XGBoost on Multi-head attention E(n) Equivariant Graph Neural Network to predict Dipole Moment

Dhruv Mittal
CND, IIIT Hyderabad
UG3
2020113017

Nachiket Patil
CSE, IIIT Hyderabad
UG3
2020101018

Abstract

This work introduces a new model to learn graph neural networks equivariant to rotations, translations, reflections and permutations along with multi-head attention and XGBoost. Our model takes in graph-represented molecules as input and passes the input through equivariant message passing layers with attention mechanism applied on the embeddings. These learned features are then extracted by the XGBoost layer to accurately predict the dipole moment vector of the molecule. This method can easily be generalized for predicting other equivariant molecular properties.

***Keywords*—** Graph Neural Network, Attention, Molecular property prediction, equivariant, XGBoost

1 Introduction

Although deep learning has largely replaced hand-crafted features, many advances are critically dependent on inductive biases in deep neural networks. An effective method to restrict neural networks to relevant functions is to exploit the symmetry of problems by enforcing equivariance with respect to transformations from a certain symmetry group. Many other papers use equivariant networks to predict invariant properties of molecules like the dipole moment scalar, polarizability, etc. In order to truly test the equivariance, predicting the dipole moment vector can be done.

This study utilizes the framework XGraphBoost to integrate the GCN-based feature extractions of EGNN and the conventional classifier XGBoost for the molecular property prediction problem.

Attention mechanisms have become almost a de facto standard in many sequence-based tasks. One of the benefits of attention mechanisms is that they allow for dealing with variable sized inputs, focusing on the most relevant parts of the input to make decisions. We introduce an attention-based equivariant graph neural network for the task of property prediction of molecules.

2 Dataset

The QM9 dataset (Ramakrishnan et al., 2014) has become a standard in machine learning as a chemical property prediction task. The QM9 dataset consists of small molecules represented as a set of atoms (up to 29 atoms per molecule), each atom having a 3D position associated and a five dimensional one-hot node embedding that describe the atom type (H, C, N, O, F). The dataset labels are a variety of chemical properties for each of the molecules which are estimated through regression.

The QM9 dataset only contains invariant property of molecules and are not appropriate for evaluating an equivariant network. So, we use the charge distributions and the position vector to calculate the dipole moment vector of the molecule.

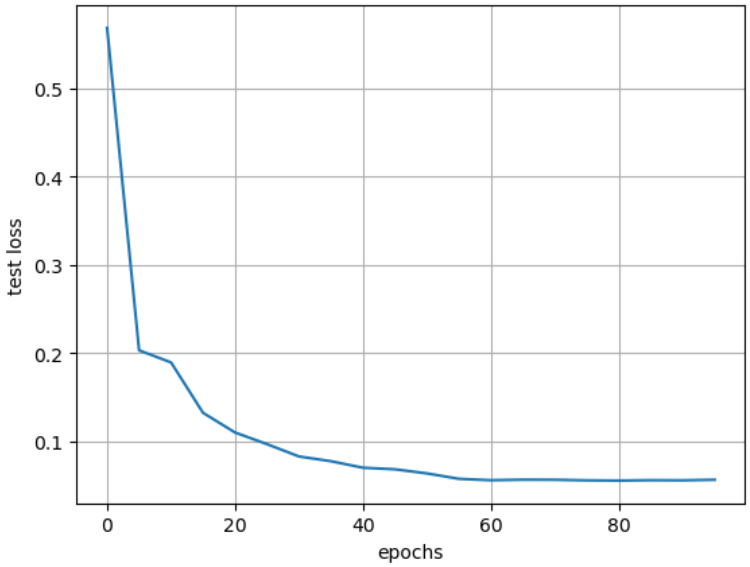

Figure 1: Plot for test loss vs number of epochs obtained after training the baseline model on our system. Parameters -> Number of Message Passing Layers: 7; Number of epochs: 1000; Learning rate: 0.0005; Number of Features: 128; Best Test Loss: **0.05602136177922785**

2.1 Data preparation

We imported the dataset partitions from (Anderson et al., 2019), 100K molecules for training, 18K for validation and 13K for testing. Dipole moment was used as a chemical property for prediction. We optimized and report the Mean Squared Error between predictions and ground truth.

3 Baseline Model

This paper is based on the paper "E(n) Equivariant Graph Neural Networks" by Victor Garcia Satorras, Emiel Hogeboom, Max Welling. The baseline model introduces a new model to learn graph neural networks equivariant to rotations, translations, reflections and permutations called E(n)- Equivariant Graph Neural Networks (EGNNs). In contrast with existing methods, this work does not require computationally expensive higher-order representations in intermediate layers while it still achieves competitive or better performance. Whereas existing methods are limited to equivariance on 3 dimensional spaces, this model is easily scaled to higher-dimensional spaces.

The E(n) Equivariant Graph Neural Network preserves equivariance to rotations and translations on a set of n-dimensional coordinates and it also preserves equivariance to permutations on the set of nodes in the same fashion as GNNs.

The dataset used is QM9 which has molecules with their 3D coordinates. The input to the model is the 3D coordinates of each atom and an embedding of the atom properties. The dipole moment vector was calculated from the charge distribution and the position vectors provided in the dataset according to the formula

$$\bar{\mu} = \sum_{i \in [1,N]} q_i r_i \tag{1}$$

We trained the baseline model on our system, the test loss obtained is shown in Figure 1.The test loss for prediction of dipole moment vector obtained was 0.056.

The baseline model architecture contains 7 layers of message passing which makes updates to the coordinate features such that they remain equivariant as seen in Figure 2. Code availability: `https://github.com/vgsatorras/egnn/tree/main`

$$\mathbf{m}_{ij} = \phi_e\left(\mathbf{h}_i^l, \mathbf{h}_j^l, \left\|\mathbf{x}_i^l - \mathbf{x}_j^l\right\|^2, a_{ij}\right) \tag{3}$$

$$\mathbf{x}_i^{l+1} = \mathbf{x}_i^l + C\sum_{j\neq i}\left(\mathbf{x}_i^l - \mathbf{x}_j^l\right)\phi_x\left(\mathbf{m}_{ij}\right) \tag{4}$$

$$\mathbf{m}_i = \sum_{j\neq i}\mathbf{m}_{ij} \tag{5}$$

$$\mathbf{h}_i^{l+1} = \phi_h\left(\mathbf{h}_i^l, \mathbf{m}_i\right) \tag{6}$$

Figure 2: Message Passing layer used in baseline model.

4 Methodology

4.1 Graph Neural Networks

Graph neural networks (GNNs) are a class of neural networks that operate on graph-structured data. In contrast to traditional neural networks that operate on vector or grid-structured data, GNNs can handle arbitrary graphs, including directed, undirected, and weighted graphs.

A graph is represented as a set of nodes $\mathcal{V}$ and edges $\mathcal{E}$, where each edge (i, j) connects node i to node j. The atoms of a molecule are represented as nodes, and the edges are the interactions between these atoms. Each node i has a feature vector $\mathbf{x}_i$ such as atom type, position and atomic charge, and each edge (i, j) has an optional edge feature vector $\mathbf{e}_{ij}$. Note that every node is connected to each other to incorporate all interactions albeit they'll be weak for far away nodes.

The core idea of GNNs is to iteratively update node features by aggregating information from the features of its neighboring nodes and edges. The updates are performed via message passing, where each node sends a message to its neighbors, and the received messages are used to compute an updated feature vector.

The message passing step can be mathematically formulated as follows:

$$\mathbf{m}_{i\to j}^{(t)} = \mathrm{Msg}^{(t)}(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)}, \mathbf{e}_{ij}^{(t)})$$

where $\mathrm{Msg}^{(t)}$ is a message function that takes as input the current feature vectors of nodes i and j, and the edge feature vector (if any) between them, and outputs a message vector $\mathbf{m}_{i\to j}^{(t)}$ from node i to node j at iteration t.

The messages are then aggregated at each node j to obtain a summary vector $\mathbf{u}_j^{(t)}$

Finally, the updated feature vector $\mathbf{x}_j^{(t+1)}$ for each node j is computed by combining the current feature vector with the summary vector:

$$\mathbf{x}_j^{(t+1)} = \mathrm{Update}^{(t)}(\mathbf{x}_j^{(t)}, \mathbf{u}_j^{(t)})$$

The message passing and update steps are performed for multiple iterations until convergence.

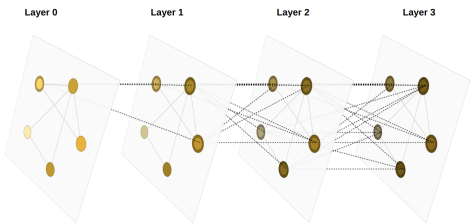

Figure 3: Typical Graph Neural Network. Each dotted line is representative of contributing in the update step of the neighbouring nodes.

GNNs are used in a variety of applications, including node classification, link prediction, and graph classification. GNNs are useful in this paper because they are specifically designed to operate on graph-structured data. Specifically, the paper uses a new type of GNN called E(n)-Equivariant Graph Neural Networks (EGNNs), which are designed to be equivariant to rotations, translations, reflections, and permutations.

4.2 Equivariance

Equivariance is a property of mathematical functions that captures how they transform under a group of symmetries or transformations. Specifically, a function f is equivariant to a transformation g if applying the transformation to the input of the function results in the same transformation being applied to the output of the function, up to a certain group action. That is, for a group G and a function $f : X \to Y$, f is equivariant to a group action $g \in G$ if:

$$f(g(x)) = \rho_g(f(x))$$

where ρ_g is a group representation that captures how the function transforms under g. In other words, the output of f is transformed in the same way as the input under g, but possibly in a different way under different group elements.

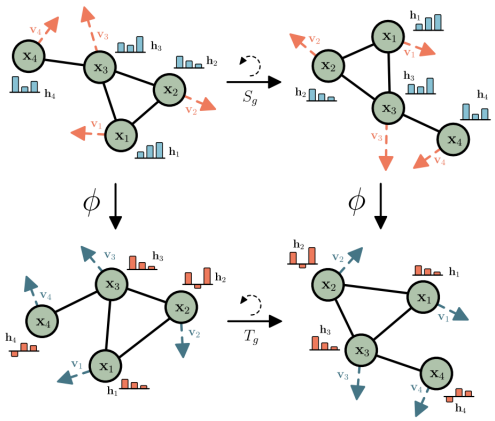

Figure 4: Example of rotation equivariance on a graph with a graph neural network ϕ

In the context of molecules, equivariance is important because molecules exhibit various symmetries and transformations that should be preserved by any learning model that operates on them. For example, molecules may exhibit rotational, translational, or reflectional symmetries, as well as permutation symmetries arising from the exchangeability of atoms. If a learning model is not equivariant to these transformations, it may not be able to capture important structural and chemical properties of the molecules, or it may require a larger amount of training data to learn them. By designing models that are equivariant to these transformations, we can improve their accuracy and robustness, and reduce the amount of data required to train them. This is why the E(n)-Equivariant Graph Neural Networks (EGNNs) introduced in the paper mentioned earlier are specifically designed to be equivariant to rotations, translations, reflections, and permutations, as these transformations are relevant to the representation of molecular structures.

4.3 XGBoost

XGBoost stands for "eXtreme Gradient Boosting," and it is a powerful machine learning algorithm that is widely used for classification and regression tasks. XGBoost is based on gradient boosting, a machine learning technique that builds an ensemble of weak predictive models and combines them to make more accurate predictions.

XGBoost uses a gradient boosting framework and employs a set of advanced techniques to overcome some of the limitations of traditional gradient boosting. These techniques include:

1. XGBoost includes L1 and L2 regularization, which helps in preventing overfitting.
2. Tree pruning is applied to reduce the size of the trees, which helps to improve the generalization performance of the model.
3. Cross-validation can help to optimize the hyperparameters of the model.
4. XGBoost can be run in parallel, which can help to speed up the training process.

Overall, XGBoost is known for its high accuracy and performance on a wide range of machine learning tasks, and it has become a popular algorithm in the data science community.

4.4 Multi-head Attention

Attention in deep learning is a mechanism that allows neural networks to selectively focus on specific parts of input data that are most relevant to a given task. In deep learning, attention can be thought of as a set of weights assigned to different input features, indicating their relative importance for the task at hand. The attention mechanism can improve the accuracy and efficiency of deep learning models by reducing the amount of irrelevant information the model needs to process.

Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. With a single attention head, averaging inhibits this.

We use the scaled dot product attention as used in (Ashish Vaswani et al., 2017). Figure 5 gives a visual representation of the attention layers.

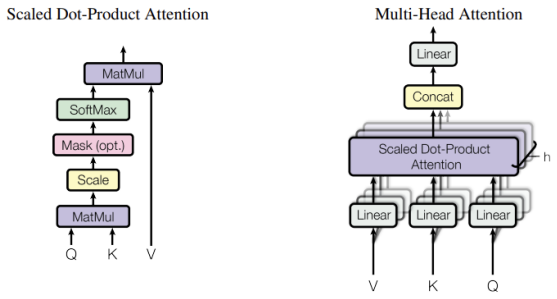

Figure 5: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

4.5 Our Approach

Our proposed model is built upon the baseline model "E(n) Equivariant Graph Neural Networks". The input to the model is in the form of a graph with the molecules represented as nodes and the interactions between the molecules as edges of the graph. An Equivariant Graph Convolutional Layer is defined which takes in as input the node embeddings, coordinate embeddings and the edge features and outputs a transformation on the node and coordinate embeddings. Concisely: $h^{l+1}, x^{l+1} = EGCL[h^l, x^l]$. The EGCL is defined in Figure 2. Here, we can observe the equivariant nature of the message passing layer, if we translate x_i^l by t then x_i^{l+1} is also translated by t and if we rotate the coordinate embedding by a matrix R then x_i^{l+1} will also be rotated.

We propose to incorporate multi-head attention in the model to discard irrelevant features and obtain more accurate results. Attention mechanism has been applied on the node embeddings and coordinate embeddings parallely which are concatenated. We use the scaled dot product attention

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

Finally, we introduce an XGBoost layer to our model. The output layer of the attention-based EGNN is replaced by a supervised learner XGBoost. XGBoost takes in the learned representations of the input data as sample features and predicts the dipole moment vector.

5 Results

We trained our proposed model on the QM9 dataset with a train/test/validation split of 100K/13K/18K respectively. Our model can estimate the dipole moment vector of a given molecule but can be easily modified to predict other equivariant properties. We optimized and report the Mean Squared Error between predictions and ground truth.

We ran our model for a 100 epochs and obtained the test loss curve as seen in Figure 6

We compare our model with various settings: once without multi-head attention, once without XGBoost and once without both, which is the Baseline model. The test loss curves for each of these are shown for comparison in Figure 7

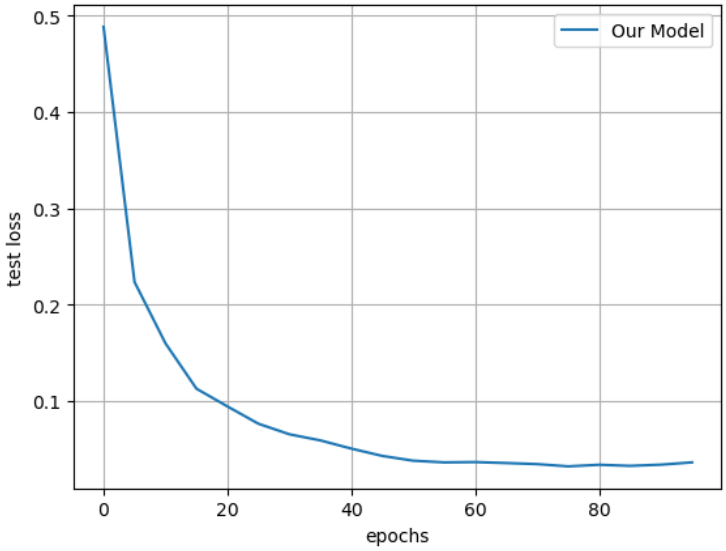

Figure 6:

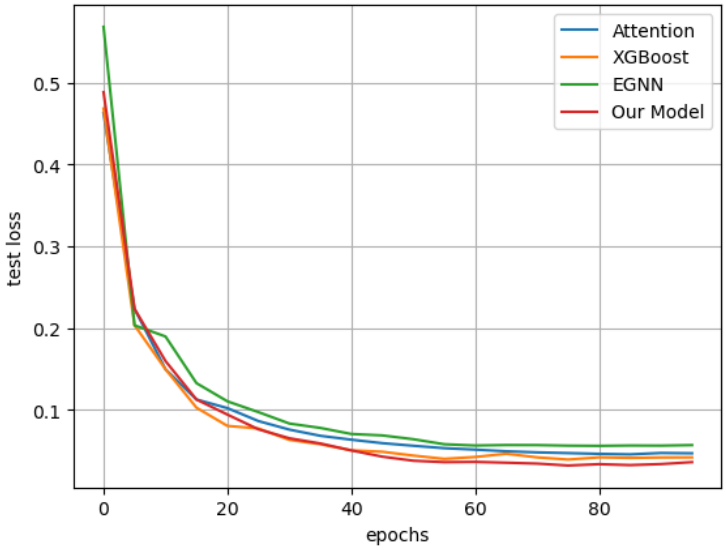

Figure 7:

We can observe the effects of the changes by comparing the test loss curves. The test loss between Attention Curve (Blue) and Baseline Curve (Green) decreases at a similar rate at the start but after a some epochs, the test loss of the blue curve decreases significantly more compared to the green curve displaying the fact that the Attention model was able to learn relevant features accurately with lesser training as compared to the Baseline Model. When we compare the test loss curves of XGBoost model (Orange) and Baseline model we observe that the orange curve is consistently below the green curve after a few epochs displaying the higher accuracy of prediction of the XGBoost layer compared to the baseline model. Although, it was noticed that XGBoost model took a much longer time to train compared to the Baseline Model. Our proposed model outperforms the three comparison models by predicting much more accurately with the trade-off for time taken for training.

The final test losses can be compared in Table 1

Task Units	μ D
EGNN	0.056
Attention	0.046
XGBoost	0.039
Our Model	0.032

Table 1: Mean Squared Error for the dipole moment vector prediction benchmark in QM9 dataset.

6 Future Work

- Tuning of hyperparameters: The hyperparameters of a model such as learning rate, regularization parameter, and number of layers can significantly impact the performance of the model. Tuning these hyperparameters using techniques such as grid search or Bayesian optimization can help to improve the results of a model.
- Model evaluation with a different dataset: We must ensure that the model is able to generalize to new, unseen data. There may be some inherent patterns among the datapoints of a dataset that the model learns and performs well on the test set. Upon evaluating a dataset with very different types of molecules we may truly observe the generalisability of the model.

7 Conclusion

We introduced a modified E(n) equivariant deep architecture for graphs in this study. The findings were significantly enhanced by the XGBoost layer after we added multi-head attention to the model.

The graph attentional layer used in these networks is parallelizable across all nodes in the graph and enables (implicitly) assigning different importances to various nodes within a neighbourhood, and handles various neighbourhood sizes.

The use of the supervised learner XGBoost as the multi-head attention EGNN model's output layer demonstrates that the performance of the neural networks may still be enhanced by using traditional machine learning techniques.

References

[1] Li, Yuan, et al. "EGNN: Constructing explainable graph neural networks via knowledge distillation." Knowledge-Based Systems 241 (2022): 108345.

[2] Daiguo Deng, Xiaowei Chen, Ruochi Zhang, Zengrong Lei, Xiaojian Wang, and Fengfeng Zhou Journal of Chemical Information and Modeling 2021 61 (6), 2697- 2705 DOI: 10.1021/acs.jcim.0c01489

[3] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016.

[4] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

[5] Ramakrishnan, Raghunathan, et al. "Quantum chemistry structures and properties of 134 kilo molecules." Scientific data 1.1 (2014): 1-7.

[6] Anderson, B., Hy, T.-S., and Kondor, R. Cormorant: Covariant molecular neural networks. arXiv preprint arXiv:1906.04015, 2019.

[7] Sanchez-Lengeling, et al., "A Gentle Introduction to Graph Neural Networks", Distill, 2021.

[8] Veličković, Petar, et al. "Graph attention networks." arXiv preprint arXiv:1710.10903 (2017).