import sympy as sp
import matplotlib.pyplot as plt
from matplotlib import colors
# always start by initializing the symbols for k, z, and E
```

```
k = sp.symbols('k', real=True)
z, E = sp.symbols('z E', complex=True)
```

F.3.1 A generic one-band example (p2g.SpectralGraph):

Characteristic polynomial:

$$P(E,z) := h(z) - E = z^4 - z - z^{-2} - E$$

Its Bloch Hamiltonian (Fourier transformed Hamiltonian in momentum space) is a scalar function:

$$h(z) = z^4 - z - z^{-2}$$

where the phase factor is defined as $z := e^{ik}$.

Expressed in terms of crystal momentum k:

$$h(k) = e^{4ik} - e^{ik} - e^{-2ik}$$

The valid input formats to initialize a `p2g.SpectralGraph` object are:

1. Characteristic polynomial in terms of `z` and `E`:
   - as a string of the Poly in terms of `z` and `E`
   - as a `sympy.Poly` with {`z`, `1/z`, `E`} as generators
2. Bloch Hamiltonian in terms of `k` or `z`
   - as a `sympy.Matrix` in terms of `k`
   - as a `sympy.Matrix` in terms of `z`

All the following `characteristics` are valid and will initialize to the same characteristic polynomial and therefore produce the same spectral graph:

```
char_poly_str = '-z**-2 - E - z + z**4'

char_poly_Poly = sp.Poly(
    -z**-2 - E - z + z**4,
    z, 1/z, E # generators are z, 1/z, E
)

phase_k = sp.exp(sp.I*k)
char_hamil_k = sp.Matrix([-phase_k**2 - phase_k + phase_k**4])

char_hamil_z = sp.Matrix([-z**-2 - E - z + z**4])
```

Let us just use the string to initialize and see a set of properties that are computed automatically:

```
sg = p2g.SpectralGraph(char_poly_str, k=k, z=z, E=E)
```

Characteristic polynomial:

```
sg.ChP
```

>>> Poly$\left(z^4 - z - \frac{1}{z^2} - E,\ z, \frac{1}{z}, E,\ domain = \mathbb{Z}\right)$

Bloch Hamiltonian:

- For one-band model, it is a unique, rank-0 matrix (scalar)

```
sg.h_k
```

>>>

$$\left[e^{4ik} - e^{ik} - e^{-2ik}\right]$$

```
sg.h_z
```

>>>

$$\left[-\frac{-z^6+z^3+1}{z^2}\right]$$

The Frobenius companion matrix of `P(E)(z)`:

- treating `E` as parameter and `z` as variable
- Its eigenvalues are the roots of the characteristic polynomial at a fixed complex energy `E`. Thus it is useful to calculate the GBZ (generalized Brillouin zone), the spectral potential (Ronkin function), etc.

```
sg.companion_E
```

>>>

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & E \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Number of bands & hopping range:

```
print('Number of bands:', sg.num_bands)
print('Max hopping length to the right:', sg.poly_p)
print('Max hopping length to the left:', sg.poly_q)
```

>>>

```
Number of bands: 1
Max hopping length to the right: 2
Max hopping length to the left: 4
```

A real-space Hamiltonian of a finite chain and its energy spectrum:

```python
H = sg.real_space_H(
    N=40,          # number of unit cells
    pbc=False,     # open boundary conditions
    max_dim=500  # maximum dimension of the Hamiltonian matrix (for numerical accuracy)
)

energy = np.linalg.eigvals(H)

fig, ax = plt.subplots(figsize=(3, 3))
ax.plot(energy.real, energy.imag, 'k.', markersize=5)
ax.set(xlabel='Re(E)', ylabel='Im(E)', \
xlim=sg.spectral_square[:2], ylim=sg.spectral_square[2:])
plt.tight_layout(); plt.show()
```

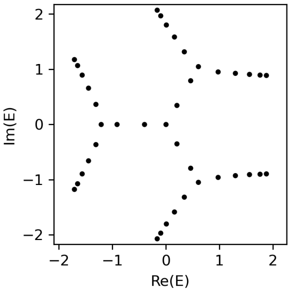

Figure A7: Finite spectrum one band

The Set of Spectral Functions

```python
phi, dos, binaried_dos = sg.spectral_images()

fig, axes = plt.subplots(1, 3, figsize=(8, 3), sharex=True, sharey=True)
axes[0].imshow(phi, extent=sg.spectral_square, cmap='terrain')
axes[0].set(xlabel='Re(E)', ylabel='Im(E)', title='Spectral Potential')
p2, p98 = np.percentile(dos, (2, 99))
# ^ Clip extreme DOS to increase visibility.
norm = colors.Normalize(vmin=p2, vmax=p98)
axes[1].imshow(dos, extent=sg.spectral_square, cmap='viridis', norm=norm)
axes[1].set(xlabel='Re(E)', title='Density of States')

axes[2].imshow(binaried_dos, extent=sg.spectral_square, cmap='gray')
axes[2].set(xlabel='Re(E)', title='Graph Skeleton')
```

```
plt.tight_layout()
plt.show()
```

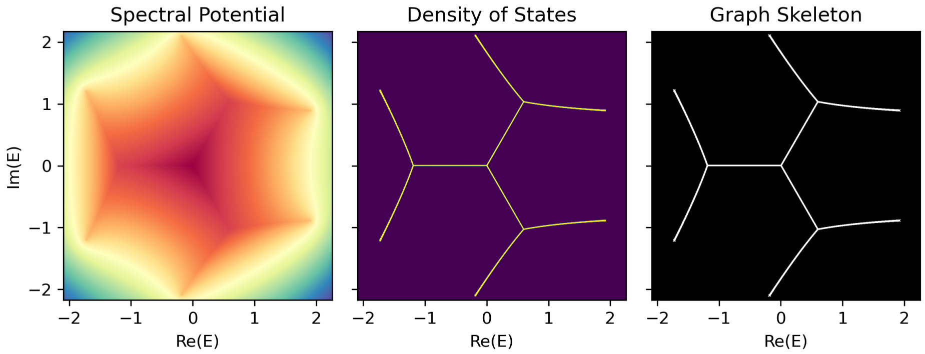

Figure A8: Spectral images one band

The spectral graph $\mathcal{G}$

```
graph = sg.spectral_graph()

fig, ax = plt.subplots(figsize=(3, 3))
pos = nx.get_node_attributes(graph, 'pos')
nx.draw_networkx_nodes(graph, pos, alpha=0.8, ax=ax,
            node_size=50, node_color='#A60628')
nx.draw_networkx_edges(graph, pos, alpha=0.8, ax=ax,
            width=5, edge_color='#348ABD')
plt.tight_layout(); plt.show()
```

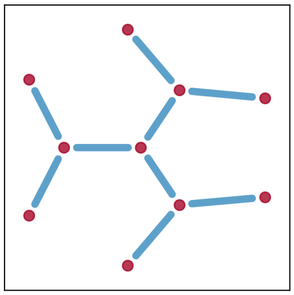

Figure A9: Spectral graph one band

If `tensorflow` or `torch` is available, `poly2graph` will automatically use them and run on **CPU** by default. If other device, e.g. GPU / TPU is available, one can pass `device = {device string}` to the method `spectral_images` and `spectral_graph`:

```python
SpectralGraph.spectral_images(device='/cpu:0')
SpectralGraph.spectral_graph(device='/gpu:1')
SpectralGraph.spectral_images(device='cpu')
SpectralGraph.spectral_graph(device='cuda:0')
...
```

However, some functions may not have gpu kernel in `tf`/`torch`, in which case the computation will fallback to CPU.

F.3.2 A generic multi-band example (`p2g.SpectralGraph`):

Characteristic polynomial (four bands):

$$P(E,z) := \det(\mathbf{h}(z) - E\,\mathbf{I}) = z^2 + 1/z^2 + Ez - E^4$$

One of its possible Bloch Hamiltonians in terms of z:

$$\mathbf{h}(z) = \begin{bmatrix} 0 & 0 & 0 & z^2+1/z^2 \\ 1 & 0 & 0 & z \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

```python
sg_multi = p2g.SpectralGraph("z**2 + 1/z**2 + E*z - E**4", k, z, E)
```

Characteristic polynomial:

```python
sg_multi.ChP
```

>>> $\mathrm{Poly}\left(z^2 + zE + \frac{1}{z^2} - E^4,\ z, \frac{1}{z}, E,\ domain = \mathbb{Z}\right)$

Bloch Hamiltonian:

- For multi-band model, if the `p2g.SpectralGraph` is not initialized with a `sympy Matrix`, then `poly2graph` will use the companion matrix of the characteristic polynomial `P(z)(E)` (treating `z` as parameter and `E` as variable) as the Bloch Hamiltonian – this is one of the set of possible band Hamiltonians that possesses the same energy spectrum and thus the same spectral graph.

```python
sg_multi.h_k
```

>>>

$$\begin{bmatrix} 0 & 0 & 0 & 2\cos(2k) \\ 1 & 0 & 0 & e^{ik} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

```
sg_multi.h_z
```

>>>

$$\begin{bmatrix} 0 & 0 & 0 & z^2 + \frac{1}{z^2} \\ 1 & 0 & 0 & z \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

The Frobenius companion matrix of `P(E)(z)`:

```
sg_multi.companion_E
```

>>>

$$\begin{bmatrix} 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & E^4 \\ 0 & 0 & 1 & -E \end{bmatrix}$$

Number of bands & hopping range:

```
print('Number of bands:', sg_multi.num_bands)
print('Max hopping length to the right:', sg_multi.poly_p)
print('Max hopping length to the left:', sg_multi.poly_q)
```

>>>

```
Number of bands: 4
Max hopping length to the right: 2
Max hopping length to the left: 2
```

A real-space Hamiltonian of a finite chain and its energy spectrum:

```
H_multi = sg_multi.real_space_H(
    N=40,         # number of unit cells
    pbc=False,    # open boundary conditions
    max_dim=500   # maximum dimension of the Hamiltonian matrix (for numerical
    ↪  accuracy)
)

energy_multi = np.linalg.eigvals(H_multi)

fig, ax = plt.subplots(figsize=(3, 3))
ax.plot(energy_multi.real, energy_multi.imag, 'k.', markersize=5)
ax.set(xlabel='Re(E)', ylabel='Im(E)', \
xlim=sg_multi.spectral_square[:2], ylim=sg_multi.spectral_square[2:])
plt.tight_layout(); plt.show()
```

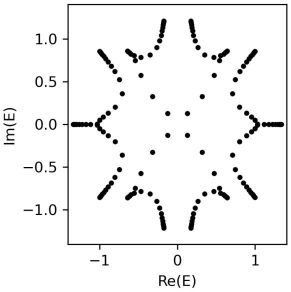

Figure A10: Finite spectrum multi band

The Set of Spectral Functions

```python
phi_multi, dos_multi, binaried_dos_multi = sg_multi.spectral_images(device='/cpu:0')

fig, axes = plt.subplots(1, 3, figsize=(8, 3), sharex=True, sharey=True)
axes[0].imshow(phi_multi, extent=sg_multi.spectral_square, cmap='terrain')
axes[0].set(xlabel='Re(E)', ylabel='Im(E)', title='Spectral Potential')
axes[1].imshow(dos_multi, extent=sg_multi.spectral_square, cmap='viridis', norm=norm)
axes[1].set(xlabel='Re(E)', title='Density of States')
axes[2].imshow(binaried_dos_multi, extent=sg_multi.spectral_square, cmap='gray')
axes[2].set(xlabel='Re(E)', title='Graph Skeleton')
plt.tight_layout(); plt.show()
```

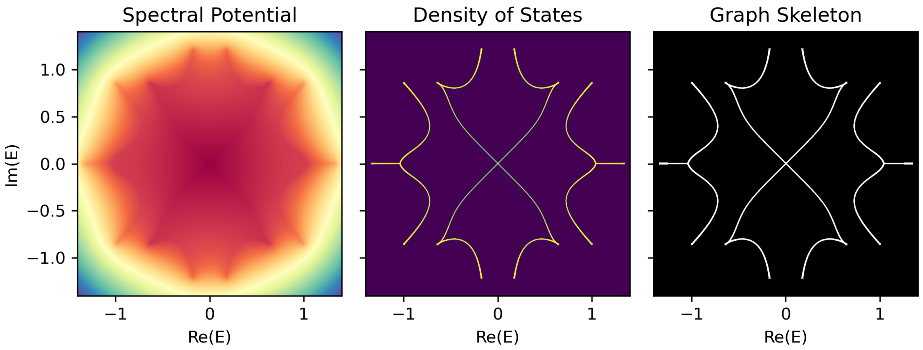

Figure A11: Spectral images multi band

The spectral graph $\mathcal{G}$

```python
graph_multi = sg_multi.spectral_graph(
    short_edge_threshold=20,
    # ^ node pairs or edges with distance < threshold pixels are merged
)

fig, ax = plt.subplots(figsize=(3, 3))
pos_multi = nx.get_node_attributes(graph_multi, 'pos')
```

```
nx.draw(graph_multi, pos_multi, ax=ax,
        node_size=10, node_color='#A60628',
        edge_color='#348ABD', width=2, alpha=0.8)
plt.tight_layout(); plt.show()
```

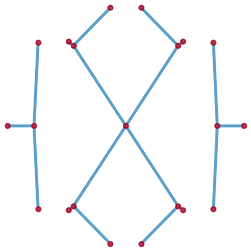

Figure A12: Spectral graph multi band

F.4 Node and Edge Attributes of the Spectral Graph Object

The spectral graph is a `networkx.MultiGraph` object.

- Node Attributes
  1. `pos` : (2,)-numpy array
     - the position of the node $(\mathrm{Re}(E), \mathrm{Im}(E))$
  2. `dos` : float
     - the density of states at the node
  3. `potential` : float
     - the spectral potential at the node
- Edge Attributes
  1. `weight` : float
     - the weight of the edge, which is the **length** of the edge in the complex energy plane
  2. `pts` : (w, 2)-numpy array
     - the positions of the points constituting the edge, where `w` is the number of points along the edge, i.e., the length of the edge, equals `weight`
  3. `avg_dos` : float
     - the average density of states along the edge
  4. `avg_potential` : float
     - the average spectral potential along the edge

```
node_attr = dict(graph.nodes(data=True))
edge_attr = list(graph.edges(data=True))
print('The attributes of the first node\n', node_attr[0], '\n')
print('The attributes of the first edge\n', edge_attr[0][-1], '\n')
```

>>>

```
The attributes of the first node
 {'pos': array([-0.20403848, -2.11668106]),
  'dos': 0.0011466597206890583,
  'potential': -0.655870258808136}

The attributes of the first edge
 {'weight': 1.4176547247784077,
  'pts': array([[-2.04038482e-01, -2.11668106e+00],
       [-1.99792382e-01, -2.11243496e+00],
       ...
       [ 5.94228396e-01, -1.02967935e+00]]),
  'avg_dos': 0.10761458,
  'avg_potential': -0.5068641}
```

F.4.1 A generic multi-band class (p2g.CharPolyClass):

Let us add two parameters {a,b} to the aforementioned multi-band example and construct a p2g.CharPolyClass object:

```
a, b = sp.symbols('a b', real=True)

cp = p2g.CharPolyClass(
    "z**2 + a/z**2 + b*E*z - E**4",
    k=k, z=z, E=E,
    params={a, b}, # pass parameters as a set
)
```

> > >

```
Derived Bloch Hamiltonian `h_z` with 4 bands.
```

View a few auto-computed properties

Characteristic polynomial:

```
cp.ChP
```

> > > $\operatorname{Poly}\left(z^2 + a\frac{1}{z^2} + bzE - E^4, z, \frac{1}{z}, E, domain = \mathbb{Z}\left[a, b\right]\right)$

Bloch Hamiltonian:

```
cp.h_k
```

> > >

$$\begin{bmatrix} 0 & 0 & 0 & \left(a + e^{4ik}\right)e^{-2ik} \\ 1 & 0 & 0 & be^{ik} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

```
cp.h_z
```

>>>

$$\begin{bmatrix}0 & 0 & 0 & \frac{a}{z^2} + z^2 \\ 1 & 0 & 0 & bz \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0\end{bmatrix}$$

The Frobenius companion matrix of `P(E)(z)`:

```
cp.companion_E
```

>>>

$$\begin{bmatrix}0 & 0 & 0 & -a \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & E^4 \\ 0 & 0 & 1 & -Eb\end{bmatrix}$$

An Array of Spectral Functions

To get an array of spectral images or spectral graphs, we first prepare the values of the parameters `{a,b}`

```
a_array = np.linspace(-2, 1, 6)
b_array = np.linspace(-1, 1, 6)
a_grid, b_grid = np.meshgrid(a_array, b_array)
param_dict = {a: a_grid, b: b_grid}
print('a_grid shape:', a_grid.shape,
    '\nb_grid shape:', b_grid.shape)
```

>>>

```
a_grid shape: (6, 6)
b_grid shape: (6, 6)
```

Note that **the value array of the parameters should have the same shape**, which is also **the shape of the output array of spectral images**

```
phi_arr, dos_arr, binaried_dos_arr, spectral_square = \
    cp.spectral_images(param_dict=param_dict)
print('phi_arr shape:', phi_arr.shape,
    '\ndos_arr shape:', dos_arr.shape,
    '\nbinaried_dos_arr shape:', binaried_dos_arr.shape)
```

>>>

```
phi_arr shape: (6, 6, 1024, 1024)
dos_arr shape: (6, 6, 1024, 1024)
binaried_dos_arr shape: (6, 6, 1024, 1024)
```

```python
from mpl_toolkits.axes_grid1 import ImageGrid

fig = plt.figure(figsize=(13, 13))
grid = ImageGrid(fig, 111, nrows_ncols=(6, 6), axes_pad=0,
                 label_mode='L', share_all=True)

for ax, (i, j) in zip(grid, [(i, j) for i in range(6) for j in range(6)]):
    ax.imshow(phi_arr[i, j], extent=spectral_square[i, j], cmap='terrain')
    ax.set(xlabel='Re(E)', ylabel='Im(E)')
    ax.text(
        0.03, 0.97, f'a = {a_array[i]:.2f}, b = {b_array[j]:.2f}',
        ha='left', va='top', transform=ax.transAxes,
        fontsize=10, color='tab:red',
        bbox=dict(alpha=0.8, facecolor='white')
    )

plt.tight_layout()
plt.savefig('./assets/ChP_spectral_potential_grid.png', dpi=72)
plt.show()
```

An Array of Spectral Graphs

```python
graph_flat, param_dict_flat = cp.spectral_graph(param_dict=param_dict)
print(graph_flat, '\n')
print(param_dict_flat)
```

```
[<networkx.classes.multigraph.MultiGraph object at 0x000001966DFCD190>,
<networkx.classes.multigraph.MultiGraph object at 0x000001966DFCECF0>,
...
<networkx.classes.multigraph.MultiGraph object at 0x000001966DFCE750>]

{a:
array([-2. , -1.4, -0.8, -0.2,  0.4,  1. , -2. , -1.4, -0.8, -0.2,  0.4,
        1. , -2. , -1.4, -0.8, -0.2,  0.4,  1. , -2. , -1.4, -0.8, -0.2,
        0.4,  1. , -2. , -1.4, -0.8, -0.2,  0.4,  1. , -2. , -1.4, -0.8,
       -0.2,  0.4,  1. ]),
b:
array([-1. , -1. , -1. , -1. , -1. , -1. , -0.6, -0.6, -0.6, -0.6, -0.6,
       -0.6, -0.2, -0.2, -0.2, -0.2, -0.2, -0.2,  0.2,  0.2,  0.2,  0.2,
        0.2,  0.2,  0.6,  0.6,  0.6,  0.6,  0.6,  0.6,  1. ,  1. ,  1. ,
        1. ,  1. ,  1. ])}
```

The spectral graph is a `networkx.MultiGraph` object, which cannot be directly returned as a multi-dimensional numpy array of `MultiGraph`, except for the case of 1D array. Instead, we return a flattened list of `networkx.MultiGraph` objects, and the accompanying `param_dict_flat` is the dictionary that contains the corresponding flattened parameter values.

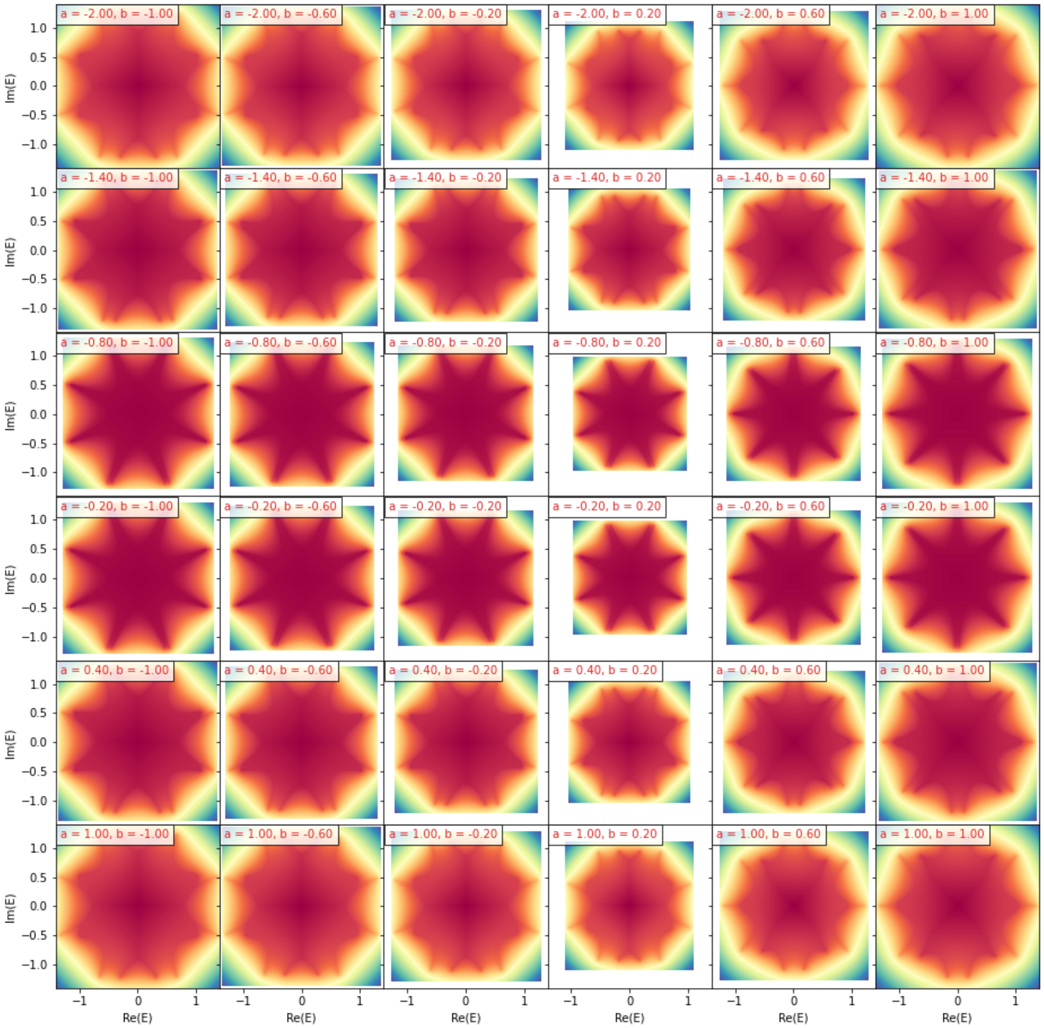

Figure A13: ChP spectral potential grid

It's recommended to pass the values of the parameters as `vectors` (1D arrays) instead of higher dimensional `ND arrays` to avoid the overhead of reshaping the output and the difficulty to retrieve / postprocess the spectral graphs.