Crystal Hypergraph Convolutional Networks

Alexander J. Heilman[1], Weiyi Gong[1], and Qimin Yan[1*]

[1]*Department of Physics, Northeastern University, Boston, MA 02115, USA*

Abstract

Graph representations of solid state materials that encode only interatomic-distance information lack geometrical resolution, resulting in degenerate representations that may map distinct structures to equivalent graphs. Here we propose a hypergraph representation scheme for materials that allows for the association of higher-order geometrical information with hyperedges. Hyperedges generalize edges to connected sets of more than two nodes, and may be used to represent triplets and local environments of atoms in materials. This generalization of edges requires a different approach in graph convolution, which is developed in this work. These crystal hypergraph convolutional networks are trained based on various property prediction tasks for a vast set of solid-state materials available via MatBench. Results presented here focus on the improved performance of models based on both pair-wise edges and local environment hyperedges. These results demonstrate that hypergraphs are an effective and efficient method for incorporating geometrical information in material representations.

Introduction

Machine learning has proven to be a computationally cost-effective and powerful predictive tool in the screening of large sets of material systems for certain material properties [1–4]. Some of the most effective state-of-the-art models applied to invariant target predictions represent material systems as graphs [5–10]. These graphs encode physical properties in feature vectors associated with graph components, and update or 'learn' these features with a trained graph neural network or message passing network [11].

One problem with such graphical representations, however, is the lack of representation of higher-order geometrical structure, since the constructed crystal graphs can only include pair-wise descriptors. This may make it difficult or even impossible for models to distinguish between compositionally similar but structurally distinct systems with unique material properties [12]. Some other works have approached this problem by including higher-order geometrical features such as overlapping bonds' angles [12–14]. However, these approaches come with an increase in the total number of messages exchanged through convolution that is quadratic with respect to $\bar{N}_{\text{edges}}$, the average number of edges per atom. Yet other approaches reduce the complexity of the representation. In particular, molecular graph coarsening models [15–18] aim to be more efficient in their descriptors, and hence representations, by coarsening the graphs representing molecular systems through convolution and aggregation, aiming to keep only the most important groupings. Another common approach is to directly incorporate the directional vectors of edge features and maintain equivariance through convolution [19–23], though this requires substantially different convolutional architectures than those adopted in invariant networks, often at a higher computational cost though with the additional ability to directly predict coordinate-system dependent quantities.

Here, we propose the concept of *crystal hypergraphs* to address this lack of geometrical information in the more restrictive graph representations. In a crystal hypergraph, we may define larger (than strictly pair-wise) hyperedges that correspond to higher-order geometrical structures of material systems explicitly, such as triplets of neighboring atoms, or coordination polyhedra/motifs [24–29]. These different structures then may have different coordinate invariant features associated with them, such as angles and local order parameters [30–32], respectively. Note that in that regard, crystal hypergraphs are naturally heterogeneous in their hyperedges, since there are different feature sets for different types of hyperedges.

*Corresponding Author: q.yan@northeastern.edu

Of course, the definition of a more general hypergraph representation requires the generalization of the message passing framework mentioned above. Here, we propose three possible approaches to such a generalization that handle the now-variable size of hyperedges. In a certain sense, these allow for the learning of a certain type of 'cluster-correlation expansion' (CCE) [33–36] by the model, where clusters of interest correspond to the hyperedges defined. Indeed, atomic cluster expansions have previously been utilized to great effect in the generation and application of transferable interatomic potentials [37–39].

In many-body expansions, such as CCE, expansions are often truncated to exclude higher-order terms since their associated parameters are typically observed to decay quickly with increasing distance from the central atom. Similarly, in multi-pole expansions, higher-order terms (such as quadrupole, octupole, and beyond) often contribute considerably less to long-range field effects. The intuition gained from these considerations further support the use of only a localized set of descriptors for atom-centric models. This intuition is used in the present work to motivate the use of neighborhood aggregation for hyperedge message passing, wherein only atomic features local to hyperedges are incorporated in their associated messages.

As a proof of concept, we propose and implement a crystal hypergraph convolutional model (CHGCNN) that incorporates invariant geometric features for bonds, triplets, and motifs in crystals as hyperedge features. This provides us a unique opportunity to demonstrate the importance of different order structures for these different material property prediction tasks. Namely, we compare the performance of models based on atom, bond, and triplet information against those incorporating atom, bond, and motif information (i.e. first shell hyperedges) on various predictive tasks with varying data sizes.

Results presented here indicate that first-shell (motif) hyperedges may be sufficient, if not more informative, than triplet hyperedges for many common predictive tasks. This comes at a substantially lower computational cost, in terms of the total number of messages exchanged through graph convolution.

The structure of this work is as follows. First, the concept of crystal hypergraphs is introduced, with a particular focus on different types of hyperedges and their corresponding feature sets. Three generalized message passing frameworks are then considered, and a specific model architecture is presented. Finally, this specific architecture is used on various datasets to evaluate the performance of different sets of hyperedge types.

Results

Crystal Hypergraph Construction

Hypergraphs are a very general framework describing relations between an abstract set $\mathcal{V}$ of nodes or vertices, defined by a set of hyperedges $\mathcal{H}$ containing arbitrary subsets of $\mathcal{V}$. Consequently, the set of all hypergraphs on a set of nodes is more general than, and contains all possible, topological sets and simplicial complexes on $\mathcal{V}$ [40]. The method proposed here reduces the intrinsic limitations of invariant crystal graph features by allowing the explicit incorporation of higher-order geometrical information in the form of hyperedges, which can be used to directly represent these higher-order structures. Furthermore, the more general definition of hypergraphs allows for complete freedom in the choice of structures to be included, as opposed to the more rigid definition and utilization of representations in simplicial-complex networks [41–43].

A crystal hypergraph $\mathcal{H} = \{\mathcal{V}, H\}$ is a collection of nodes $v_i \in \mathcal{V}$ and hyperedges $h_j \in H$ (containing an arbitrary number of nodes, see Supplementary Note S5), where the hyperedges are most generally heterogeneous. That is, we generally wish to describe different types of hyperedges (e.g. bonds, triplets, and motifs) in the same crystal hypergraph. These objects then have associated feature vectors encoding relevant physical information, which we also refer to as v_i and h_j, since the indices specify all the relevant information for their association to particular nodes and hyperedges, respectively.

For the purpose of modeling material systems, we need to identify what different order structures are most important in their representation. Of course, atomic and bond level information is particularly important. However, typical crystal graph construction including only atoms and bonds (See the Methods section) lacks higher-order geometrical information of crystals. That is, distinct local geometrical environments of atoms (motifs) may be mapped degenerately to the same crystal graph. As a simple example of the low resolution manifest in crystal graphs containing only bonds, consider the two atomic systems in Figure 1: one with a local cubic symmetry, and another with a square anti-prism local envi-

ronment; but both with the same bonding atoms. As demonstrated in Figure 1, both structures would map to the same crystal graph, but could be easily distinguished with an additional descriptor describing the local geometry of each central atom. As such, higher-order structures may also be of interest, such as triplets of atoms and local environments of atoms, which we refer to here, generally, as motifs.

Each of the aforementioned structures also has a natural set of distinct, coordinate-system invariant features that may be associated with them. At the triplet level (where two bonds share some common node), there is always a corresponding angle. While at the motif level, order parameters [30, 31] or continuous symmetry measures [44, 45] may be used to describe 3-dimensional coordination environments quantitatively. These different order structures may all be represented in a single crystal hypergraph. Below, we discuss the generation of, and association of features with, all of the above-mentioned structures in crystalline solids. The scaling of the number of messages for each hyperedge type is demonstrated for an example material in Supplementary Note S6.

Bonds, or pair-wise atomic connections, are determined in the same manner as in a crystal graph. A commonly applied criterion for the formation of edges between atoms is a combination of a maximum distance cutoff r_{max} and a maximum number of neighbors for each node N_{max}. That is, for each atom, edges are constructed between its node and its $\leq N_{max}$-th closest neighbors in the crystalline structure within a shell of radius r_{max}..

Triplet hyperedges are then formed from the set of bonds. For each set of bonds connected by one node, a triplet hyperedge is formed. The feature of these triplet hyperedges is also a Gaussian expansion, though now of the angle formed by the unit vectors of the two bonds [13]. Triplet hyperedges give us a way to incorporate some angular resolution into our representation scheme in a coordinate-system-invariant manner. For a node with N bonds then, there will be $N(N-1)/2$ triplets. Thus, the price we pay for complete angular resolution of any two bonds is a quadratic increase in the number of hyperedges. For a comparison between our inclusion of triplet information and the more usual construction by way of a line-graph, see Supplementary Note S4.

Motif determination may be achieved by a wide range of functions, and is akin to an algorithmic determination of coordination number [46]. Possible features for these motifs then include: Zimmermann's 35 local structure order parameters (LSOPs) [30, 31]; continuous symmetry measures [44] (CSMs, e.g. 'distance to a perfect shape') for 59 common coordination environments; or a combination of both (see Supplementary Note S1 for more detail). In essence, both are just sets of quantitative measures designed to describe 3 dimensional physical shape. Motif hyperedges thus give us a way to describe the local geometry of sites in material systems with much fewer hyperedges. Since each node will contribute one motif hyperedge, for a crystal with N_{nodes} nodes, we just have N_{nodes} motifs. A schematic depicting the generation of these crystal hypergraphs is given in Figure 2.

Crystal Hypergraph Convolution

We now must consider a message passing framework analogous to *Gilmore, et al* [11] but applying to hypergraph structures. That is, we now have:

$$
\begin{aligned}
m_v^{t+1} &= \sum_{h_j \in \mathcal{N}(v)} M_t(n_v^t, h_j^t, \{n_w^t | n_w \in h_j\}), \\
n_v^{t+1} &= U_t(n_v^t, m_v^{t+1}), \\
\hat{y} &= R(\{n_v^T\}),
\end{aligned}
$$

so that each node is still updated according to some layer-wise update function U_t, aggregating messages m^{t+1} formed from origin node features, hyperedge features h_j, and hyperedge neighborhood features $n_w \in h_j$. This update occurs node-wise and then after T layers, some readout function R is used to output the corresponding predicted value $\hat{y}$, which utilizes the set of learned node features.

The biggest difference here is that we now need a message forming function M_t that accounts for a set of node features $\{n_w^t | n_w \in h_j\}$ which may vary in size between different hyperedges (even of the same type). This stands in opposition to the case of regular edges, where we are assured a fixed size of two nodes per edge.

One approach would be to fix the dimensionality of each type of hyperedge, or have a different convolutional operator for each different size hyperedge (as is effectively the approach taken with line graph networks [47]). Here, however, we wish to maintain generality in edge size so we need not fix

hyperedge sizes for each hyperedge type, since structures of different sizes may be described by similar metrics. For example, we may wish that motifs resembling polyhedra with different numbers of vertices are described by common sets of features.

Of course, there should be different message and update functions for each different order structure (bonds, triplets, motifs, etc.) with different features. This is accounted for by treating the data as a heterogeneous graph, with different hyperedge types. Below, we consider three strategies that allow us to apply our convolutional operator to hyperedges of arbitrary size.

Three general approaches for message passing that account for this multi-order nature have been considered in this work: **1.** the construction of a hyperedge relatives graph, upon which regular graph convolution may be applied; **2.** total exchange hyperedge message convolution, which completely generalizes the CGCNN [6] and ALIGNN [13] models to hypergraphs; and **3.** neighborhood aggregation, which balances performance of the former approach by forming a single neighborhood feature for each hyperedge. These methods are depicted in Figure 3.

Each approach has a different computational cost in terms of the total number of messages, along with a potentially different practical definition of a hypergraph. These considerations are presented below, with a specific convolutional structure and empirical results on common test datasets presented after.

Hyperedge Relatives Graph

We may define a dual graph $\mathcal{D}(h)$ to a hypergraph h to be a graph in which nodes represent the hyperedges of the hypergraph, and connections represent the overlap of respective hyperedge neighborhoods. In the case of a crystal hypergraph with heterogeneous hyperedges, this dual graph is a graph with heterogeneous nodes. We term this heterogeneous dual graph of a crystal hypergraph the relatives graph for simplicity. Atomic features may be included in this framework by adding a singleton hyperedge for each node.

The definition of the relatives graph allows us to perform the usual methods of graph convolution on hyperedge features. Such an approach also allows us to define our relatives graph as we would a graph, with just a standard edge index.

However, this approach lacks the interaction of neighboring features in convolution via the connecting hyperedge. That is, without a clear definition of the edge attribute, messages are generally of the form below:

$$m_v^t = \sum_{h_j \in \mathcal{N}(v)} M_t(n_v^t, h_j^t)$$

in which we simply discard the neighborhood of other node features contained in the hyperedge.

Computationally, this approach has a total number of messages that scales linearly with average hyperedge size, since each hyperedge only contributes one message to each node it contains. Accounting only for node-hyperedge connections in a relatives graph derived from a hypergraph with m hyperedges of average order n, the total number of messages per convolution will scale as $\mathcal{O}(nm)$.

Total Exchange Message Passing

Of course, we may wish to incorporate the neighboring features of some representation via their connecting hyperedge. This may be accomplished by simply forming a message for every pair of connected representations along with their connecting hyperedge's representation.

$$m_v^t = \sum_{h_j \in \mathcal{N}(v)} \sum_{n_w \in h_j} M_t(n_v^t, h_j^t, n_w^t),$$

Here, though, we have introduced a new summation which may drastically increase the number of messages for larger hyperedges. In this scheme, if each hyperedge contains an average of n nodes and there are m hyperedges, the total number of messages exchanged per node-wise convolution will scale as $\mathcal{O}(n^2 m)$.

Neighborhood Aggregation

Since the number of messages will scale tremendously with larger hyperedges in the framework described above, we may seek a way to incorporate the neighborhood of contained node features of a hyperedge

into a single message. In this case, we may form a single 'neighborhood feature' representative of all contained nodes of a hyperedge by way of an aggregation function. This aggregation function takes as input a variable number of node features and returns a single feature of fixed dimension. Here then, we use message functions of the form:

$$m_v^{t+1} = \sum_{h_j \in \mathcal{N}(v)} M_t(n_v^t, h_j^t, \mathrm{AGG}(\{n_w^t | n_w \in h_j\})$$

where AGG denotes some order-invariant aggregation function on a set of node features, such as component-wise minimum, maximum, average, or a combination of these with learnable weights. The use of an aggregation function here thus allows for the passing of one message per hyperedge, while accounting for the entire environment of local node features in the hyperedge. It achieves this by transforming the variable set of contained node features to a fixed size output, such that we may use the same message function across sets of hyperedges of differing orders. This results in a set of node-wise messages that scales linearly with the average size of hyperedges, so that we now have a relationship of order $\mathcal{O}(nm)$ again.

MatBench Results

Crystal hypergraph networks provide a unique opportunity to investigate the importance of different order structures in the prediction of various material properties. Specifically, we may compare performance between models based on different types of hyperedges to probe the relevance of certain structures (e.g. motifs vs. triplets) in material property prediction. From the different hyperedge types considered here, we build three different models based on the architecture given in Figure 4. We base our graph model on the basic bond-only graph network CGCNN [6], and compare its performance against two models incorporating two additional types of hyperedges: bond-and-triplet and bond-and-motif models. Note that these additional hyperedge convolutional layers are incorporated directly into CGCNN's publicly available code for fair comparison. For compound models (including more than one hyperedge type) each CHGConv layer performs convolution over the hyperedges in ascending order of hyperedge size. These models were each trained on sets of training data from MatBench (v0.1) [48]. Details on the hyperparameters and training protocol can be found in Supplementary Note S3. Note that models using both motif and triplet-level edges, in general, diverged through training or did not perform any better than models using just motifs or triplets. As such, we only compare models using one or the other here. For all tests, datasets were split into 80% train / 20% test subsets across 5 folds. The training subsets were then further split into 90% train / 10% validation sets, where the model with the best performance on the validation set (withheld through training) was then used to predict the values of the final test sets. Note that MatBench provides fixed fold splits for benchmarking purposes. Performance on each fold is reported in Supplementary Note S9, along with the calculated uncertainty for each model applied to each task.

We first focus on the comparative performance of models incorporating only bonds (CGCNN), models incorporating both bond-and-motif hyperedges and models with both bond-and-triplet hyperedges on five smaller MatBench (v0.1) target datasets [48]. Mean absolute error (MAE) on test sets for these regression tasks are reported in Table 1. These five datasets consist of the following targets: the highest frequency phonon peak ω_p for 1,265 materials, refractive indices n for 4,764 materials, formation energies E_f for a set of 18,928 perovskite materials, and 10,987 bulk and shear moduli, K_{vrh} and G_{vrh}, respectively. On all tasks of these smaller sets, the larger models incorporating both bonds and higher-order hyperedges performed best, with the motif-based models performing best overall. For refractive index prediction, the motif-based models had an average MAE of 0.432 across the test sets compared to an MAE of 0.440 for the bond-and-triplet model, with both showing improvement over the bond-only model with an MAE of 0.599. Perovskite formation energy prediction results were very close for the three models, while the bond-and-motif model again had the best performance with an MAE of 42.4 meV/atom, the bond-and-triplet model was close behind at an MAE of 42.7 meV/atom, both showing only a slight improvement over the bond-only model's performance with an MAE of 45.2 eV/atom. However, since perovskites are a class of materials with a relatively standard structure, it should be unsurprising that the inclusion of additional structural information has little to no effect on performance. In the prediction of highest frequency phonon peak, the bond-and-motif model had the best performance with an average MAE of

52.9 cm^{-1} across test sets compared to 60.7 cm^{-1} for the triplet based model and 57.8 cm^{-1} for the bond-only model. For elastic targets, motif based models boasted average MAEs of 0.0805 Log_{10}(GPa) and 0.0635 Log_{10}(GPa) on the test sets of bulk moduli G_{vrh} and shear moduli K_{vrh}, respectively. This is compared to the performance of the bond-and-triplet models with MAEs of 0.0846 Log_{10}(GPa) and 0.0666 Log_{10}(GPa) on G_{vrh} and K_{vrh}, both being an improvement over the bond-only models with MAEs of 0.0895 Log_{10}(GPa) and 0.0712 Log_{10}(GPa). This may be indicative of the importance of such information in the relation of stress to infinitesimal strain, since the local environments of atoms would be of particular importance in considerations of both bulk (K_{vrh}) and shear response (G_{vrh}).

To see if the improved performance of motif-based models generalizes to larger data sets, we now compare the performance of bond and bond-and-motif models on three more target properties for a much larger sets of materials from the Materials Project database [49], again provided by MatBench (v0.1). Targets of these sets include: formation energy E_f for 132,752 materials; and band gap E_g and metallicity for 106,113 materials. Results for this set of tests are reported in Table 2 with MAE reported for regression tasks and area under curve (AUC) for classification tasks. For the Materials Project datasets, the bond-and-motif model performed best on the band gap and metallicity target sets, with the bond-only model then performing best on the formation energy task. In the prediction of band gap, the bond-and-motif model performed better than that without motifs, with an MAE of 0.23 eV vs. an MAE of 0.30 eV for the bond-only model. This trend also held for the metal/nonmetal classification task, with the best performance on the test set again by the bond-and-motif model with an AUC of 0.958, compared to an AUC of 0.952 for the bond-only model. In the prediction of formation energy, the bond-and-motif model performed worse on the test sets overall with an MAE of 39.7 eV/atom, compared to an MAE of 33.7 eV/atom for the bond-only model. This lack of improvement suggests that bond-level features may be adequate for formation energy tasks.

Discussion

State-of-the-art GNN models applied to material property prediction often represent material systems as graphs with relatively low geometrical resolution. This low resolution is often increased by associating bond angles with auxiliary line graphs derived from the graph itself. The primary argument of our work is that hypergraphs are a more natural representation of material structures that allow us to explicitly incorporate geometrical information with different substructures of choice in one unified representation. The results suggest that such an approach allows for a substantial decrease in computational cost compared to line-graph or triplet methods, by incorporating such geometrical information with single local environment hyperedges for each node as opposed to triplets of atoms for each pair of overlapping bonds, the number of which scales with the average number of bonds per atom N as $N(N-1)/2$. This is shown within one unified framework to have comparable performance on a number of common predictive tasks. In principle, these motif-level hyperedges may be added to any existing atomistic graph neural network. The results in this work further suggest that these additional hyperedges may improve performance for these models across many supervised predictive tasks, with a relatively small computational cost since the motif-level hyperedges contribute substantially less messages per convolutional layer compared to the number of bonds.

A similar observation was made in the AMDNet architecture [50], where motif information (included via an additional 'motif graph' for each material) also improved performance on most tasks, but here we compare results directly to the inclusion of bond angles via triplets. Our results indicate that one local neighborhood feature per atom may be sufficient to describe the local geometries of atoms for many predictive tasks, as opposed to the more data-intensive triplet representation scheme usually employed (often by way of line graphs). This ability of the motif-based models to more effectively distinguish compositionally similar but structurally distinct materials is further emphasized with an example in Supplementary Note S7. Taking this as a learned guiding principle, future crystal representations may benefit from reduced size while being assured similar geometric resolution. However, it should be noted that the greatest improvements seen here were on smaller datasets, suggesting the inductive bias of the motif-level hyperedges yields diminishing returns with larger datasets. This suggests that with large enough training data sets, continuous convolution filters for hyperedges encoding distance or angle alone may be sufficient for larger targets. However, given the intrinsic difficulty of the production of most

target data in materials science, the specific methods presented here may still prove beneficial in many applications.

Future works may investigate more powerful hypergraph convolutional operators that automatically detect motifs [27, 29, 51]; or apply this framework to molecular systems [52, 53] with functional groups. Inter-order convolution may also be of interest for certain tasks, where different hyperedge types may update each other's representations as opposed to just atom representations. Note that inter-order convolution would allow for a complete generalization of previous line-graph convolution schemes, where triplets effectively update their respective bonds' representations through convolution, as in [12, 13]. Other order structures (beyond motif-level) may also be of interest, such as hyperedges representing defect complexes or entire unit cells. Equivariant features and convolution [20, 22, 54] may also be incorporated for the prediction of coordinate-system dependent properties of materials from hypergraph representations, with the present work being focused on coordinate-system invariant features and targets.

Methods

Crystal Graphs

A common representation of crystalline systems in machine learning is via graphs (that is, collections of nodes and binary connections between them). We may define a crystal graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ as a set of vertices $v_i \in \mathcal{V}$, corresponding to each atom i, and edges $e_{ij} \in \mathcal{E}$, where edges are determined by some physical criteria. Physical information is then associated with the objects in these graphs by way of feature vectors. These are vectors with components describing the physical characteristics of their corresponding graph component, and which may be further 'learned' or updated through a graph neural network.

The nodes' feature vectors encode the atomic information of the sites they describe. Two usual techniques include: explicitly engineered feature vectors (as in [6]); and the learning of encodings for atomic sites based only on their atomic number (as in [7]), beginning with some random initialization. Edge features are often derived exclusively from their distance.

Crystal graphs are usually constructed solely for use in graph convolutional neural networks. Perhaps the most general framework in which we may define graph convolution is the message passing framework defined by Gilmore [11]. A message passing network updates nodes based on 'messages' generated by the features of, and passed through, neighboring nodes (that is, nodes sharing an edge).

Alternatively, one could include atomic position in the node features or a vector direction in edge features. However, this generally requires a unique treatment of such coordinate-system dependent information through convolution if the output is to maintain *invariance* with respect to changes in the coordinate system. As such, often only coordinate system invariant features are included in crystal graph representations, such as distance and atomic properties.

Crystal Hypergraph Construction

Initial atomic features were those used in CGCNN [6], consisting of a concatenated set of 92 one-hot encoded atomic properties. Bonds were formed for a maximum number of neighbors $N_{max} = 12$ found within a shell of radius $r_{max} = 8\mathring{A}$. For bond features, a Gaussian expansion of interatomic distance of dimension 40 ranging from 0 to 8 Åwas used, mainly to align with our bond-only benchmark CGCNN [6]. Triplet features were a Gaussian expansion of the cosine of bond angle, also of dimension 40. Motif features were a concatenation of 59 continuous symmetry measures, based on a preliminary test that indicates including both LSOPs and CSMs may be redundant (see Supplementary Note S8). Here, we use a modified Voronoi algorithm with a cut-off radius implemented as CrystalNN in pymatgen. Note this is a much stricter algorithm than that used to determine edges and triplets, since the motifs features depend heavily on the selected first-shell.

Model Architecture

In the model considered in this work, initial node and hyperedge features were first passed into a linear embedding layer (with no activation function). These embedded features were then fed into a set of Crystal HyperGraph Convolutional (CHGConv) layers which utilize the neighborhood aggregation

method presented above (see Supplementary Note S2). In CHGConv, we use a set of CGConv [6] layers applied to consecutively larger hyperedge types, taking as input the origin node of the hyperedge, the connecting hyperedge feature as the edge feature, and an aggregated set of neighborhood features as the connected node feature. Inspired by *ChemGNN* [8], the AGG function chosen for neighborhood aggregation through message passing on hyperedges was a combined set of component-wise maximum, minimum, average, and standard deviation, all with learnable attention weights. This neighborhood feature is then projected down to the hidden node dimension. In this manner, for every CHGConv layer, the atoms are updated by each hyperedge type chosen once (see Fig. 4). Note that each CGConv for different hyperedge types have independent trainable parameters.

These learned node features are then mean pooled to form a crystal vector, which is passed to a fully connected layer and then projected down to a one-dimensional (scalar) output for regression. In the case of classification tasks, the fully connected layer, after mean pooling, utilized a dropout mechanism and output a probability distribution of classes by way of a softmax activation function.

Hyperparameters for Testing

For each convolutional structure, testing was done for a model with 3 convolutional layers. Each convolutional layer consists of back-to-back convolution from the smallest to the largest hyperedge type (for example two bond & motif layers consist of a total sequence of bond, motif, bond and motif convolution).

Stochastic gradient descent (SGD) was used as an optimizer through training with an initial learning rate of 0.01. A multi-step learning rate scheduler divided this learning rate by a factor of 10 at epoch 150, with training running for a total of 300 epochs.

Hidden node features were of dimension 64 through all convolutional layers, and a hidden output layer of dimension 128 was used (similar to CGCNN's architecture). The loss functions utilized were Mean Squared Error (MSE, for regression tasks) and cross entropy (for classification tasks). Accuracy is then reported in Mean Absolute Error (MAE) for regression tasks and area under curve (AUC) for classification tasks. For a table summarizing these hyperparameters, see Supplementary Note S3.

Results reported were averaged over the 5 pre-fixed MatBench [48] folds for nested cross-validation. These datasets are divided into 80% for training and 20% for test for each fold, with a further 10% of the training subset being withheld from training and used as an indicative validation set, where the best performance on this dataset was used to select the model applied to the test set.

Data Availability

The processed data including bond, triplet, and motif features is available in a Zenodo repository with DOI: 10.5281/zenodo.14756640.

Code Availability

The code used in this paper's results were built on pytorch-geometric and can be found in the following Github repository: https://github.com/qmatyanlab/CHGCNN.

Acknowledgments

This work is supported by the U.S. Department of Energy, Office of Science, Basic Energy Sciences, under Award No. DE-SC0023664. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231 using NERSC award BES-ERCAP0029544.

Author Contributions

A.H. and Q.Y. conceived the research and wrote the manuscript. W.G. provided partial software support and was involved in discussions. Q.Y. supervised the project.

Competing Interests

There are no conflicts to declare.

References

[1] L. Ward, A. Agrawal, A. Choudhary, and C. Wolverton, *npj Comput. Mater.* **2**, 1–7 (2016).

[2] J. Wei, X. Chu, X.-Y. Sun, K. Xu, H.-X. Deng, J. Chen, Z. Wei, and M. Lei, *InfoMat* **1**, 338–358 (2019).

[3] Y. Liu, C. Niu, Z. Wang, Y. Gan, Y. Zhu, S. Sun, and T. Shen, *J. Mater. Sci. Tech.* **57**, 113–122 (2020).

[4] M. H. Mobarak, M. A. Mimona, M. A. Islam, N. Hossain, F. T. Zohura, I. Imtiaz, and M. I. H. Rimon, *Appl. Surf. Sci. Adv.* **18**, 100523 (2023).

[5] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, *J. Chem. Phys.* **148**, 241722 (2018).

[6] T. Xie and J. C. Grossman, *PRL* **120**, 145301 (2018).

[7] C. Chen, W. Ye, Y. Zuo, C. Zheng, and S. P. Ong, *Chem. Mater.* **31**, 3564–3572 (2019).

[8] C. Chen, E. Xu, D. Yang, C. Yan, T. Wei, H. Chen, Y. Wei, and M. Chen, *Neural Comput. Apl.* **37**, 3287–3301 (2024).

[9] J. Cheng, C. Zhang, and L. Dong, *Commun. Mater.* **2**, 92 (2021).

[10] C. W. Park and C. Wolverton, *Phys. Rev. Mater.* **4**, 063801 (2020).

[11] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, *34th Int. Conf. Mach. Learn.* **70**, 1263–1272 (2017).

[12] R. Ruff, P. Reiser, J. Stühmer, and P. Friederich, *Digit. Discov.* **3**, 594–601 (2024).

[13] K. Choudhary and B. DeCost, *npj Comput. Mater.* **7**, 1–8 (2021).

[14] C. Chen and S. P. Ong, *Nat. Comput. Sci.* **2**, 718–728 (2022).

[15] H. Hajiabolhassan, Z. Taheri, A. Hojatnia, and Y. T. Yeganeh, *J. Chem. Info. Model.* **63**, 3275–3287 (2023).

[16] B. E. Husic, N. E. Charron, D. Lemm, J. Wang, A. Pérez, M. Majewski, A. Krämer, Y. Chen, S. Olsson, G. De Fabritiis, et al., *J. Chem. Phys.* **153** (2020).

[17] B. H. Lee, J. P. Larentzos, J. K. Brennan, and A. Strachan, *npj Comput. Mater.* **10**, 208 (2024).

[18] C. Cai, D. Wang, and Y. Wang, *arXiv:2102.01350* (2021).

[19] V. G. Satorras, E. Hoogeboom, and M. Welling, in International conference on machine learning (PMLR, 2021), 9323–9332.

[20] M. Geiger and T. Smidt, *arXiv:2207.09453* (2022).

[21] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, *Nat. Commun.* **13**, 2453 (2022).

[22] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley, *arXiv:1802.08219* (2018).

[23] Y. Zhong, H. Yu, M. Su, X. Gong, and H. Xiang, *npj Comput. Mater.* **9**, 182 (2023).

[24] L. Pauling, *J. Chem. Soc.* **51**, 1010–1026 (1929).

[25] R. King, *J. Chem. Ed.* **73**, 993 (1996).

[26] D. Waroquiers, X. Gonze, G.-M. Rignanese, C. Welker-Nieuwoudt, F. Rosowski, M. Gobel, S. Schenk, P. Degelmann, R. André, R. Glaum, et al., *Chem. Mater.* **29**, 8346–8360 (2017).

[27] J. Dan, X. Zhao, Q. He, N. D. Loh, and S. J. Pennycook, *Microsc. Microanal.* **28**, 3002–3003 (2022).

[28] Z. Yang and L.-H. Tang, *Phys. Rev. B* **79**, 045402 (2009).

[29] K. Sheriff, Y. Cao, and R. Freitas, *npj Comput. Mater.* **10**, 215 (2024).

[30] N. E. R. Zimmermann, M. K. Horton, A. Jain, and M. Haranczyk, *Front. Mater.* **4**, 34 (2017).

[31] N. E. Zimmermann and A. Jain, *RSC Adv.* **10**, 6063–6081 (2020).

[32] E. E. Santiso and B. L. Trout, *J. Chem. Phys.* **134**, 6 (2011).

[33] J. H. Chang, D. Kleiven, M. Melander, J. Akola, J. M. Garcia-Lastra, and T. Vegge, *J. Phys. Condens. Matter* **31**, 325901 (2019).

[34] A. van de Walle, *Nat. Mater.* **7**, 455–458 (2008).

[35] J. Sanchez, F. Ducastelle, and D. Gratias, *Physica A Stat. Mech. Appl.* **128**, 334–350 (1984).

[36] Q. Wu, B. He, T. Song, J. Gao, and S. Shi, *Comput. Mater. Sci.* **125**, 243–254 (2016).

[37] R. Drautz, *Phys. Rev. B* **99**, 014104 (2019).

[38] G. Dusson, M. Bachmayr, G. Csányi, R. Drautz, S. Etter, C. van der Oord, and C. Ortner, *J. Comput. Phys.* **454**, 110946 (2022).

[39] Y. Lysogorskiy, C. v. d. Oord, A. Bochkarev, S. Menon, M. Rinaldi, T. Hammerschmidt, M. Mrovec, A. Thompson, G. Csányi, C. Ortner, and R. Drautz, *npj Comput. Mater.* **7**, 97 (2021).

[40] R. Mulas, D. Horak, and J. Jost, *Graphs, simplicial complexes and hypergraphs: spectral theory and topology*, edited by F. Battiston and G. Petri (Springer International Publishing, Cham, 2022), 1–58.

[41] F. Eijkelboom, R. Hesselink, and E. J. Bekkers, in International conference on machine learning (PMLR, 2023), 9071–9081.

[42] D. Shi and G. Chen, *Natl. Sci. Rev.* **9**, nwac038 (2022).

[43] S. Ebli, M. Defferrard, and G. Spreemann, *arXiv:2010.03633* (2020).

[44] M. Pinsky and D. Avnir, *Inorg. Chem.* **37**, 5575–5582 (1998).

[45] D. Waroquiers, J. George, M. Horton, S. Schenk, K. A. Persson, G.-M. Rignanese, X. Gonze, and G. Hautier, *Acta Crystallogr. B* **76**, 683–695 (2020).

[46] H. Pan, A. M. Ganose, M. Horton, M. Aykol, K. A. Persson, N. E. R. Zimmermann, and A. Jain, *Inorg. Chem.* **60**, 1590–1603 (2021).

[47] Z. Chen, X. Li, and J. Bruna, *arXiv:1705.08415* (2017).

[48] A. Dunn, Q. Wang, A. Ganose, D. Dopp, and A. Jain, *npj Comput. Mater.* **6**, 138 (2020).

[49] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. A. Persson, *APL Mater.* **1**, 011002 (2013).

[50] H. R. Banjade, S. Hauri, S. Zhang, F. Ricci, W. Gong, G. Hautier, S. Vucetic, and Q. Yan, *Sci. Adv.* **7**, eabf1754 (2021).

[51] S. Zhang, Z. Hu, A. Subramonian, and Y. Sun, *arXiv:2012.12533* (2020).

[52] C. R. Collins, G. J. Gordon, O. A. Von Lilienfeld, and D. J. Yaron, *J. Chem. Phys.* **148**, 24 (2018).

[53] N. Fedik, R. Zubatyuk, M. Kulichenko, N. Lubbers, J. S. Smith, B. Nebgen, R. Messerly, Y. W. Li, A. I. Boldyrev, K. Barros, et al., *Nat. Rev. Chem.* **6**, 653–672 (2022).

[54] K. Yan, C. Fu, X. Qian, X. Qian, and S. Ji, *arXiv:2403.11857* (2024).

Figure Legends/Captions

Figure 1: **Degenerate Graph Example.** An example of two distinct geometries that are mapped to the same distance-based crystal graph. With inclusion of a first-shell feature vector encoding local geometry however, these structures are mapped to two distinct crystal hypergraphs. Note these are two possible coordination environments in oxides, determined statistically in [26].

Figure 2: **Crystal Hypergraph Construction.** Typical construction loop for a crystal hypergraph. First, pair-wise bonds/edges are determined, then triplets are derived from overlapping pairs of bonds, and finally motifs are determined as first-shells of neighbors by some (generally more restrictive) criteria. Features for each and upper bounds on numbers of hyperedges for each type are also listed. Note that RBF abbreviates radial basis function (such as Gaussian, Bessel, etc.); CSM, continuous symmetry measure; and, LSOP, local structure order parameter.

Figure 3: **Hypergraph Message Passing.** Overview of three possible message functions M for nodes that generalize the message function in [11] to hyperedges h with more (or less) than two nodes. Note that AGG specifies some aggregation function acting on the set of node features in the hyperedge, such as max, min, or average. These messages m are then used through T convolutional layers to update the node representations n before readout.

Figure 4: **Model Architecture.** Example architecture for the crystal hypergraph convolutional network implemented in this work. Essentially, the model is a generalization of CGCNN's [6] model architecture with CGConv being replaced by R hypergraph convolutional layers (CHGConv). Here, CHGConv updates nodes first according to edges (bonds), and then according to triplets or motifs.

Hyperedge Types	ω_p (*1,265*) MAE (cm^{-1})	n (*4,764*) MAE	E_f (*18,829*) MAE (meV/Atom)	$\text{Log}_{10}(G_{vrh})$ (*10,987*) MAE (Log_{10}GPa)	$\text{Log}_{10}(K_{vrh})$ (*10,987*) MAE (Log_{10}GPa)
Bond (*CGCNN*[6])	57.8	0.599	45.2	0.0895	0.0712
Bond & Triplet	60.7	0.440	42.7	0.0846	0.0666
Bond & Motif	**52.9**	**0.432**	**42.4**	**0.0805**	**0.0636**
Bond Messages	112,596	960,912	1,135,680	1,061,628	1,061,628
Triplet Messages	927,135	5,756,865	13,032,891	10,281,801	10,281,801
Motif Messages	9,383	80,076	94,640	88,469	88,469

Table 1: Test results averaged over 5 folds of nested cross-validation for five MatBench target sets: highest frequency phonon peak ω_p, refractive index n, perovskite formation energy E_f, and bulk moduli K_{vrh} and shear moduli G_{vrh}. Note that the italicized numbers below the target name in parentheses correspond to the total size of each dataset, while test results are reported as an average on the test sets of the 5 MatBench [48] folds. Best results are indicated in bold. The number of messages contributed per layer for each type of hyperedge is listed below the line dividing the tables elements for each target set, respectively. Note that the bond-only results are the posted best performance for CGCNN [6], since our code is a direct adaption of theirs with additional motif and triplet layers. Performance on each MatBench fold is given in Supplementary Note S9.

Hyperedge Types	E_f (*132,752*) MAE (eV/Atom)	E_g (*106,113*) MAE (eV)	Metal/Non-metal Classification (*106,113*) AUC
Bond (*CGCNN*[6])	**33.7**	0.30	95.2%
Bond & Motif	39.7	**0.23**	**95.8**%
Bond Messages	46,435,896	38,219,808	38,219,808
Motif Messages	3,869,658	3,184,984	3,184,984

Table 2: Test results for three Materials Project target sets: formation energy E_f, band gap E_g, and metallicity. Note that the italicized numbers below the target name in parentheses correspond to the total size of each dataset, while test results are reported as an average on the test sets of the 5 MatBench [48] folds. Best results are indicated in bold. The number of messages contributed per layer for each type of hyperedge is listed below the line dividing the tables elements for each target set, respectively. Note that the bond-only results are the posted best performance for CGCNN [6], since our code is a direct adaption of theirs with additional motif and triplet layers. Performance on each MatBench fold is given in Supplementary Note S9.

Tables