Crystal Hypergraph Neural Networks

A Universal Framework for Material Machine Learning

Alex Heilman

September 19, 2023

Overview

- Crystal Graphs Lack Higher Order Geometrical Info of Crystal Structure

Overview

- Crystal Graphs Lack Higher Order Geometrical Info of Crystal Structure – Motifs!

Overview

- Crystal Graphs Lack Higher Order Geometrical Info of Crystal Structure – Motifs!

- Use Hypergraphs, with Motif Info/Shapes Encoded as Higher Order Hyperedge (Hedge) Attributes

Overview

- Crystal Graphs Lack Higher Order Geometrical Info of Crystal Structure – Motifs!

- Use Hypergraphs, with Motif Info/Shapes Encoded as Higher Order Hyperedge (Hedge) Attributes

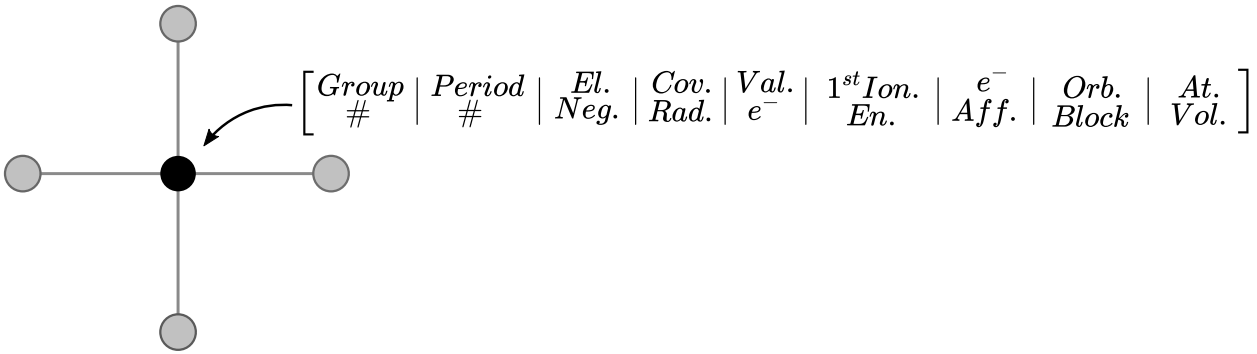
How to handle these?

- Convert Hypergraphs into 'Relatives' (Hetero) Graph to Feed into Usual (Hetero) Graph Neural Networks, like CGCNN

- Apply a novel convolutional structure adaptable to variable size hyperedges

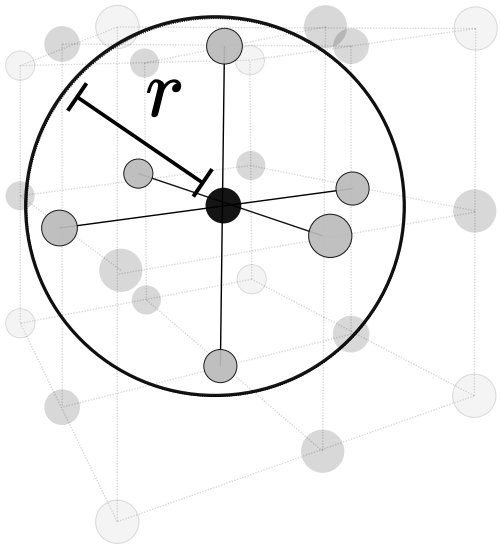
Usual Crystal Graph Construction (a la CGCNN)

- Atoms are nodes, with initial node features determined by atomic properties

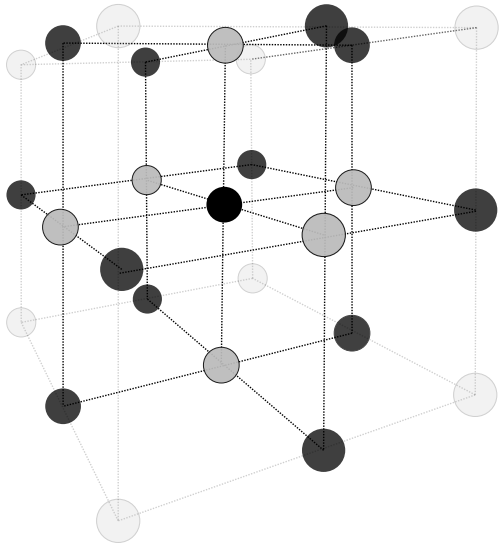


Crystal Graphs cont. I

- Edges are determined by distance cutoff (4 Ang.) and maximum number of neighbors (12)



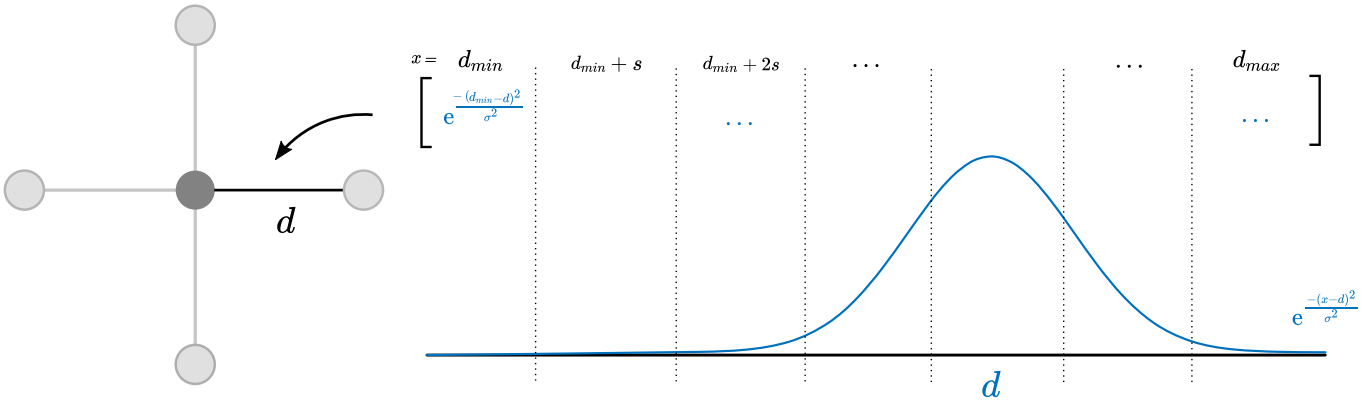
(within) Max radius r



N nearest neighbors

Crystal Graphs cont. II

Edge attributes then are a Gaussian distance expansion



Graph Limitations

Problem: Only encodes distances between atoms!

Recent works: Atomistic Line graph generates a second graph (the line graph) associated with the crystal graph

In line graph, nodes represent edges and line graph edges represent overlapping/connected edges.

Allows for encoding of angle information (between bonds) as edge attributes in line graph

Hypergraphs

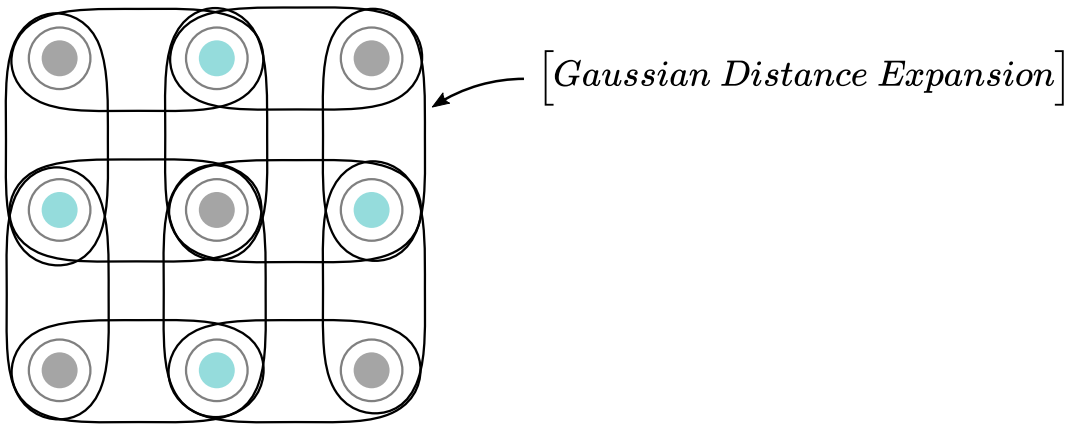
Hypergraphs allow us to have edges containing more than (or less than) two nodes.

Natural way to encode features with higher order structure, where hypergraph nodes still represent atoms of underlying crystal structure.

Can treat all different order structures on equal footing: each has a corresponding hyperedge with a feature

Pair Hedges

We first generate pair-wise (second order) hyperedges, equivalent to regular crystal graph edges.



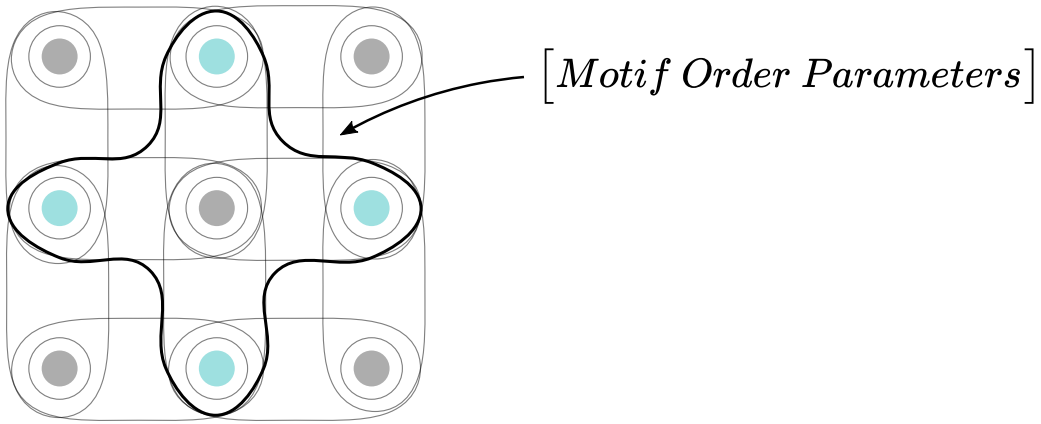
Motif Hedges

Wish to include higher order geometrical information of crystal structure. Motifs have important information!

Motif Hedges

Wish to include higher order geometrical information of crystal structure. Motifs have important information!

Generate motif hedges, associate local structure order parameters (describing local environment's shape) as feature.



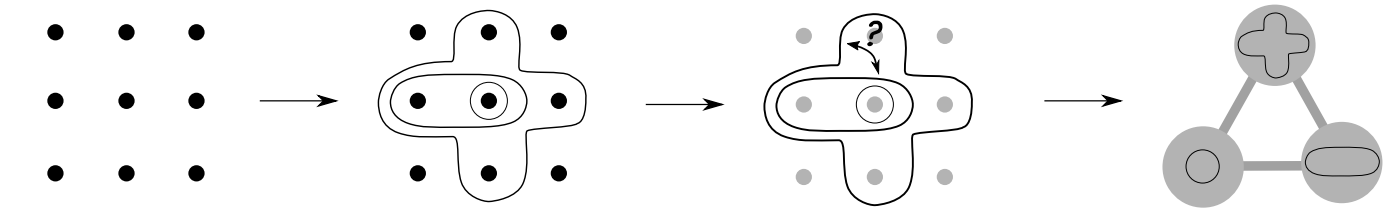
Hypergraph Convolutions: Two Approaches

First idea: form a relatives **graph** corresponding to the original hypergraph, allowing us to use the usual graph convolution functions

Second idea: apply a novel **hypergraph convolution** directly to the hypergraph

Relatives Graph Construction

We term this inherited graph from the hypergraph the relatives graph, since connected nodes are related structures.



Note that this is naturally a heterogeneous graph, in that there are different types of nodes

Relatives Graph Convolution

Applying arbitrary graph convolutional operators to this relatives graph essentially leaves us with the following form for the layer-to-layer updating node-wise.

$$\begin{aligned} x_i \to x_i &+ f_{xx}(x_i \oplus x_j) \\ &+ f_{xb}(x_i \oplus b_j) \\ &+ f_{xm}(x_i \oplus m_j) \end{aligned}$$

Each different order structure has it's own (directional) update function.

Unfortunately, this seems to hinder performance. Perhaps this occurs since neighboring node features may not interact as strongly without being concatenated in the message forming phase.

Hypergraph Message Passing

In the case of a hypergraph, the neighborhood of nodes relevant to each message are now a set (instead of the single neighboring node feature of classic MPNNs)

$$m_v^{t+1} = \sum_{h_j \in \mathcal{N}(v)} M_t(n_v^t, \underbrace{h_j^t, \{n_w^t | n_w \in h_j\}}_{e_{vw}, n_w})$$

$$n_v^{t+1} = U_t(n_v^t, m_v^{t+1})$$

$$\hat{y} = R(\{n_v^T\})$$

The rest of the structure may essentially remain the same

Hypergraph Convolution

To deal with the variable sized neighborhoods of features, we simply aggregate the neighborhood features in the message passing phase, so that we essentially have the following form of convolution:

$$x_i \to x_i + f_b(x_i \oplus e_{ij} \oplus \text{AGG}(\{x_i, x_j\})) \\ + f_m(x_i \oplus m_j \oplus \text{AGG}(\{n_w^t | n_w \in m_j\}))$$

Current Problems/Outlook:

- CGConv baseline should line up better with CGCNN from the original github
- What other features/structure orders should be included?
- What is the best balance between efficiency and efficacy? Do we even need pairwise edges or motif edges?
- Novel use case: apply to scenarios where motifs are effectively smallest structural unit of crystalline system (ceramics)

Other Project: Spherical Harmonic Decomposition of Elastic Tensor