Crystal Hypergraph Neural Networks

A Universal Framework for Material System Machine Learning

Qimin Yan, Alex Heilman

October 24, 2023

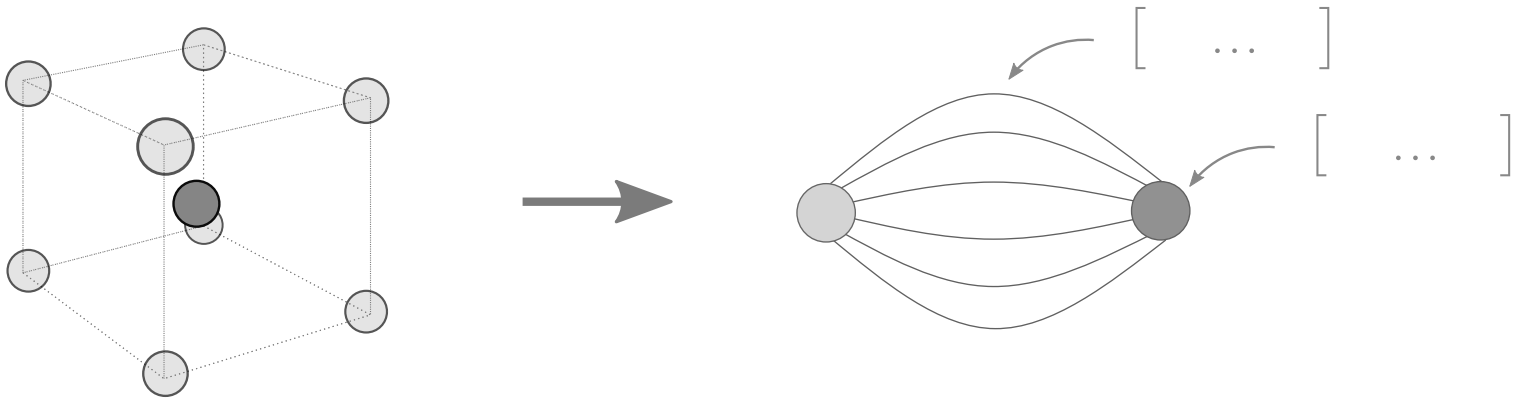
Machine Learning on Crystal Systems

To perform predictive tasks for material systems, such as crystalline structures, we essentially need two things:

- A way to represent the material system mathematically
- A trainable predictive model or set of functions which takes, as input, the material system's representation (as well as a large set of data)

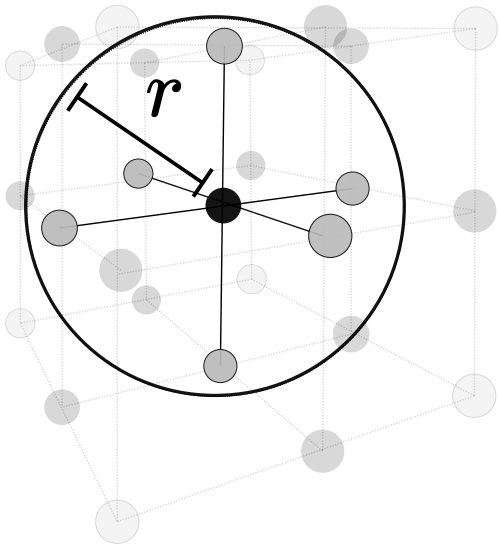
Usual Crystal Graph Construction (a la CGCNN)

The usual technique is to represent crystalline systems as graphs (nodes and edges), with atomic features associated with nodes and geometric features associated with edges

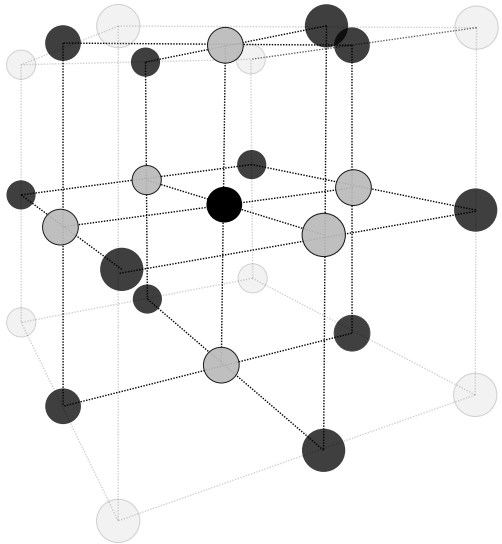


Crystal Graphs cont. I

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



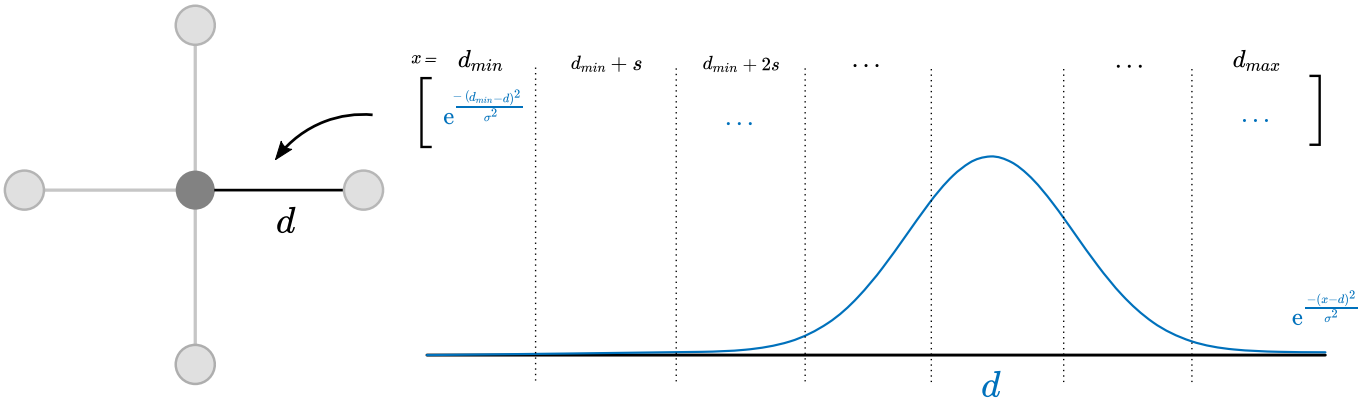
(within) Max radius r



N nearest neighbors

Crystal Graphs cont. II

Edge attributes then are a Gaussian distance expansion



Message Passing on Graphs

Now, we need to update these features through some trainable function that acts on graph representations.

This is most generally accomplished by a message passing network applied to graph representations.

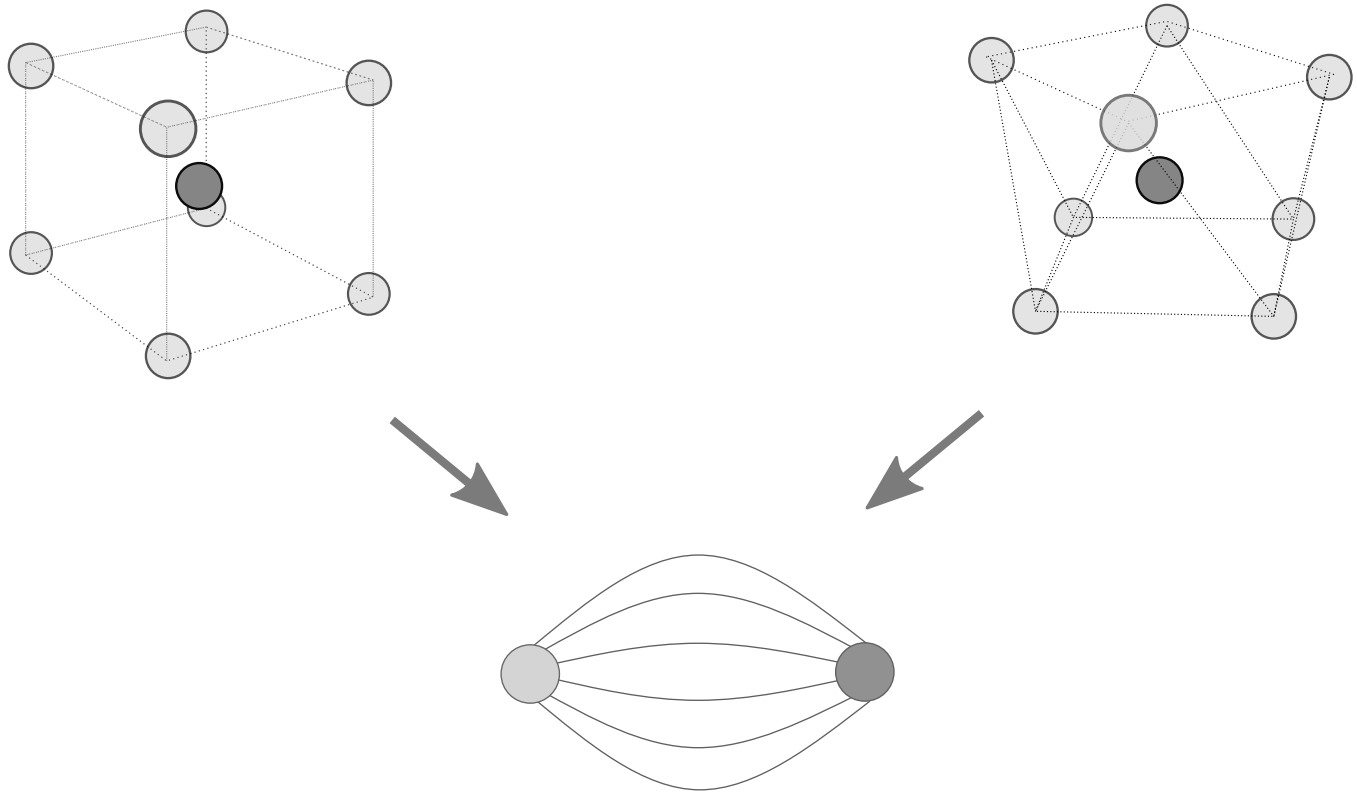
$$m_i^{t+1} = \sum_{n_j \in \mathcal{N}(i)} M_t(n_i^t, e_{ij}, n_j^t)$$

$$n_i^{t+1} = U_t(n_i^t, m_i^{t+1})$$

Here, each node n from layer t to $t+1$ is updated according to an update function U, which takes as input messages formed from each pair of nodes containing the node to be updated.

Graph Limitations

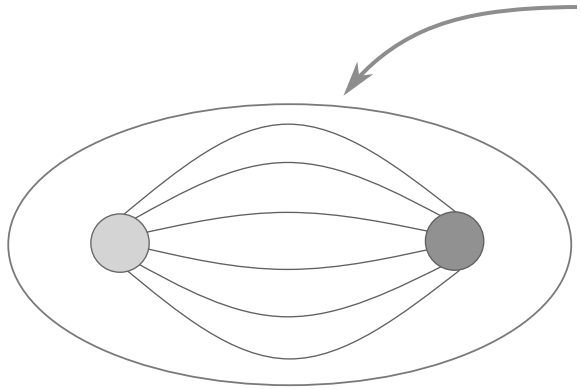
Problem: Our underlying representation encodes only distances between atoms!



So, as a rough example, the above two crystalline structures would have the same representations!

Solution: Hypergraphs!

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

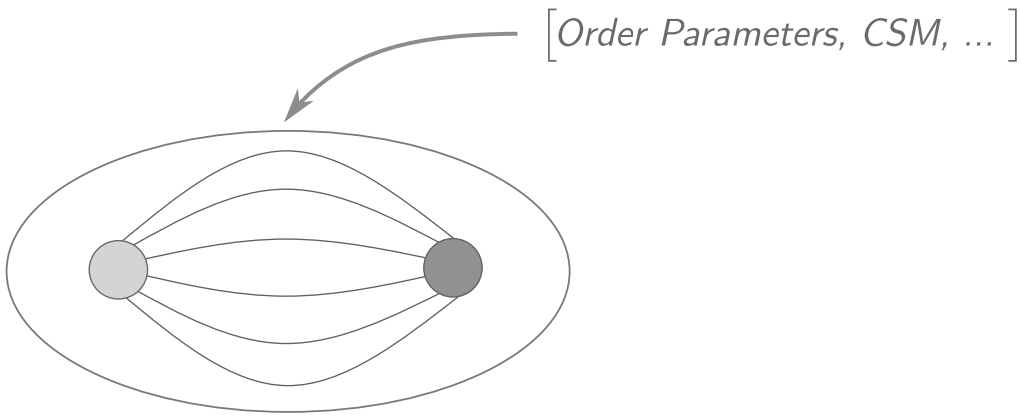


This gives us a natural way to encode features with higher order structure, where hypergraph nodes still represent atoms of underlying crystal structure.

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

Local Environments as Hyperedges

Considering our previous problem, we can now encode this lost geometrical information in larger hyperedges:



This geometrical information can be encoded quantitatively with continuous symmetry measures, local structure order parameters, etc., depending on task

Extending Message Passing to Hypergraphs

Now, we need a suitable convolutional structure that applies to hypergraphs...

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

$$m_i^{t+1} = \sum_{h_j \ni x_i} M_t(n_i^t, \underbrace{h_j^t, \{n_w^t | n_w \in h_j\}}_{e_{ij}, n_j})$$

$$n_i^{t+1} = U_t(n_i^t, m_i^{t+1})$$

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:

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

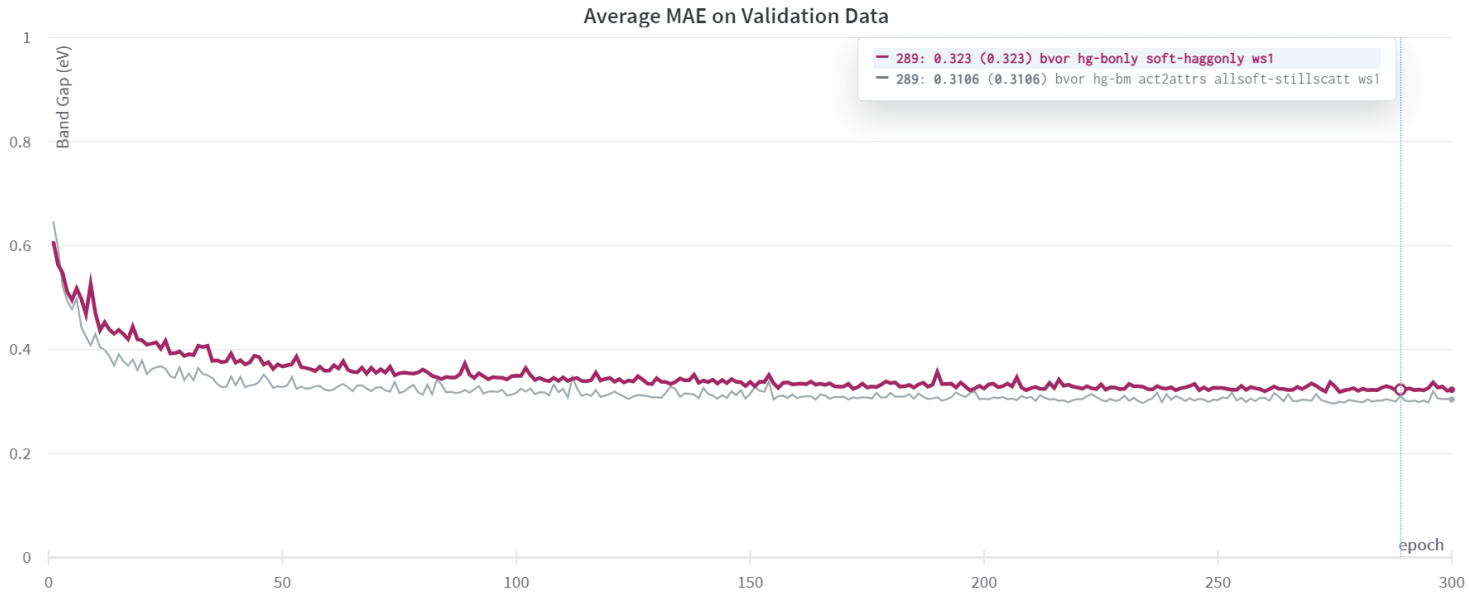
where we only consider pair-wise and local environment hyperedges above (with other order structures, such as triplet and unit cell level hyperedges also fitting into this framework)

Preliminary Results

Preliminary test results suggest this local environment feature benefits most predicitive tasks!

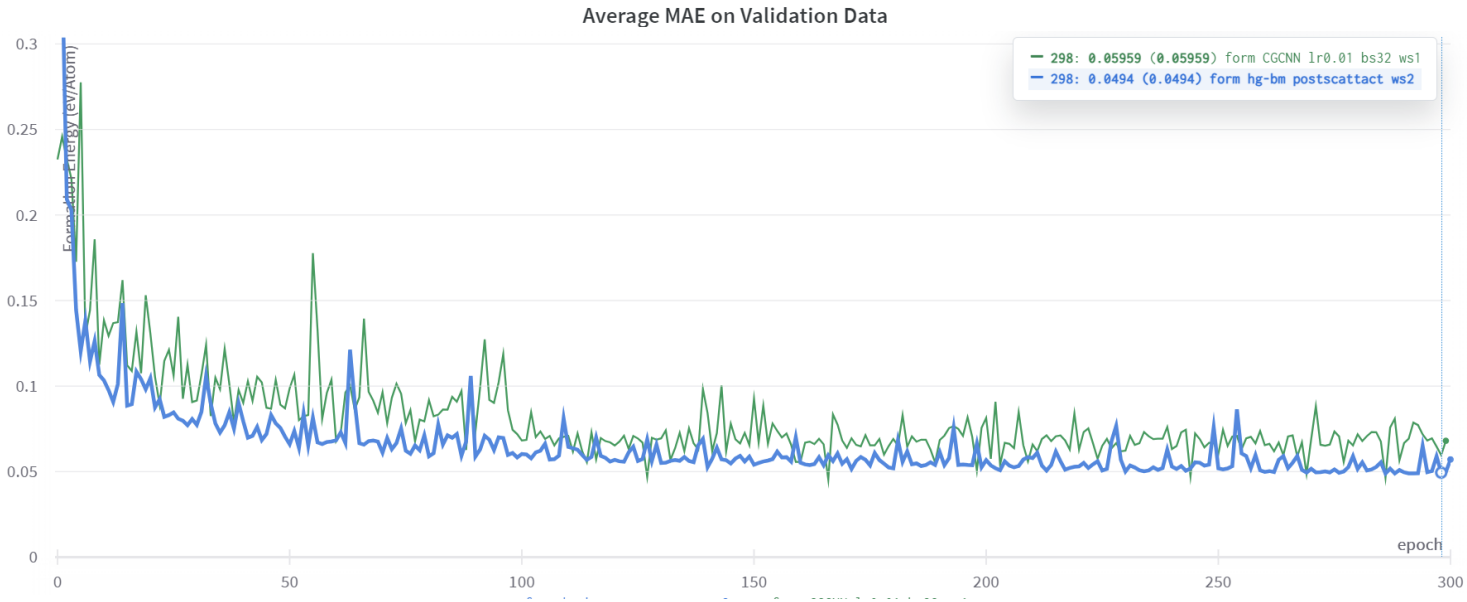
Let's compare performance for crystal graphs, with only bond-level edges, to crystal hypergraphs with bond and local-environment hyperedges (using the same convolutional strucuture for both) on the Materials Project dataset.

Preliminary Results: Band Gap



Here, purple is for bond-only graphs; while grey is for bond and motif level hypergraphs.

Preliminary Results: Formation Energy



Here, green is for bond-only graphs; while blue is for bond and motif level hypergraphs.