Quiz for DeePTB Recruiting

Zhanghao Zhouyin, Jijie Zou & Qiangqaing Gu

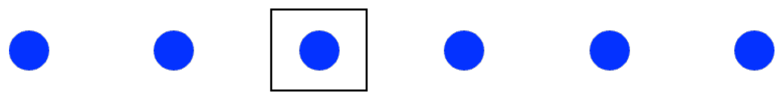
2025-03-16

Instruction For Submission: please submit a single jupyter notebook containing both text answers and the demonstration code. please specify your environment for running the submitted notebook.

Q1: Tight-binding (TB) Knowledge Test (30 marks)

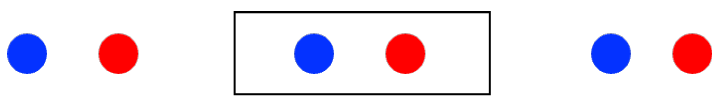
TB approach is a common tool for energy band analysis in condensed matter physics. Based on this method, **derive the energy bands for three types of atomic chains** (as depicted in the figure) under the following conditions. Assume each atom has only one orbital and neglect spin degrees of freedom, only consider the nearest interaction approximation. **Provide necessary formula derivations, complete code, and band structure plots** (parameters can be freely chosen as long as they yield physically reasonable results).

(10 marks) a. One-dimensional periodic atomic chain with uniform spacing. Each unit cell contains a single atom.



(10 marks) b. One-dimensional periodic atomic chain with uniform spacing. Each unit cell contains two distinct atoms.

(10 marks) c. One-dimensional periodic atomic chain with non-uniform spacing. Each unit cell contains two distinct atoms.



Q2: E(3) Group Equivariant Knowledge Test (30 marks)

(5 marks) a. What is the E(3) Group? What is its relation with the SO(3) group and the O(3) group? What are the Clebsch-Gordan coefficients?

(5 marks) b. For equivariant quantities, what is the meaning when we say it is equivariant? Please explain it in words and mathematical expressions if possible. Please name a few equivariant quantities.

(20 marks) c. For a quantum operator in atomic orbital basis, what is the equivariance meaning? Please write down the formula. (where operators are matrix under basis). Please write a small code to prove that the two operator blocks provided are equivalent.

```
O1: array([[ 1.32661455e-01,  2.88550062e-03, -5.17760229e-04],
           [ 6.94089346e-05, -1.87958878e-02,  5.87849393e-04],
           [-7.18132899e-04,  5.75661802e-04, -1.89961633e-02]])
vec: array([-0.0193,  5.3152,  0.0518])

O2: array([[-0.0188, -0.0006,  0.0005],
           [-0.0006, -0.0190, -0.0007],
           [-0.0023, -0.0005,  0.1327]])
vec: array([0., 0., 1.])
```

O1 are O2 are p-p orbital interaction blocks of an LCAO Hamiltonian, with a specific direction vector denoted as "vec", in x,y,z style. (hint, use spatial rotation).

Q3: GNN Knowledge Test (30 marks + 20 bonus)

The general GNN/MPNN structure with edge update:

$$\mathbf{m}^{ij,L} = M_L\left(\mathbf{n}^{i,L-1}, \mathbf{n}^{j,L-1}, \mathbf{e}^{ij,L-1}\right)$$

$$\mathbf{n}^{i,L} = U_L\left(\mathbf{n}^{i,L-1}, \sum_{j\in\mathcal{N}(i)} \mathbf{m}^{ij,L}\right)$$

$$\mathbf{e}^{ij,L} = \mathcal{N}_L\left(\mathbf{n}^{i,L}, \mathbf{n}^{j,L}, \mathbf{e}^{ij,L-1}\right)$$

(5 marks) a. Please explain the update schedule of the above GNN formula.

(5 marks) b. Below is our GNN/MPNN-like network that has a strictly local receptive field used in DeePTB. Please read the update formula, and tell the difference between it and the conventional

GNN/MPNN (mainly, where does the locality come from?).

$$
\begin{aligned}
\mathbf{V}^{ij,L} &= \mathcal{V}_L\left(\mathbf{n}^{i,L-1}, \mathbf{V}^{ij,L-1}\right) \\
\mathbf{m}^{ij,L} &= M_L\left(\mathbf{n}^{i,L-1}, \mathbf{V}^{ij,L}\right) \\
\mathbf{n}^{i,L} &= U_L\left(\mathbf{n}^{i,L-1}, \sum_{j\in\mathcal{N}(i)} \mathbf{m}^{ij,L}\right) \\
\mathbf{e}^{ij,L} &= \mathcal{N}_L\left(\mathbf{n}^{i,L}, \mathbf{V}^{ij,L}, \mathbf{n}^{j,L}, \mathbf{e}^{ij,L-1}\right)
\end{aligned}
$$

(20 marks) c. Please implement a POC code of the updating formula using your favourite deep-learning package. Test its locality on a 2-D honeycomb lattice with first-nearest-neighbour. Show that the update framework in b is invariant with the initial change of node information that is outside the receptive field. (for simplicity, you can assume each node in the lattice only has a single scalar feature).

(20 bonus) d. If we are dealing with a very large graph, inference on a single device is very time-consuming or facing out of memory error. Please explain and analyse the benefits of the the strictly local framework in this case. Can you come up with a sub-graph parallelization method enabling multi-device inference with the above local GNN? Please explain your idea with technical insights into the vital difficulties (no code is needed but you are welcome to write one).