

Atomic partial charge prediction with graph neural networks

David Parker
Dr. Cassiano Langini

May 31, 2023

Outline

1 Introduction

2 Goal

3 Methods

4 Results

5 Discussion

6 Conclusion

Introduction

Atomic partial charge

Coulomb's law:

$$|F| = k_e \frac{|q_1||q_2|}{r^2} \quad (1)$$

Where

- Coulomb constant: $k_e = 9 \cdot 10^9 \frac{\text{N} \cdot \text{m}^2}{\text{C}^2}$
- q_1 and q_2 are the charges in Coulombs (C)
- r is the distance between the charges in meters (m)

Atomic partial charge

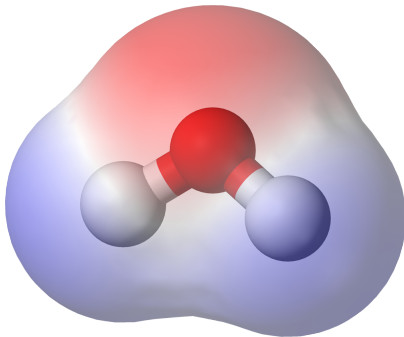


Figure: Representation of the electronic charge distribution of a water molecule. The oxygen atom has a partial negative charge, while the hydrogen atoms have partial positive charges[2].

Partial charge calculation

Experimentally (Spectroscopy):

- X-ray diffraction, Nuclear magnetic resonance, UV-Vis, etc

Computationally: Tradeoffs speed VS accuracy

- Quantum mechanics calculations: Evolution of a system through time

$$i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle = \hat{H} |\psi(t)\rangle \quad F = \frac{\partial}{\partial t} p$$

Quantum counterpart of Newton's second law of motion

- Force fields: Calculations based on a set of parameters

Goal

Create a dataset of molecules and use the Charm General Force Field (CGenFF) to calculate partial charges for each atom

Train a graph neural network to predict atomic partial charges

Methods

Can also be represented
as a graph:

- Nodes with attributes
- Edges with attributes
- Global attributes

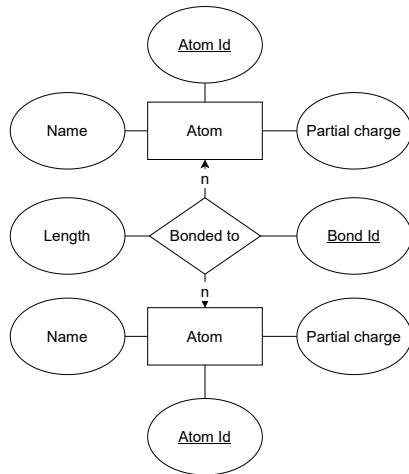


Figure: Data model for the mini-world

Molecular Dataset: mol2 file

```
@<TRIPOS>MOLECULE
0002bf2c5ea40c3014dbcc0182ea4598
50 52
SMALL
USER_CHARGES
CgenFF charge penalty mean/max 2.42094 / 30.403
@<TRIPOS>ATOM
  1 N1      0.1212   0.064  -0.3102 N.pl3  0 0   -0.163
  2 N2      4.9006  -0.7556   0.5231 N.4   0 0   -0.426
  [...]
48 H48      6.451  -0.6653  -0.8628 H   0 0    0.090
49 H49      5.9904  -2.2808  -0.3819 H   0 0    0.090
50 H50      6.9083  -1.2723   0.7106 H   0 0    0.090
@<TRIPOS>BOND
  1 1 3 1
  2 1 10 1
  [...]
51 21 49 1
52 21 50 1
@<TRIPOS>MOLECULE
24d10877593ae175c68f8d409e6df5ae
26 25
```

Figure: Example of a mol2 file. Each molecule begins with a `@ < TRIPOS > MOLECULE` tag, then the different fields are separated by their own tags.

Molecular Dataset: Molecules

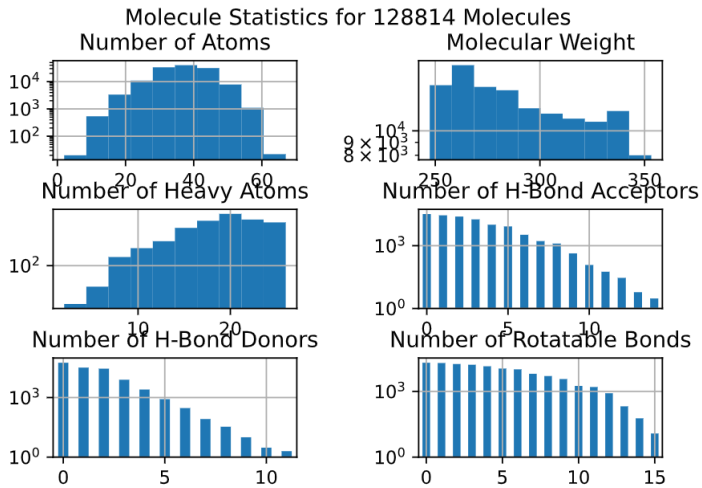


Figure: Distributions of different molecular properties in the dataset.

Molecular Dataset: Atoms

Atom Statistics for 128814 molecules with 4720914 atoms total

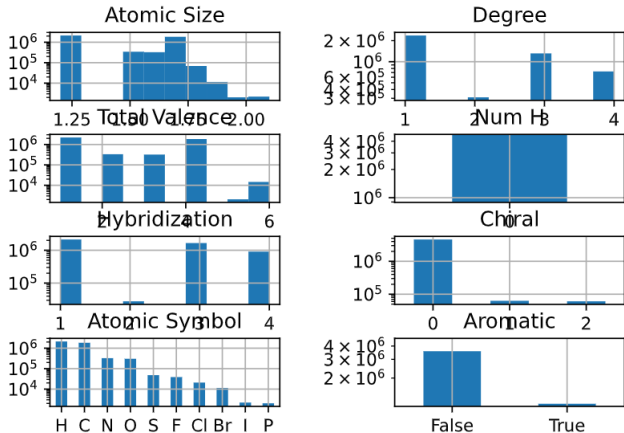


Figure: Distributions of different atomic properties in the dataset.

Message-passing neural network

$$\begin{aligned} \mathbf{e}'_k &= \phi^e(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}) & \bar{\mathbf{e}}'_i &= \rho^{e \rightarrow v}(E'_i) \\ \mathbf{v}'_i &= \phi^v(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u}) & \bar{\mathbf{e}}' &= \rho^{e \rightarrow u}(E') \\ \mathbf{u}' &= \phi^u(\bar{\mathbf{e}}', \bar{\mathbf{v}}', \mathbf{u}) & \bar{\mathbf{v}}' &= \rho^{v \rightarrow u}(V') \end{aligned}$$

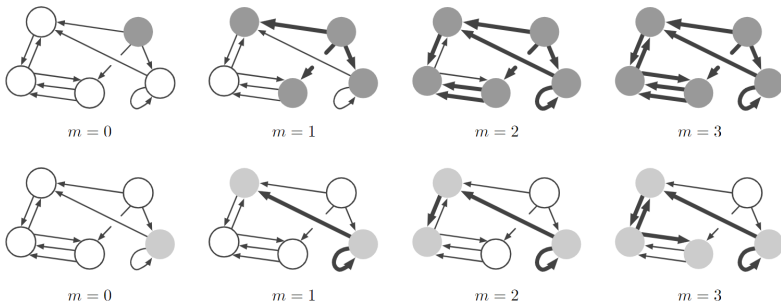


Figure: Message-passing neural network. Three update functions ϕ and three aggregation functions ρ are used to update the edge, node and global attributes[1].

Message-passing neural network

Node features: Element type

- Preprocess layer: 2 linear layers with batch normalization and ReLU activation functions to embed the element type into an internal continuous representation.
- Edge function layer: several linear layers with batch normalization and ReLU activation functions to compute edge features based on the node features of the adjacent nodes.
- GNN layer: applies a convolution operation to update the node features using the edge features as weights.
- GRU layer: applies a gated recurrent unit to update the node features.
- Predict layer: 2 linear layers with batch normalization and ReLU activation functions, followed by a linear layer with a single output.

Results

Benchmarks

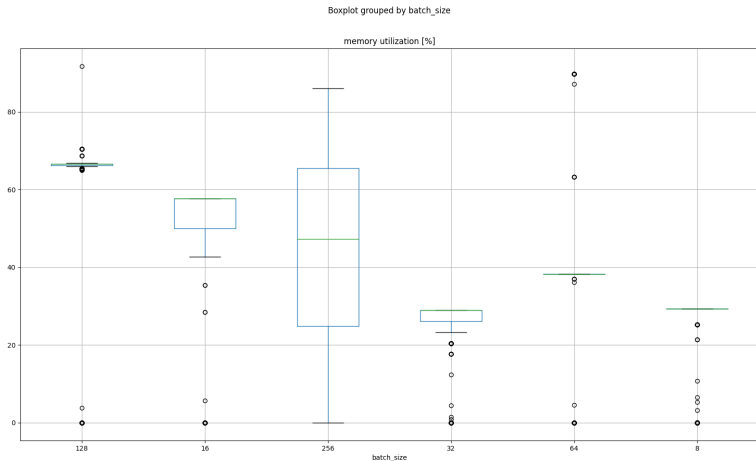


Figure: Benchmarks of GPU utilisation for different batch sizes.

Absolute errors

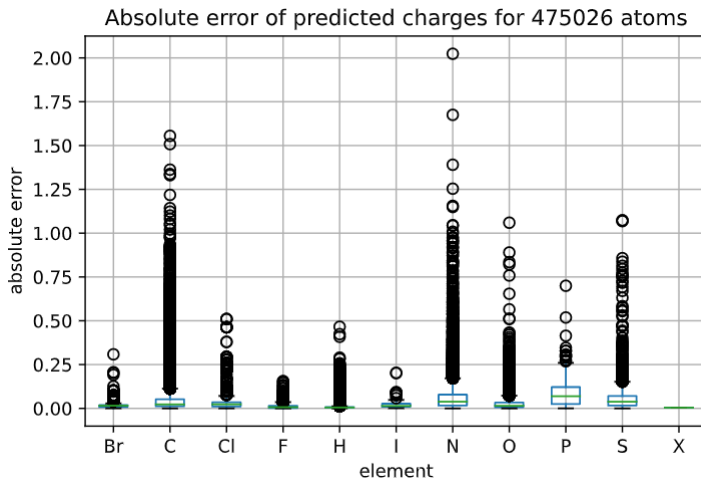


Figure: Distributions of the absolute errors of the predicted partial charges for each type of atom.

Root mean squared error

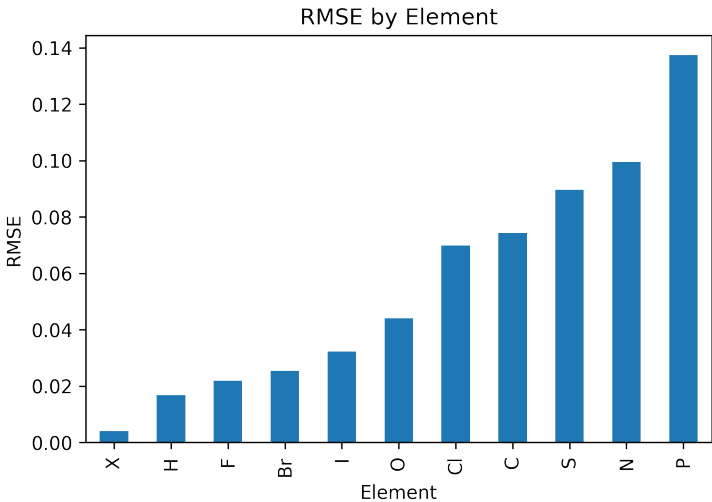


Figure: Root mean squared error for each type of atom.

Mean absolute error

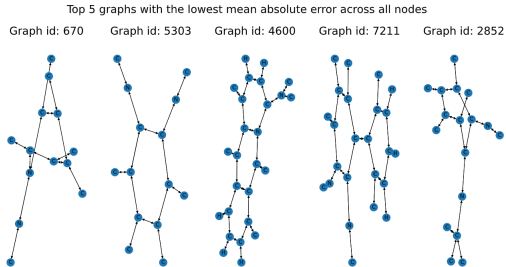
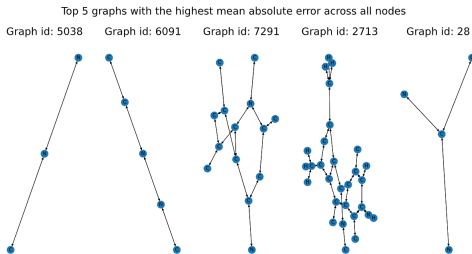
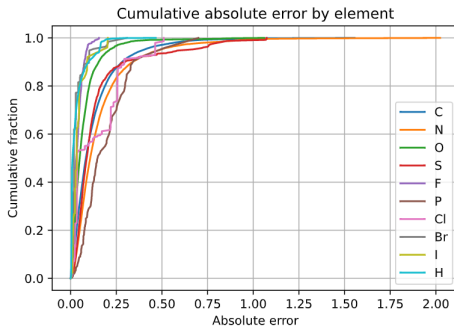


Figure: Visualizations of the graphs with the highest and lowest MAE.

Our results compared to the literature



Patrick Bleiziffer, Kay Schaller, and Sereina Riniker

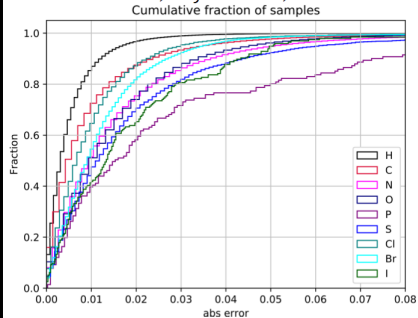


Figure: Cumulative fraction of each type of atom as a function of the absolute error. Our results on the left, the results of the literature on the right[3].

Discussion

Dataset generation

- The dataset is technically easy to generate:
 - SQL queries to extract the data from the database
 - Use a force field to compute the partial charges
- The hard part is to choose relevant molecules:
 - We chose drug-like molecules. Molecular weight around 300 g/mol, specific number of rotatable bonds, etc. (see Figure 4)

- Choice of the model:
 - We went with a MPNN as introduced by Gilmer et al.[4]
- Achitecture of the model:
 - We would like to add physical constraints to the model, so that the predicted partial charges make more sense

Conclusion

- We have generated a dataset of 8000 molecules with partial charges
- We have trained a MPNN to predict the partial charges
- The model has a RMSE of 0.054
 - Elementary charge of a proton: 1.6×10^{-19} C

Acknowledgements

Thanks to Cassiano and Yi for the help and interesting discussions

A big thank you to Kurt and all of you for an exciting lecture!

References

- [1] Peter W. Battaglia et al. *Relational inductive biases, deep learning, and graph networks*. 2018. arXiv: 1806.01261 [cs.LG].
- [2] Benjah-bmm27. *Water*. URL: <https://commons.wikimedia.org/w/index.php?curid=1498405>. (accessed: 2023-05-29).
- [3] Patrick Bleiziffer, Kay Schaller, and Sereina Riniker. “Machine Learning of Partial Charges Derived from High-Quality Quantum-Mechanical Calculations”. In: *Journal of Chemical Information and Modeling* 58.3 (2018). PMID: 29461814, pp. 579–590. DOI: 10.1021/acs.jcim.7b00663. eprint: <https://doi.org/10.1021/acs.jcim.7b00663>. URL: <https://doi.org/10.1021/acs.jcim.7b00663>.
- [4] Justin Gilmer et al. “Neural Message Passing for Quantum Chemistry”. In: *CoRR* abs/1704.01212 (2017). arXiv: 1704.01212. URL: <http://arxiv.org/abs/1704.01212>.