Graph neural network potentials for molecular dynamics simulations of water cluster anions

Alfonso Gijón[1*][0000-0002-7400-8182], Miguel Molina-Solana[1,2][0000-0001-5688-2039], and Juan Gómez-Romero[1][0000-0003-0439-3692]

[1] Dept. of Computer Science and AI, University of Granada, Spain
[2] Dept. of Computing, Imperial College London, United Kindom
`* Corresponding author: alfonso.gijon@ugr.es`

Abstract. Regression of potential energy functions is one of the most popular applications of machine learning within the field of materials simulation since it would allow accelerating molecular dynamics simulations. Recently, graph-based architectures have been proven to be especially suitable for molecular systems. However, the construction of robust and transferable potentials, resulting in stable dynamical trajectories, still needs to be researched. In this work, we design and compare several neural architectures with different graph convolutional layers to predict the energy of water cluster anions, a system of fundamental interest in chemistry and biology. After identifying the best aggregation procedures for this problem, we have obtained accurate, fast-evaluated and easy-to-implement graph neural network models which could be employed in dynamical simulations in the future.

Keywords: Graph Neural Networks · Atomic Potentials · Molecular Dynamics · Message Passing · Convolutional layers

1 Introduction

Calculating macroscopic properties (mechanical, thermodynamic, electronic...) of materials through simulation of their microscopic components is a very active field with high impact in science and engineering. This process is usually done employing statistical physics by taking averages of the instantaneous properties on the phase space (in which all possible states of a system are represented), sampled with a suitable distribution. One of the most powerful and widely used technique to sample the phase space and obtain dynamical trajectories is molecular dynamics (MD). In a few words, MD is an algorithm that receives an initial state as input and produces an output state at the following time-step by solving the motion equations. Another popular approach is Monte Carlo (MC), where the phase space is sampled by generating random moves which are accepted or rejected following the Metropolis algorithm. In both MC and MD methods, the potential energy function describes the underlying interactions of the system and is a key ingredient for the quality of a simulation. The most time-consuming

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_25

part of a simulation is the evaluation of the energy for MC and both energy and forces for MD. Forces acting on each atom are computed as derivatives of the potential energy with respect to the atomic positions and are necessary to update the state. Accurate first principles potentials require significant computational resources and it is challenging for simulations to converge when the system size increases.

Different machine learning (ML) architectures have been proposed in recent years as a powerful tool for predicting material properties and molecular interactions [9,2]. In particular, graph neural networks (GNNs) have shown to be especially helpful for evaluating energy and forces of molecular systems, which can reduce the computational cost of molecular dynamics by multiple orders of magnitude [6,7]. Graphs are non-Euclidean data structures able to model systems with complex interactions, such as molecules, as well as respect symmetries like permutational, rotational and translational invariance.

Although the accuracy of complex architectures for predicting static properties has been extensively proven on benchmark datasets [11,6], the robustness of GNN potentials when employed in real MD simulations has still to be improved [10]. Therefore, further investigations are necessary to obtain stable and transferable potentials based on GNNs. Flexible and complex architectures equipped with physics constraints could be a good direction to explore in future work.

In this work, we analyze the performance of different GNN architectures on a dataset of water cluster anions, with chemical formula $(H_2O)_N^-$. This system has been intensively investigated as model for the hydrated electron in bulk water, a species of fundamental interest, which is involved in important chemical and biological electron transfer processes [1,3]. However, despite the huge experimental and theoretical effort made to understand how the excess electron or other negatively-charge species are bound to water clusters [8], there are still open questions.

This paper is organized as follows. The system under study is presented in section 2, together with the basis of graph neural networks. Some methodological details are explained in section 3, while the main results are exposed in section 4. Lastly, some conclusions and possible future work are pointed out in section 5.

2 Background

The properties of a material are determined by its underlying interactions, which can be codified into the potential energy function $V(\mathbf{r}_1, \ldots, \mathbf{r}_N)$, a quantity dependent on the atomic position coordinates $\mathbf{r} = (x, y, z)$. The knowledge of the forces acting on each atom (computed as derivatives of the potential energy function with respect to the atomic positions) of the system would allow to numerically integrate the motion equations and obtain the updated positions. If this process is repeated iteratively, one generates a dynamical trajectory of the system which can be employed to compute macroscopic properties.

In water cluster anions, the excess electron is not attached to a specific water molecule, but is bound globally to several water molecules or to the whole clus-

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_25

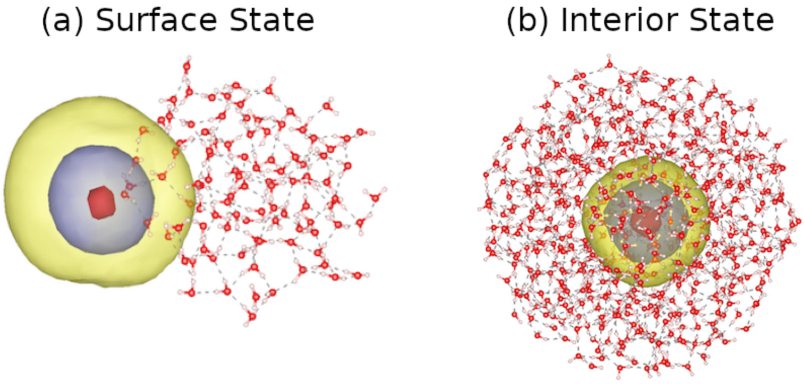

Fig. 1. Schemes of surface (a) and interior (b) states of the excess electron of two water clusters anions.

ter. Depending on the position of the excess electron we can distinguish surface and interior states, see Figure 1. The exact description of the interactions is unaffordable, but is not necessary for our purposes. The neutral water molecules are approximated as classical entities interacting through a SPC/F effective potential, while the excess electron is represented by its wave function as a quantum particle living in the pseudopotential generated by the atoms [1,3]. This way, the total potential energy can be considered as the sum of two terms, the first one describing the atomic interactions and the second one taking account the electronic contribution of the excess electron:

$$E(\mathbf{r}_1,\ldots,\mathbf{r}_N) = V_{\mathrm{SPC/F}}(\mathbf{r}_1,\ldots,\mathbf{r}_N) + E_0(\mathbf{r}_1,\ldots,\mathbf{r}_N). \quad (1)$$

While the atomic term is evaluated very fast, the electronic contribution involves the numerical resolution of a 3D single-particle Schrödinger equation[3], which is time-expensive and frequently becomes a bottleneck to converge MD simulations and therefore to compute macroscopic properties.

Graphs serve to model a set of objects (nodes) and their interactions (edges). When applied to molecules, each atom is considered a node and edges are usually assigned to chemical bonds. Both nodes and edges are defined by their features, $\mathbf{x}_i$ and $\mathbf{e}_{i,j}$, respectively. Deep learning (DL) methods can be applied to graph-structured data to perform classification and regression tasks, among others, to give rise to graph neural networks, able to learn about local relations among nodes through the exchange of messages carrying information on the environment of each node [5]. A convolutional (Conv) or message passing (MP) layer

[3] The 3D one-particle Schrödinger equation yields:

$$H\psi_e(\boldsymbol{r}_1,\ldots,\boldsymbol{r}_N;\boldsymbol{r}_e) = E_0\psi_e(\boldsymbol{r}_1,\ldots,\boldsymbol{r}_N;\boldsymbol{r}_e),$$

where the hamiltonian operator is $H = -\nabla^2/2m_e + V_{\mathrm{W}-e}(\boldsymbol{r}_1,\ldots,\boldsymbol{r}_N;\boldsymbol{r}_e)$.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_25

updates the node features taking into account information from neighbor nodes and edges:

$$\mathbf{x}'_i = \texttt{UPDATE}\left[\mathbf{x}_i, \texttt{AGGREGATE}_{j\in\mathcal{N}(i)}\left(\mathbf{x}_i, \mathbf{x}_j, \mathbf{e}_{ij}\right)\right] \tag{2}$$

$$= \gamma\left(\mathbf{x}_i, \square_{j\in\mathcal{N}(i)}\,\phi\left(\mathbf{x}_i, \mathbf{x}_j, \mathbf{e}_{ij}\right)\right) = \mathrm{Conv}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{e}_{ij}). \tag{3}$$

Symbol $\square$ denotes a differentiable, permutation invariant function (sum, mean or max), γ and ϕ represent differentiable functions such as MLP (multilayer perceptron) or a different operation involving node and edge features, and $\mathcal{N}(i)$ represents the set of nodes connected to node i. There are plenty of different convolutional layers which could be used to construct a GNN model, so we perform a systematic study of six of them in next section. Apart from the architecture of a GNN, the embedding process is crucial for the goodness of a model. That is, the encoding of the original data into a graph structure should capture the important features and relations among nodes.

3 Methods

We employ a database of water cluster structures with size ranging from 20 to 237 water molecules. Each element of the database consists of the Cartesian position coordinates of all atoms composing the cluster, along with the atomic and electronic contributions of the total energy associated to that geometry. The geometries are extracted from previous MD equilibrium simulations at 50, 100, 150 and 200 K of temperature, carried out with a software available at https://github.com/alfonsogijon/WaterClusters_PIMD. Finally, the database is formed by 1280 cluster geometries, splitted in 80 % for training and 20 % for testing purposes. Throughout this work, we employ atomic units for energy and distance, unless the opposite is explicitly said.

Each cluster geometry is encoded into a graph structure, where nodes represent atoms and edges represent chemical bonds. Each node contains 4 features, describing the chemical properties of each atom: the first one identifies the atom type through its atomic number (8 for oxygen and 1 for hydrogen) and the 3 remaining features correspond to the atom's position, that is, $\mathbf{x}_i = (Z_i, x_i, y_i, z_i)$. Only one edge feature is considered, as the distance between two connected atoms:

$$e_{ij} = \begin{cases} r_{ij} & i \,\text{and}\, j \,\text{are connected,} \\ 0 & \text{elsewhere.} \end{cases} \tag{4}$$

Each atom is connected to all atoms inside a cutoff sphere (if $r_{ij} < r_c$). The cutoff radius is a tunable parameter, which is set to 6, which implies an average number of 5 neighbor or connected atoms. The cutoff r_c was chosen as the minimum value so that a bigger value does not affect significantly the accuracy of the model. Together with the edge features, for some convolutional layers is

DOI: 10.1007/978-3-031-36027-5_25

convenient to define a binary adjacency matrix, non-zero when two atoms are connected and zero elsewhere:

$$A_{ij} = \begin{cases} 1 & i \,\text{and}\, j \,\text{are connected,} \\ 0 & \text{elsewhere.} \end{cases} \tag{5}$$

Our GNN models receive a graph as input and provide a value of the energy as output. The model architecture is summarized in 3 steps:

(i) 2 convolutional layers:

$$\mathbf{x}_i^{k+1} = \text{Conv}\left(\mathbf{x}_i^k, \mathbf{x}_j^k, \mathbf{e}_{ij}^k\right)_{j \in \mathcal{N}(i)}.$$

(ii) Global sum pooling:

$$\mathbf{x}^F = \text{GlobalSumPool}\left(\mathbf{x}_1^2, \ldots, \mathbf{x}_N^2\right) = \sum_{i=1}^{N} \mathbf{x}_i^2.$$

(iii) 2 fully connected layers:

$$E = \text{NN}\left(\mathbf{x}^{\text{F}}\right).$$

First, two convolution layers are applied to the node features, which are updated to account for each node environment. Then, the node features are pooled into a final vector of node features. Finally, a fully connected neural network with 2 hidden layers predicts the energy from the final feature vector. We fixed the number of convolutional and hidden layers to 2 because more layers do not produce an important improvement in the final performance. For each hidden layer of the dense neural network, we used 128 units.

In the results section, the performance of several available types of convolutional layers is tested, while keeping constant the rest of the architecture. To define and train the graph-models the Spektral software [4] was employed, an open-source project freely available on https://graphneural.network/.

4 Results and Discussion

To analyze the performance of different convolutional layers for the energy regression task, up to 6 types of Conv layers were used, namely CrystalConv, GATConv, GCSConv, ECCConv, AGNNCong and GeneralConv. Specific details on each architecture can be consulted on the official Spektral website. For each architecture, two different networks were used to predict each contribution to the total energy, remind Equation 1. Each model was trained until the loss function (set as MAE) converged to a constant value, which typically occurred between 100 and 150 epochs.

The accuracy of the models on the test dataset is shown in Table 1. As can be seen observing the MAPE columns, the regression of the atomic energy is better than the electronic contribution. This was expected because the atomic term

	Atomic Energy			Electronic Energy		
	MAE (Ha)	MSE (Ha²)	MAPE (%)	MAE (Ha)	MSE (Ha²)	MAPE (%)
CrystalConv	0.10	0.0232	8.0	0.110	0.06571	101.8
GATConv	0.02	0.0008	1.6	0.005	0.00005	7.6
GCSConv	0.04	0.0027	2.9	0.007	0.00007	10.2
ECCConv	0.25	0.1569	16.1	0.018	0.00053	36.5
AGNNConv	0.06	0.0063	4.2	0.017	0.00059	23.9
GeneralConv	0.03	0.0015	2.2	0.014	0.00028	26.9

Table 1. Accuracy of different models with respect to different metrics.

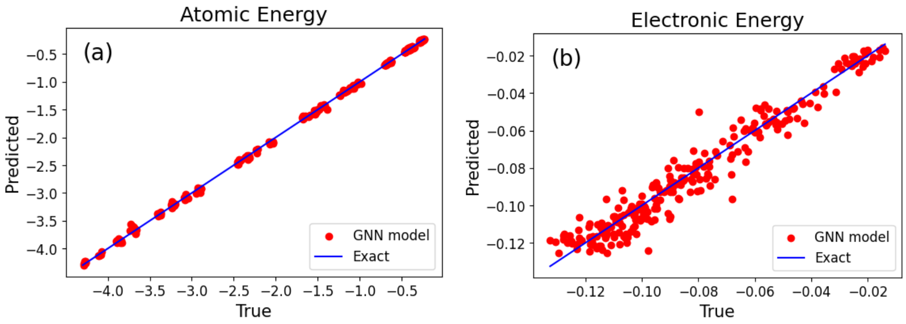

Fig. 2. Comparison between predicted and true energies for (a) atomic and (b) electronic contributions on the test dataset, for the GATConv model.

comes from a simple potential (the SPC/F model is sum of two-body terms) while the electronic term has a more complex nature, such as a Schrödinger equation. Comparing the convolutional layers, GATConv is the best architecture, obtainig a MAPE of 1.6 % for the atomic contribution and 7.6 % for the electronic one. The energy predictions for that architecture can be visualized in Figure 2, where predicted energies are compared to the true values.

GATConv layer uses an attention mechanism to weight the adjacency matrix that seems to learn very well different connections of each node (O-O, O-H, H-H bonds). It is remarkable that CrystalConv layer, specially designed for material properties [11], does yield poor results, but this can be explained by the fact that our system is finite as we are working with clusters, and CrystalConv layer was designed to represent crystal, that is, infinite periodical systems. As a matter of fact, only GATConv and GCSConv produces acceptable results for both the atomic and the electronic energies. The former weights the adjacency matrix and the latter has a trainable skip connection layer.

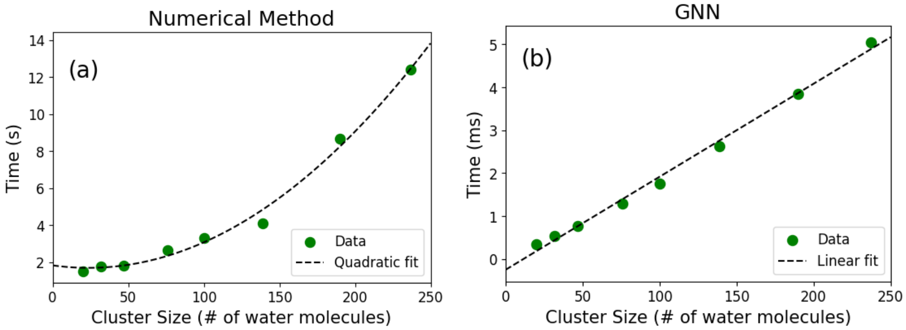

Fig. 3. Evaluation time of the total energy for the standard numerical method (a) and the optimal GNN model (b).

Regarding computational time, GNN models have enormous advantages against standard potentials. First, whilst the evaluation of the total energy is quadratic with the system size (number of water molecules) for the original numerical method, GNN models follow a linear relation, see Figure 3. Second and most important, the time necessary to evaluate the energy of a water cluster anion is in the scale of seconds for the numerical method, whereas the GNN model spend milliseconds, so that is 3 orders of magnitude faster.

5 Conclusions

As a first step, we have identified an optimum architecture to construct a GNN model able to predict accurate energies for water cluster anions. Besides, the model is easily implemented, in contrast to the numerical method needed to solve the one-electron Schrödinger equation to obtain the energy via standard numerical methods. Our GNN model is also much more faster to evaluate and could make possible to converge long simulations involving many atoms, which are necessary to compute some macroscopic properties.

Our results prove the importance of choosing an appropriate architecture, specially a suitable convolutional scheme, when constructing a GNN model for finite molecular systems as water cluster anions. GATConv and GCSConv layers yield good results and could be improved including more node features (atomic charge, atomic mass...) and edge features (type of bond, angular and dihedrical information...).

As future work, we plan to implement the calculation of the forces in a consistent way, as derivatives of the energy with respect to the atomic coordinates. This would allow to carry out molecular dynamics simulations, as long as improving the learning process and obtain more robust energy potentials.

DOI: 10.1007/978-3-031-36027-5_25

References

1. Alfonso Gijón Gijón: Classical and quantum molecular dynamics simulations of condensed aqueous systems. PhD thesis, Digital CSIC (2021), http://hdl.handle.net/10261/251865
2. Behler, J.: Four generations of high-dimensional neural network potentials. Chemical Reviews **121**(16), 10037–10072 (2021). https://doi.org/10.1021/acs.chemrev.0c00868, pMID: 33779150
3. Gijón, A., Hernández, E.R.: Quantum simulations of neutral water clusters and singly-charged water cluster anions. Phys. Chem. Chem. Phys. **24**, 14440–14451 (2022). https://doi.org/10.1039/D2CP01088G
4. Grattarola, D., Alippi, C.: Graph neural networks in tensorflow and keras with spektral [application notes]. Comp. Intell. Mag. **16**(1), 99—106 (feb 2021). https://doi.org/10.1109/MCI.2020.3039072
5. Hamilton, W.L.: Graph representation learning. Synthesis Lectures on Artificial Intelligence and Machine Learning **14**(3), 1–159
6. Klicpera, J., Becker, F., Günnemann, S.: Gemnet: Universal directional graph neural networks for molecules. In: Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems (2021), https://openreview.net/forum?id=HS_sOaxS9K-
7. Li, Z., Meidani, K., Yadav, P., Barati Farimani, A.: Graph neural networks accelerated molecular dynamics. The Journal of Chemical Physics **156**(14), 144103 (2022). https://doi.org/10.1063/5.0083060
8. Rodríguez-Segundo, R., Gijón, A., Prosmiti, R.: Quantum molecular simulations of micro-hydrated halogen anions. Phys. Chem. Chem. Phys. **24**, 14964–14974 (2022). https://doi.org/10.1039/D2CP01396G
9. Schmidt, J., Marques, M.R.G., Botti, S., Marques, M.A.L.: Recent advances and applications of machine learning in solid-state materials science. npj Computational Materials **5**(1), 83 (Aug 2019). https://doi.org/10.1038/s41524-019-0221-0
10. Stocker, S., Gasteiger, J., Becker, F., Günnemann, S., Margraf, J.T.: How robust are modern graph neural network potentials in long and hot molecular dynamics simulations? Machine Learning: Science and Technology **3**(4), 045010 (nov 2022). https://doi.org/10.1088/2632-2153/ac9955
11. Xie, T., Grossman, J.C.: Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. Phys. Rev. Lett. **120**, 145301 (Apr 2018). https://doi.org/10.1103/PhysRevLett.120.145301

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_25