Crystal Hypergraph Neural Networks

A Universal Framework for Material Machine Learning

Alex Heilman

May 9, 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

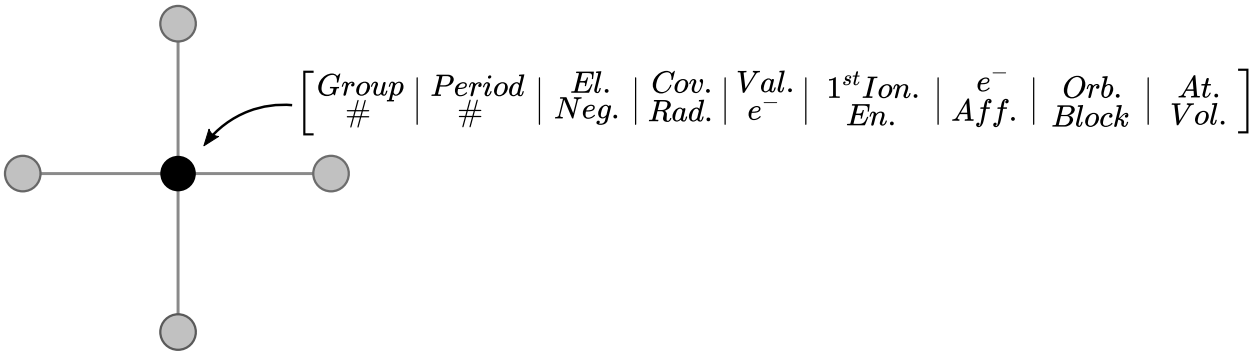
- Crystal Graphs Lack Higher Order Geometrical Info of Crystal Structure – Motifs!

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

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

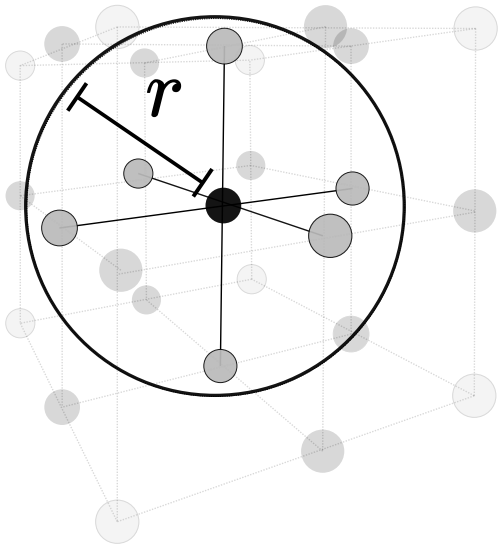
Usual Crystal Graph Construction (a la CGCNN)

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

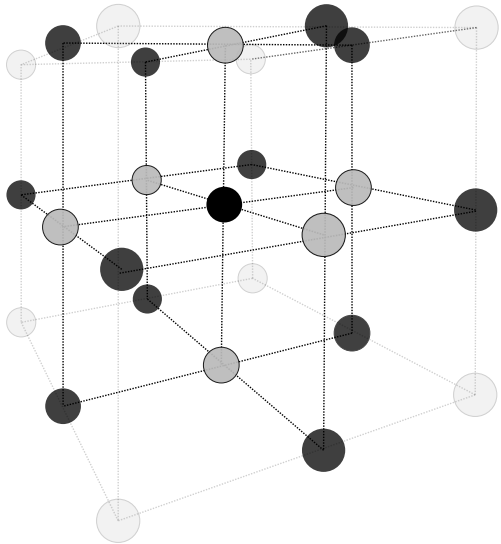


Crystal Graphs cont. I

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



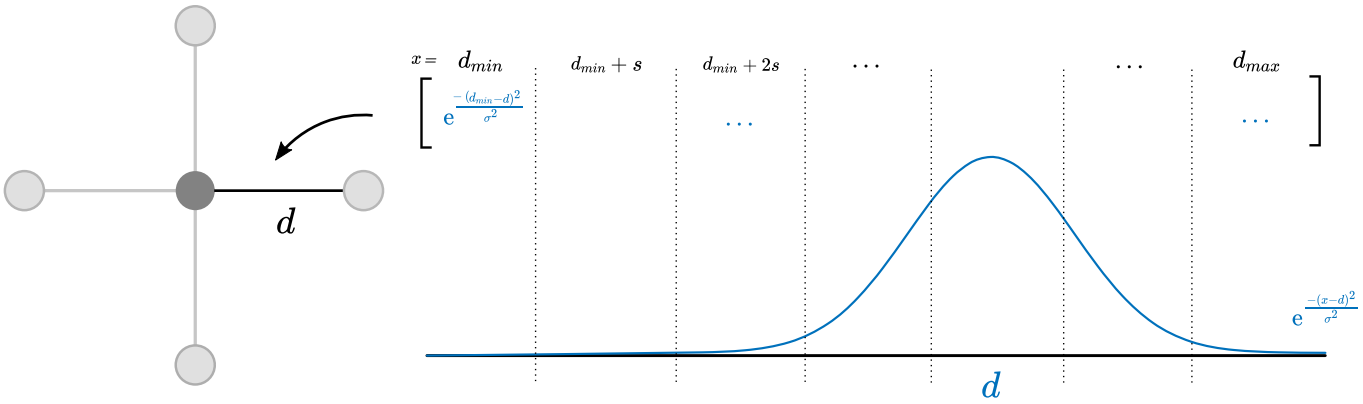
(within) Max radius r



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

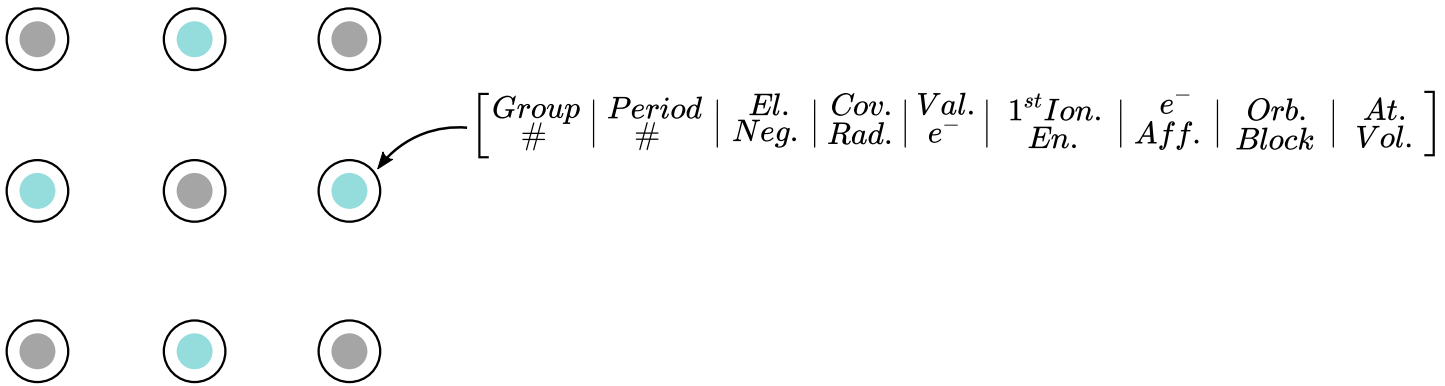
Hypergraphs allow us to have edges containing more than (or less than) two nodes.

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

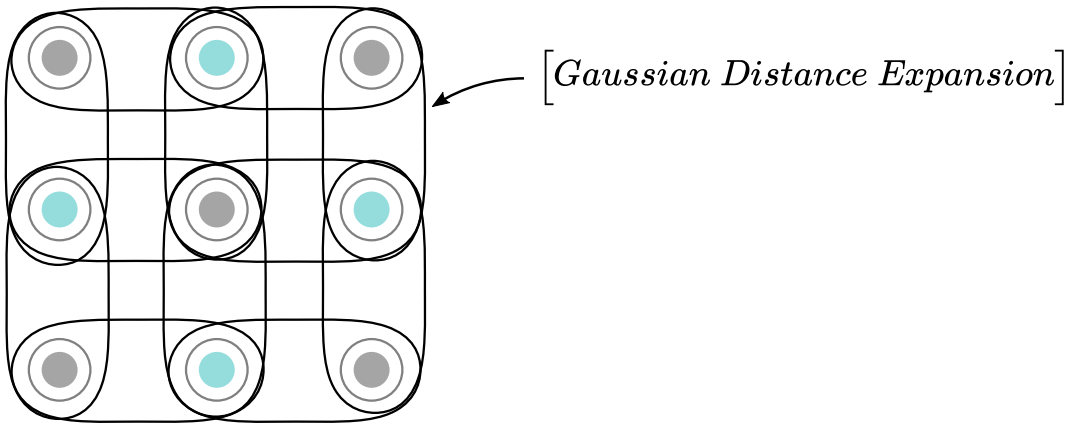
Atom Hedges

We first generate singleton hedges for each atomic site (the reason will be apparent later)



Pair Hedges

Then generate pair (second order) hedges, equivalent to regular crystal graph edges.



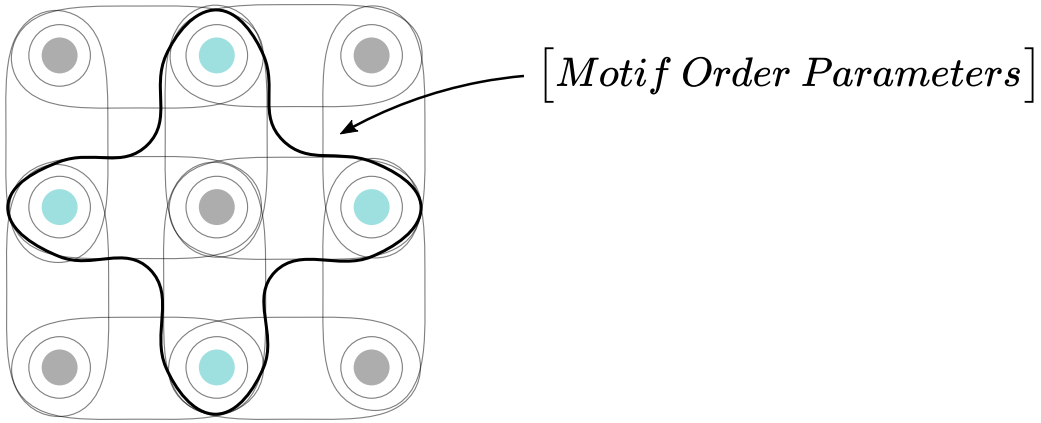
Motif Hedges

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 continuous symmetry measure for common motif shapes as feature.



Next Problem: Current hypergraph convolution doesn't allow us to associate features with hyperedges!

Next Problem: Current hypergraph convolution doesn't allow us to associate features with hyperedges!

Solution: Convert hypergraph into graph!

Next Problem: Current hypergraph convolution doesn't allow us to associate features with hyperedges!

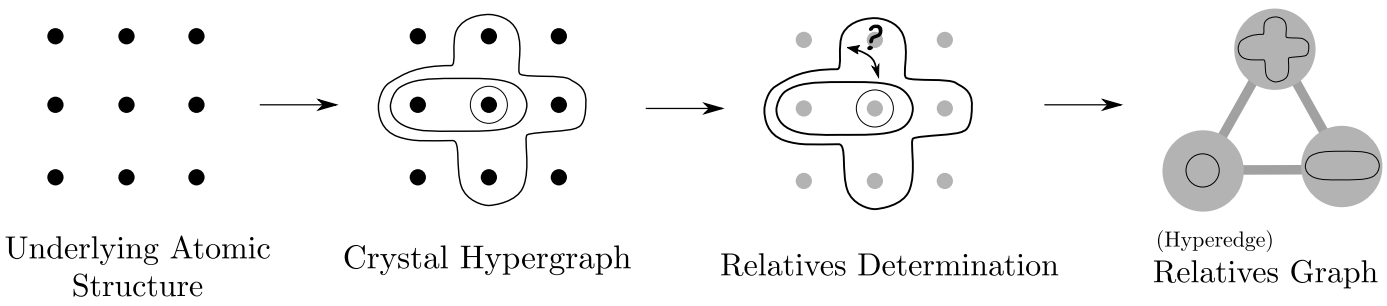
Solution: Convert hypergraph into graph! Then we can use the same graph networks previously developed (similar to Line graph's approach)

Treat all order structures on equal footing; nodes for atoms, bonded atoms, motifs, and unit cell!

Consider all order structures with common nodes connected (corresponding edge in relatives graph).

Relatives Graph

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 (can be projected into a homogeneous graph)

Important things to note:

- CGConv performs much better than SAGEConv and TransformerConv (from Pytorch Geometric Library) on physical property tasks

- Motif information shows substantial increase in performance even for underperforming convolutional structures on relatives graph

Current Problems/Outlook:

- CGConv (the most effective convolutional layer for such tasks) doesn't readily generalize
- What other features/structure orders should be included?
- What is the best balance between efficiency and efficacy?