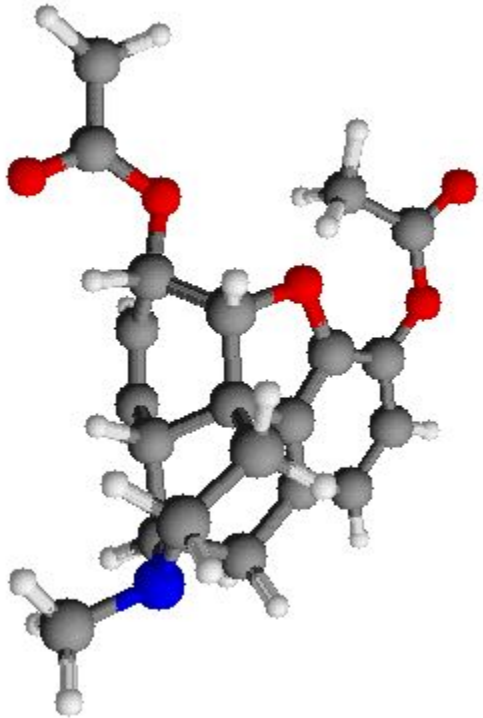
Automated Bond Relaxation

Adam Kadmani and Utkarsh Saraswat



Problem Statement

- Compound/molecular structure given (PT3 O6 C6 Zn)
- Its structure is not ideal -> apply advanced techniques to manipulate atomic interactions within these structures to achieve a “better” structure
- This not only makes **more stable compounds**, but also can unlock new possibilities in material science

Current technique: DFT simulation
Proposed solution: GNN as surrogate model

What is a non-ideal structure?

- Molecular configuration that is not at its **lowest energy state** or most stable form
- Non-ideal structures are often of interest because they can provide insights into **reaction mechanisms** and **molecular behavior under different conditions**

Why is DFT simulation so expensive?

- Density Functional Theory (DFT)
  - Helps in predicting how molecular structures will behave when their bonds are relaxed, as well as calculating the energy of the system in different configurations
    - Provides insights into the most stable configurations of atoms and the electronic properties of the resulting structures
  - Typically $\mathcal{O}(n^3)$, computationally expensive and time consuming, especially for large systems or complex

Using ML for bond relaxation

- Idea: Use **GNN as a surrogate model** for DFT
- GNN is used to approximate the results you would get from a DFT simulation but in a **fraction of the time and computational cost**
- GNN must be trained on a dataset of DFT-computed properties of various molecular structures.

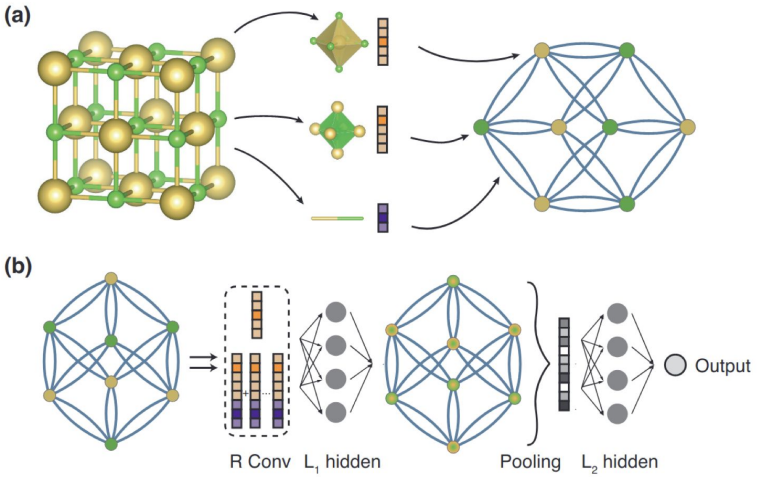
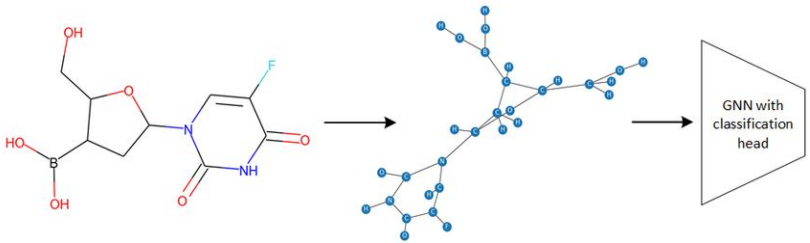


Figure 13 Transforming a crystal structure to a graph (CGCNN-paper).



Enhance Information Propagation for Graph Neural Network by Heterogeneous Aggregations - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/A-molecule-is-converted-to-graph-representation-by-treating-each-atom-as-node-and_fig2_349124961 [accessed 20 Dec, 2023]

Using ML for bond relaxation

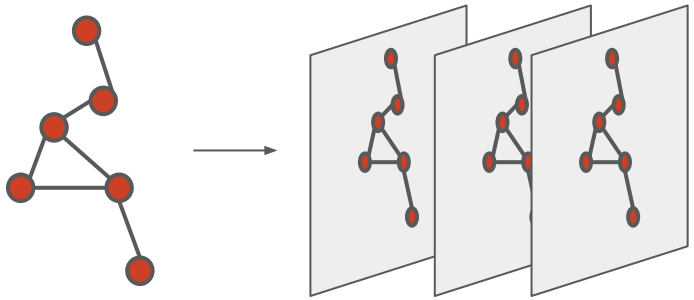


Fig 2. MPNN : at each state parameters in Φ are updated to fit the data

- Message passing neural network

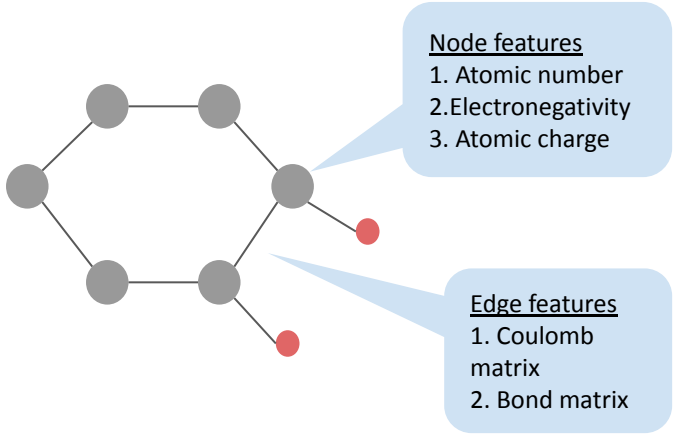
$$\mathbf{x}_i^{(k)} = \gamma^{(k)} \left(\mathbf{x}_i^{(k-1)}, \bigoplus_{j \in \mathcal{N}(i)} \phi^{(k)} \left(\mathbf{x}_i^{(k-1)}, \mathbf{x}_j^{(k-1)}, \mathbf{e}_{j,i} \right) \right),$$

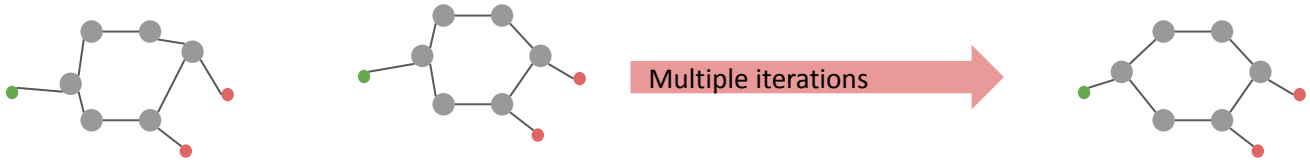
- Edge convolution

$$\mathbf{x}_i^{(k)} = \max_{j \in \mathcal{N}(i)} h_{\mathbf{\Theta}} \left(\mathbf{x}_i^{(k-1)}, \mathbf{x}_j^{(k-1)} - \mathbf{x}_i^{(k-1)} \right),$$

Equation source : https://pytorch-geometric.readthedocs.io/en/latest/tutorial/create_gnn.html

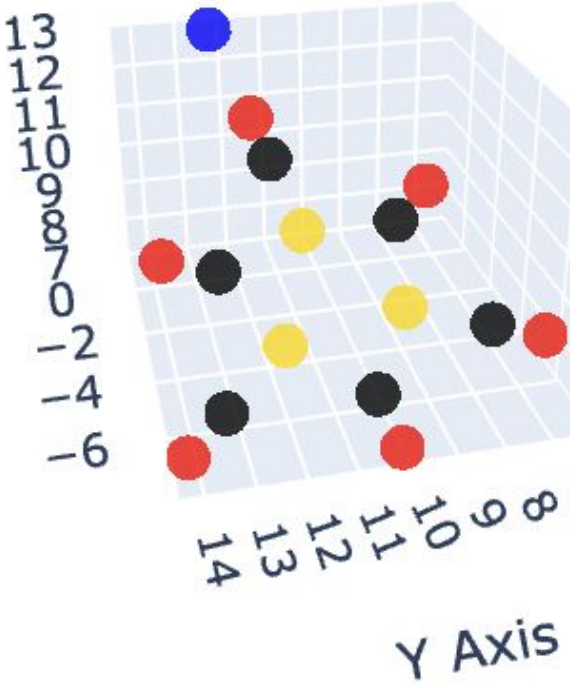
- GNN architecture: **Message Passing Neural Network (MPNN) with edge convolution**
- **Coulomb matrix in** edge features can be interpreted as distance matrix
- Aim: predict Coulomb matrix (or interatomic distance) in next DFT iteration

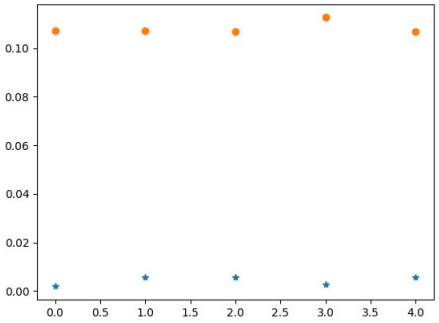


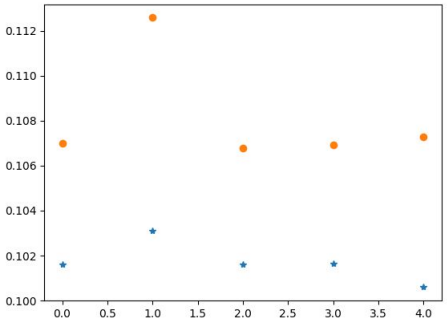


Iteration 1 *Iteration 2* *Converged state*

Fig . Convergence of DFT simulation through multiple iterations


13
12
11
10
9
8
7
0
−2
−4
−6
14
13
12
11
10
9
8
Y Axis


0.10
0.08
0.06
0.04
0.02
0.00
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


0.112
0.110
0.108
0.106
0.104
0.102
0.100
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Results

Transformers?

Thank you