MOLECULAR PROPERTY PREDICTION USING GRAPH NEURAL NETWORKS

Beatriz Braga (s233576), João Miranda (s232660), Mafalda Pires (s232433), Riccardo Conte (s241925)

ABSTRACT

Predicting chemical properties efficiently is critical for advancements in various scientific fields. Traditional computational methods, while precise, can be time-consuming. The Polarizable Atom Interaction Neural Network (PaiNN) is a novel Graph Neural Network that leverages rotationally equivariant representations to accurately model molecular interactions. Applied to the QM9 dataset, PaiNN predicts the Internal Energy at 0K of molecules with high accuracy. Performance is further enhanced using stochastic weight averaging (SWA) and its Gaussian variant (SWAG). Results indicate that deeper architectures achieve increasingly better accuracy without leading to oversmoothing. These findings demonstrate PaiNN's potential for efficient and accurate molecular property prediction.

Keywords – Polarizable Atom Interaction Neural Network (PaiNN), Molecular Property Prediction, Graph Neural Networks (GNNs), QM9, Equivariant Neural Networks

1. INTRODUCTION

The accurate prediction of chemical properties and efficient simulation of molecular dynamics are critical to advancements in fields like drug discovery, materials science, and chemical engineering. Traditional computational methods, while precise, often involve complex calculations that can be prohibitively slow and computationally expensive, especially for large molecules or extensive datasets. One promising alternative is Graph Neural Networks (GNNs), which have become a dominant and fast-growing method in deep learning with graph data[1].

Molecules have specific configurations that can be represented as graphs where atoms are represented as vertices and bonds as edges. These structures can be handled by GNNs that preserve molecular properties and structures. However, given that standard message-passing formulations do not consider rotationally equivariant representations, Kristof T. Schütt, et al.[2], proposed a different approach: Polarizable Atom Interaction Neural Network (PaiNN). This approach captures geometric information and physical interactions, as it takes rotational equivariance of molecules into account.

The main goal of this project is to implement PaiNN and apply it to predict the Internal Energy at 0K of different molecules based on nuclear charges ($Z_i \in \mathbb{N}$) and atom positions ($r_i \in \mathbb{R}^3$).

We also investigated the effect of increasing the number of layers of the model to determine how this affects accuracy and whether the model is susceptible to oversmoothing as the number of layers increases. Additionally, we implemented stochastic weight averaging (SWA) and its Gaussian variant (SWAG) to see if we could improve the accuracy further. The idea is that the latter techniques would help the model to find the optimal minimum in the last stage.

2. DATASET

We considered the QM9 dataset, a widely used benchmark in quantum chemistry and machine learning applications. This dataset contains approximately 134,000 small, stable organic molecules composed of carbon, hydrogen, oxygen, nitrogen, and fluorine (CHONF) [3, 4, 5]. Each molecule is characterized by a range of chemical properties.

3. METHODS

3.1. Preprocessing

The preprocessing of the QM9 dataset is handled directly within PaiNN. This step takes the molecular graph as input, incorporating atoms' spatial information, and returns the neighborhood of each atom i ($\mathcal{N}i$). The neighborhood consists of all atoms within a cutoff distance, a hyperparameter that defines the maximum distance for two atoms to be considered neighbors. Additionally, preprocessing computes the relative distances between neighboring atoms (rij).

To optimize computational efficiency, we designed data structures that minimize computational strain. Automated testing was conducted to ensure correctness of the preprocessing pipeline. The processed molecular data is structured in a format interpretable by the model's architecture described in the next section.

3.2. Model Architecture

The PaiNN model has an advanced design to encode chemical and molecular properties by respecting the symmetries of molecular systems, such as rotational equivariance and permutation invariance of atomic indices. It incorporates inductive biases to predict energy contributions from atoms embedded in their chemical environments, making it suitable for applications in computational chemistry and materials science.

The architecture is specifically relevant to predict scalar and vector properties, capturing the interplay between atomic positions and features within molecular structures.

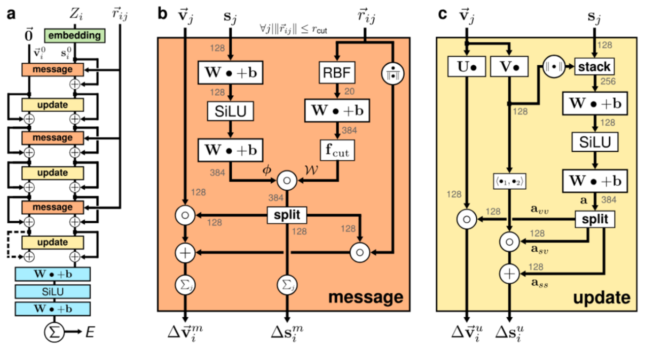

Fig. 1: PaiNN architecture representation

On a High-Level overview, The PaINN model operates on a residual structure composed of alternating message-passing and update layers, as depicted in Fig. 1. This residual framework facilitates efficient learning by refining atom-wise scalar and vector representations iteratively. The model inputs include the nuclear charges, 3D atomic positions, and the local neighborhood for each atom. The outputs are scalar-valued atomic contributions that can be aggregated for various target properties, such as the internal energy at 0K.

3.2.1. Message Layer

The message-passing block is responsible for aggregating information from the neighborhood of each atom to capture local molecular structures. It combines scalar features (invariant atom representations), vector features (equivariant atom representations), and relative positions between atoms. This layer comprises several components:

Radial Basis Functions (RBF): RBFs are applied to the interatomic distances to create continuous filters. These filters are rotationally invariant and encode the distance-dependent interactions.

Scalar Convolution: A convolution is performed on scalar features using the RBF-expanded interatomic distances, producing updated scalar messages.

Vector Convolution: For the vector features, the layer uses an equivariant message-passing mechanism. This includes scaling the directional vectors by the learned scalar filters. The process ensures that vector updates preserve the geometric directional dependencies inherent in molecular interactions.

Split Functionality: Outputs from the scalar and vector convolutions are split into residuals and for scalar and vector updates, respectively, enabling efficient refinement of features.

3.2.2. Update Layer

The update layer is designed to combine contributions from the scalar and vector features obtained in the message-passing step. This layer ensures both rotational equivariance and the preservation of scalar invariance, which are critical for accurately modeling chemical and physical properties. The main components are:

Scalar Update: Scalar features are updated through transformations involving both the neighboring scalar features and interaction terms derived from the vector features. A gated mechanism integrates these contributions.

Vector Update: Vector features are refined by transformations that preserve their directional nature. These updates account for directional interactions using invariant filters and transformations. By propagating directional information across the network, the update layer maintains the equivariant structure of the vector features.

Stack Mechanism: Scalar and vector features are combined using a stacking operation that balances the invariance of scalar properties with the equivariance of vector properties. This ensures that scalar features contribute to energy or chemical property predictions, while vector features retain information about geometric dependencies.

3.2.3. Hyperparameters

The performance and flexibility of the PaINN model are influenced by several hyperparameters that were predefined to account for better model performance:

Number of Layers: The model typically employs three layers by default, corresponding to alternating message-passing and update blocks. This parameter can be adjusted to explore different network depths.

Number of Features: The dimensionality of the scalar and vector feature representations is set to 128 by default and it controls the richness of the features learned by the network.

Number of RBF Features: Refers to the dimensionality of the radial basis function encoding used to represent pairwise atomic distances .

Cutoff Radius: Defines the maximum distance within which two atoms are considered neighbors. This parameter determines the local neighborhood for each atom and influences the computational complexity of the model.

3.3. Postprocessing

The post-processing step takes atomic-wise predictions from the model PaiNN and standardizes them to ensure comparability across molecules. It involves taking the mean and standard deviation of the atom-wise contributions. After standardization, the atomic contributions are grouped by molecule (graph index) and summed within each molecular group to obtain the final graph-level molecular property, the target Internal Energy at 0K.

3.4. Layer Optimization

We investigate the impact of varying the number of message-passing layers in PaiNN: 1, 3, 5, and 10 layers. The number of layers determines how deep the network is and affects its ability to capture interactions between atoms. Increasing the number of layers improves model's performance, but in Graph Neural Networks specifically, can lead to a phenomenon known as over-smoothing.

Over-smoothing occurs when the nodes of a graph become indistinguishable, with too many layers, by loss of meaningful local information. To optimize performance it is necessary to understand the trade-off between model depth and over-smoothing.

3.5. Bayesian Modelling

We explored bayesian modelling techniques that lead to more robust models and estimates of uncertainty. We implemented SWA [6] and SWAG [7].

3.5.1. Stochastic Weight Averaging (SWA)

SWA is a technique used in training deep neural networks. It involves collecting the weights through training and, instead of converging to a single set of weights, it averages them. By averaging the weights, the model finds a flatter region in the loss surface, which is associated with better generalization and more robust predictions, by stabilizing the final weight values. Typically, in SWA, SGD (Stochastic Gradient Descent) is run with a constant learning rate schedule starting from a pre-trained solution ([7]) and continues training for additional epochs, during which weight averaging is performed.

3.5.2. Stochastic Weight Averaging-Gaussian (SWAG)

SWAG extends SWA by modeling the uncertainty in the weights as a Gaussian distribution. During training, SWAG captures the covariance of the weights, allowing for the approximation of the posterior distribution of model weights. This enables uncertainty estimation in predictions, an important aspect in Bayesian deep learning. Similar to SWA, it leads to better generalization and more robust predictions with reliable uncertainty estimates.

At the end of training, SWAG computes both the mean and covariance of the weights. Using this posterior distribution, multiple weight samples are drawn and Bayesian averaging is used to make predictions [7].

4. RESULTS AND DISCUSSION

To evaluate the performance of our model implementation, we tested the original PaiNN model and its extensions, focusing on their ability to predict the target variable, internal energy at 0 K. The primary objective was to assess the Mean Absolute Error (MAE) under different configurations and learning strategies.

The PaiNN models, with varying numbers of layers but without SWA or SWAG, were trained and tested four times. We followed the same parameters and training techniques described by Schütt et al. [2]. Specifically, the models were trained using a batch size of 100, an initial learning rate of 5×10^{-4}, and learning rate scheduling that reduced the learning rate by a factor of 0.5 whenever the validation loss plateaued, with a patience of 5 epochs and an early stopping patience of 30 epochs.

As a baseline, the original PaiNN implementation with 3 layers achieved a mean MAE of 7.9850, as noted on Table 1.

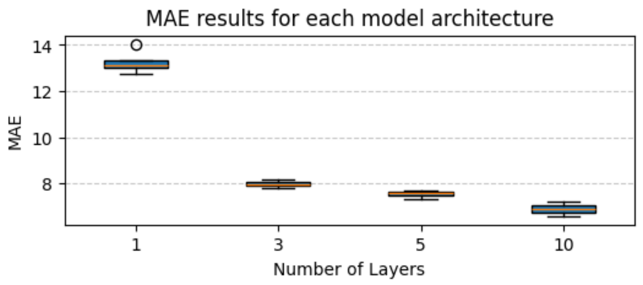

Fig. 2: Comparison of the test MAE for PaiNN models with varying numbers of layers

From the boxplot in Figure 2 we can see that there was not significant variation between the test results across models. Comparing the model configurations, the model with only 1 layer performed considerably worse, while the ones with 5 and 10 achieved slightly better results that the original model. However, despite the 10-layer model having the lowest MAE, increasing the number of layers also introduced a significant increase in complexity, taking more than twice the time to train and test than the 3-layer model. Given that the improvement was marginal while the computational cost seemed much higher, we decided to focus on the original PaiNN model for further analysis.

It is also worth mentioning that, as noted previously, we expected to observe over-smoothing as the number of layers increased. However, we observed lower MAEs as the number of layers increased. After some investigation, we concluded that the architecture of PaiNN effectively mitigates over-smoothing. The model's residual connections [1], for example, allow it to bypass certain layers, which helps preserve the network's original data and keeps features from becoming overly identical after several message-passing steps. Additionally, by separating scalar and vector features, PaiNN stops them from blending too much during message passing, preserving feature distinctiveness. PaiNN's localized update mechanism and the use of the RBF kernel with a cutoff further facilitate this by ensuring that information from distant nodes or atoms in the graph does not excessively influence one another. By focusing on neighboring nodes during message passing and limiting the influence of distant atoms, the model helps maintain distinct features across layers, preventing over-smoothing. Together, these elements help preserve feature diversity across layers, preventing over-smoothing even with increased model depth.

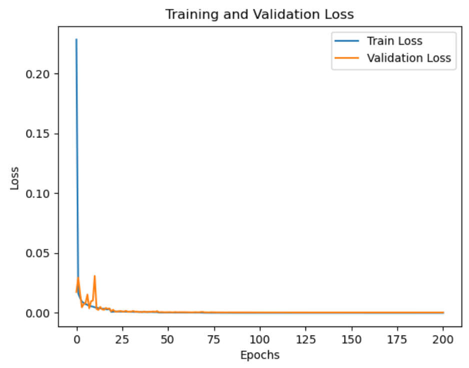

Fig. 3: Training and Validation Loss for the best 3-layer PaiNN model

From Figure 3, both training and validation losses drop quickly in early epochs before stabilizing near zero. This highlights the effectiveness of early stopping in preventing overfitting and learning rate decay in ensuring efficient convergence and generalization.

Regarding the SWA and SWAG extensions, we trained and tested each configuration starting from the pre-trained best 3-layer PaiNN model. Training and testing were performed once for each learning rate from a set of 16 values $(10^{-10}, 5 \times 10^{-9}, ..., 10^-3)$, without early stopping or learning rate scheduling, over 50 epochs.

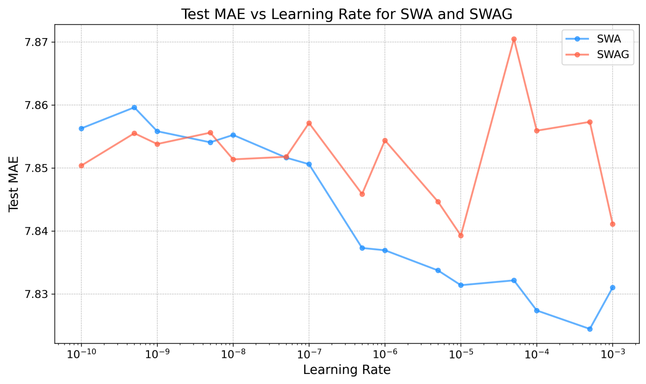

Fig. 4: Test MAE for SWA and SWAG extensions with varying learning rates

The best MAE was obtained for a learning rate of 5×10^-4 for the SWA extension and 10^-5 for the SWAG one, as shown in Figure 4. These best models represent an improvement of 2% and 1.8% from the PaiNN model with 3 layers, as summarized in Table 1.

Although the SWA and SWAG achieved improvements over the original 3-layer PaiNN model, their performance remained worse than that of the 5-layer and 10-layer ones. However, these extensions rely on weight averaging techniques, which do not significantly increase the model's training complexity compared to training deeper networks. This makes them an attractive option for scenarios where computational resources or training time are a constraint.

Table 1: MAE for different PaiNN architectures on the test set

Model	Mean MAE (meV)
PaiNN3 (original model)	7.9850
PaiNN1	13.2553
PaiNN5	7.5428
PaiNN10	6.9043
Extension	**MAE (meV)**
PaiNN3-SWA (lr=5e-4)	7.8244
PaiNN3-SWAG (lr=1e-5)	7.8393

5. CONCLUSION

In this project, the original PaiNN model and its extensions were successfully implemented and evaluated, achieving low MAEs in the prediction of molecular internal energy at 0K, comparable to those obtained in the original paper[2](5.98). Our results show that increasing the model's number of message-passing layers can improve predictive performance. Nevertheless, this improvement comes at the cost of significantly increased computational complexity.

Results from the use of SWA and SWAG extensions were positive, providing a lower-weight substitute for improving performance without deepening the model.

Future work could include more training and testing of the various model configurations to gain a larger range of outcomes and increase performance evaluation accuracy. The depth of this investigation was limited by a lack of computational resources, as DTU's HPC system was being widely used, resulting in delays in running experiments. Further investigation of model depth and SWA/SWAG setups may provide additional insights and potentially an improvement in the model's performance. Furthermore, expanding the analysis to predict numerous chemical attributes would put the model's adaptability to the test and broaden its possible uses.

Overall, the PaiNN model and its extensions offer a useful method for forecasting molecular characteristics, which can help in the search for new materials and enhance simulation accuracy, leading to improvements in areas like drug development, energy storage and sustainable materials.

6. REFERENCES

[1] William L. Hamilton. *Graph Representation Learning*. Morgan & Claypool Publishers, 2020. Chapters 5 and 6.

[2] Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International Conference on Machine Learning*, pages 9377–9388. PMLR, 2021.

[3] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

[4] Lars Ruddigkeit, Ruud Van Deursen, Lorenz C Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of chemical information and modeling*, 52(11):2864–2875, 2012.

[5] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay S. Pande. Moleculenet: A benchmark for molecular machine learning. *CoRR*, abs/1703.00564, 2017.

[6] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima in deep learning. *arXiv preprint arXiv:1803.05407*, 1803:05407, 2018.

[7] Timur D Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in Neural Information Processing Systems*, 31, 2018.

Appendix

The code of the project described in this report is documented in the following GitHub repository: `https://github.com/mafpi/02456_Project_PaiNN`