Chemistry 213/313 Homework Assignment 3 Due: May 23, 2025

In this homework assignment, you will gain experience with graph neural networks (GNNs) and their application to molecular property prediction. Specifically, you will implement several GNN models to predict electric dipole moment using the QM9 dataset following this tutorial. It is fine to use code from the tutorial, as long as you understand it. To submit the assignment, export each notebook to HTML and upload the resulting file to Canvas.

1. Data preparation and exploration.
   (a) Download the QM9 dataset and explore the features and how they vary.
   (b) Normalize the target feature (electric dipole moment) and make training, validation, and test sets. Prepare a DataLoader object for batching.
2. '2D' GNN with permutation invariance/equivariance.
   (a) Define a subclass of the MessagePassing class of the PyTorch Geometric library, MPNNLayer, that implements a message passing layer:

$$h_i^{l+1} = \phi_h \left(h_i^l, \sum_{j \in \mathcal{N}_i} \phi_e(h_i^l, h_j^l, a_{ij}) \right),$$

   using the notation in Satorras *et al.* "E(n) equivariant graph neural networks" PMLR (2021). Demonstrate that MPNNLayer is equivariant to permutations.
   (b) Implement an MPNN graph property prediction model, MPNNModel, using the layer above and demonstrate that MPNNModel is invariant to permutations.
   (c) Train the model and plot its training, validation, and test losses as functions of epoch.
3. '3D' GNN with translational and rotational invariance
   (a) Modify the model to make a new one, CoordMPNNModel, that incorporates atom coordinates into the node features.
   (b) Train the model and plot its training, validation, and test losses as functions of epoch. Compare the results with those of the previous model without coordinate information.
   (c) Test if CoordMPNNModel is rotationally and translationally invariant.
   (d) Design a new translationally and rotationally invariant message passing layer, InvariantMPNNLayer, which utilizes both atom coordinates and node features. In turn, make a new model, InvariantMPNNModel, that is also translationally and rotationally invariant.
      i. Write down the update rule of InvariantMPNNLayer and prove that it is translationally and rotationally invariant.
      ii. Implement the layer and model computationally and demonstrate that they are translationally and rotationally invariant in practice.

(e) Train the model, InvariantMPNNModel, and plot its training, validation, and test loss as functions of epoch. Compare the results with those of the previous models, MPNN-Model and CoordMPNNModel.

4. Repeat steps 3(d)-3(e) above, but now for translational and rotational equivariance instead of invariance. Compare the results with the previous models.