

# Predicting Equilibrium Geometries of Large Multicomponent Systems with Neural Networks

Mykhailo Moroz, Oleg Bovhyra, Volodymyr Franiv, Viktor Dzikovskyi

Faculty of Physics  
Ivan Franko National University of Lviv  
Lviv, Ukraine  
oleh.bovhyra@lnu.edu.ua

**Abstract**—A neural network force approximation model for large atomic systems was suggested, practical use cases were pointed out. This model was trained on ZnO cluster examples calculated using the density functional theory (DFT). Training results were analyzed and compared to DFT results.

**Index Terms**—neural network, interatomic force, empirical approximation, multicomponent atomic systems.

## I. INTRODUCTION

The calculation of equilibrium states of multicomponent systems is heavily dependent on the density functional theory (DFT) to accurately find the force approximation of a given system, but this approach is computationally very demanding, thus increasing the popularity of empirical approximations, which can calculate force or energy approximations orders of magnitude faster. Constructing such approximation can be a tricky task, but in 2007 [1] an approach based on neural networks (NN) was suggested. NN are known for their flexibility in approximating functional relationships, therefore making them a great choice for approximating all kinds of chemical bonds. The evaluation of NN potentials is several orders of magnitude faster than DFT calculations, and its computational cost scales linearly with the simulated number of atoms.

Many research groups worldwide have focused on understanding the structure, processes of formation and properties of the ZnO nanoparticles [2-4]. Numerous theoretical studies of ZnO clusters have explored optimized geometries for a range of cluster sizes, and a prevalent theoretical observation shows that a fullerene-like structures are more stable in the case for smaller clusters, while a wurtzite structure shows increased stability for larger clusters [5-6].

In this paper we present a generalized neural network method for constructing DFT-based force approximations which have ab initio accuracy and are capable of describing all types of bonding. The resulting many-body force approximations are a function of all atomic coordinates and can be used in systems of arbitrary size and composition. We apply our ideas to the construction of an NN-based many-body force approximation for ZnO clusters.

As opposed to [1] we are approximating the force instead of the potential energy surface (PES) of the system, this removes the necessity of calculating the derivatives of the neural network and the symmetry functions, and summing the partial atomic energies is no longer needed. This choice has been

made to allow the algorithm to be more easily parallelizable in order to work with GPU's. The algorithm and the neural network code was written to work with OpenCL, which can give a performance boost up to 50 times, depending on the GPU being used, resulting in a speedup of 4 orders of magnitude, compared to DFT based approaches on CPU. This method, however, has the downside of being non-invariant over spatial rotations, it no longer is capable of finding exact equilibrium geometries (PES extrema), because the force approximation (PES gradient) can no longer be trained to be exactly zero (only up to the force approximation error) at given configurations. Therefore, requiring a more careful construction and training of the NN approximation.

## II. NEURAL NETWORK BASED MODEL FOR APPROXIMATING INTERATOMIC FORCES

Building a force approximation model based on NN requires constructing an input that represents the state of the multicomponent system, a simple approach would be to feed the Cartesian coordinates and the element numbers of the atoms directly into the neural network to output a set of force vectors (Fig. 1), this, though, has a lot of disadvantages. For example: the force approximation is non-invariant for spatial rotations and translations, the neural network can only be used for a system of a given size [7].

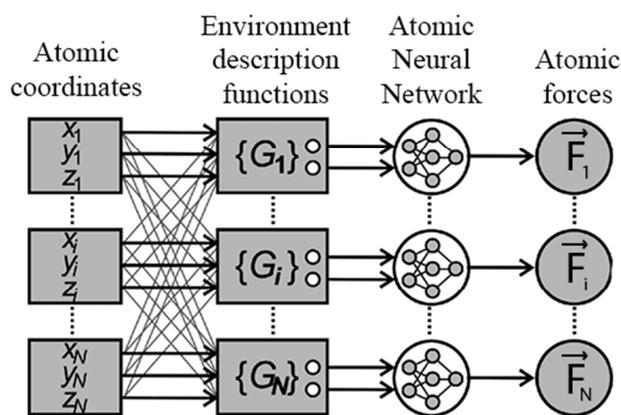


Figure 1. Schematic structure of the proposed force approximation model

A better way of constructing such a model is to calculate a limited set of functions for each atom that describes relevant information about the chemical environment of the atom, it will

be used as the input of a single atomic NN that outputs the approximated force acting on the atom. This approach was used in this paper.

#### A. Environment description functions

The construction of the environment description functions is based on this assumptions and statements:

1. The force is only dependent on the local chemical environment, usually not further than 12 Å.
2. The force is translationally invariant, but rotationally non-invariant (changes direction).
3. The force contribution is inversely proportional to the distance from the atom.
4. The description should discern between different types of atoms

Based on these assumptions we can build a function:

$$G_{ijl}^l = \sum_{k=1}^{k \neq l} m(p_k, p_l, d) \alpha_i(\vec{r}_{kl}) \beta_j(\vec{r}_{kl}). \quad (1)$$

Where  $i$  and  $j$  are the grid point indices,  $l$  is the index of our chosen atom. The sum is over all neighboring atoms in the radius of the cut-off distance.  $p_k$  is the element number of the  $k$ -th atom,  $\vec{r}_{kl}$  is the distance radius between the  $i$ -th and the  $k$ -th atom. Index  $d$  determines the chosen neighbor element number.

This function calculates an approximate density distribution on a spherical grid for each neighboring atom type (Fig. 2).

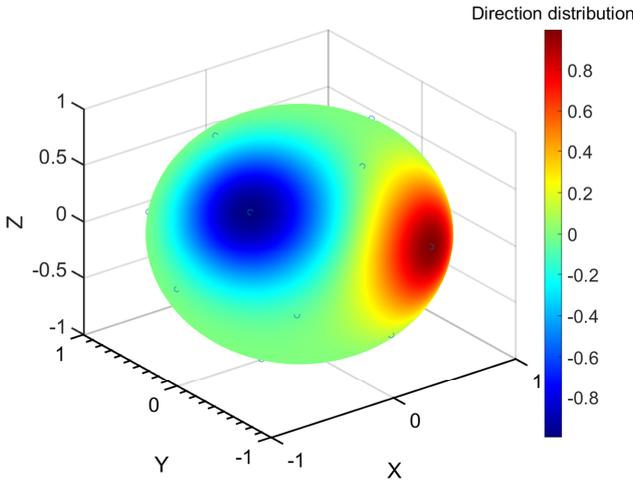


Figure 2. Uniformly distributed points on the sphere, and the difference between  $\alpha_i(\vec{r}_{kl})$  for two points

The spherical grid is constructed from a set of uniformly distributed points on a sphere, placing them on spheres of radiuses up to the cut-off distance. In this case the radiuses of the spheres were uniformly distributed.  $\vec{s}_i$  is the radius vector of the  $i$ -th uniformly distributed point on the unit sphere, or in other words, the angular resolution.  $M$  is the number of the

points on one sphere,  $P$  is the number of concentric spheres, or the radial resolution. The grid used in the ZnO tests is shown on Fig. 3.

$$\alpha_i(\vec{r}_{kl}) = \exp\left(-\left(\frac{\sqrt{M}}{\pi} \arccos\left(\frac{\vec{s}_i \cdot \vec{r}_{kl}}{|\vec{r}_{kl}|}\right)\right)^2\right). \quad (2)$$

Function  $\alpha_i(\vec{r}_{kl})$  is the angular coefficient of the  $i$ -th point. It's for is chosen in such a way, so that for points further than the closest neighbor point it will be close to zero.

$$m(p_k, p_l, d) = \begin{cases} 1, & d = p_k = p_l, \\ -1, & d = p_k \neq p_l \\ 0, & p_k \neq d \end{cases} \quad (3)$$

Function  $m(p_k, p_l, d)$  determines if the neighbor atom is of the type  $d$  and also encodes the information about if the chosen atom type is the same as the neighbor type.

$$\beta_j(\vec{r}_{kl}) = \exp\left(-\left(\frac{P}{2}\left(|\vec{r}_{kl}| - \frac{jr_{cut}}{P}\right)\right)^2\right) \gamma(\vec{r}_{kl}). \quad (4)$$

Function  $\beta_j(\vec{r}_{kl})$  is the radial distribution function, it is close to zero for the neighboring radial points.

$$\gamma(\vec{r}_{kl}) = \begin{cases} \frac{1}{2} \left( \cos\left(\frac{\pi |\vec{r}_{kl}|}{r_{cut}}\right) + 1 \right), & |\vec{r}_{kl}| < r_{cut} \\ 0, & |\vec{r}_{kl}| \geq r_{cut} \end{cases} \quad (5)$$

$\gamma(\vec{r}_{kl})$  is the cut-off coefficient, it makes the force approximation continuous for the neighbor atoms that are on the border of the cut-off sphere. This allows the Hessian matrix (i.e. the gradient of the force) to be properly calculated, if needed.

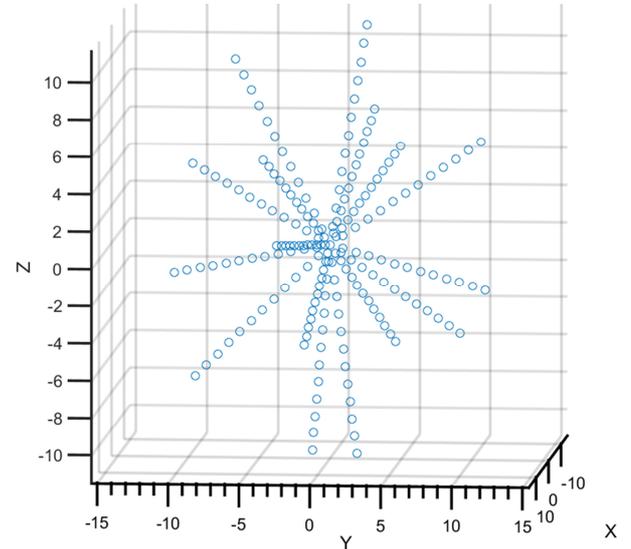


Figure 3. The grid used in the ZnO tests,  $M = 16$ ,  $P = 14$

### B. The neural network hyperparameters

A two hidden layer feed-forward neural network with ELU (Exponential Linear Unit) neuron activations was used. The number of NN inputs equal to  $M \cdot P \cdot N$ , where  $N$  is the number of atom types (elements) used. The number of outputs is equal to 3, one for each component of the force. The number of neurons per hidden layer should be proportional to the resolution of the grid, to be able to fully use the information encoded in the grid to approximate the force.

### III. DETERMINATION OF THE WEIGHT PARAMETERS AND GPU IMPLEMENTATION

The weigh parameters of the NN can be determined using the stochastic gradient descent algorithm (SGD) with ADAM [8]. The training data set should be as accurate and diverse as possible, for best results. We used randomly generated clusters of atoms, forces for which were calculated using DFT method, but for better results it is recommended to use real structures as well as random ones.

Because the DFT calculations can take a lot of time, it is possible to artificially extend the training set. The environment description functions are non-invariant for spatial rotations; therefore, we can randomly rotate the atomic systems with the DFT calculated forces to extend the training set.

When implementing the algorithm on GPU it is important to note, that access to the global memory is extremely slow, and is the main bottleneck for most GPU based algorithms, therefore requiring tricks to minimize it. To calculate the environment description functions, we need to find a sum over all neighboring atoms inside the cut-off sphere, on CPU this can be easily achieved by looping over all atoms while checking if the atom is inside the sphere, but on the GPU, for a large number of atoms, this will require a large amount of global memory calls, to be more precise, it will be  $O(n^2)$ . This can be speed up, by using a look up grid. Therefore, this sum can be calculated by checking the atoms in the neighboring grid points, making the memory calls proportional only to the cut-off sphere size, and the look-up grid resolution. Finding such a lookup table can be parallelized quite efficiently, first we need to calculate an array of the respective grid indices for each atom ( $O(n)$ ), then sort this array by the grid indices using

bitonic sort ( $O(n \log^2(n))$ ), now that the array is sorted we can save the atom id's to each grid point while accessing only one single unique grid point from one GPU core ( $O(n)$ ). Thus, this approach is only  $O(n \log^2(n))$  for  $n$  atoms.

This algorithm only allows a limited rectangular space to be used, and for a larger simulation space we need a lot more GPU memory for the look-up table.

### IV. RESULTS

To test the accuracy of finding equilibrium states we used this model on ZnO clusters (Fig. 4). Random ZnO clusters and randomized wurtzite crystals were used to calculate the DFT training data set, as well as the test data set.

The neural network had 31 neurons per hidden layer. It was trained on a Nvidia Tesla C2050 (Train PC) for 35000 SGD iterations, it took about 5 minutes to finish. The SGD ADAM optimization parameters were the following: mini-batch size of 256,  $\alpha = 0.001$ ,  $\beta_1 = 0.99$ ,  $\beta_2 = 0.9995$ ,  $\epsilon = 0.9995$ .

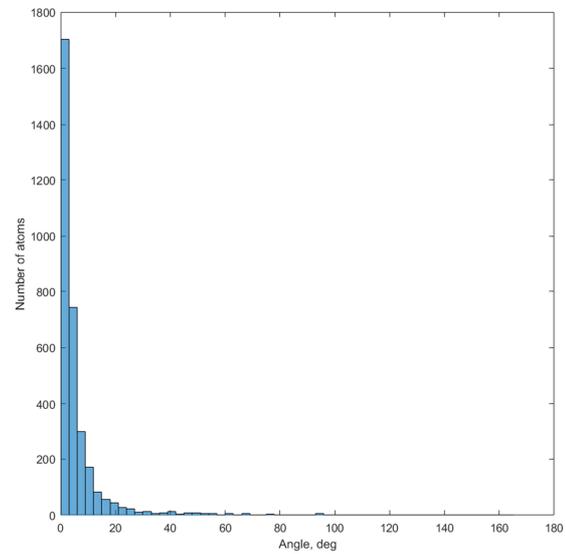


Figure 4. Angle divergence between NN and DFT calculated force vectors histogram for the test data set

DFT calculations are performed using the code in which each electronic wave function is expanded in a localized atom-

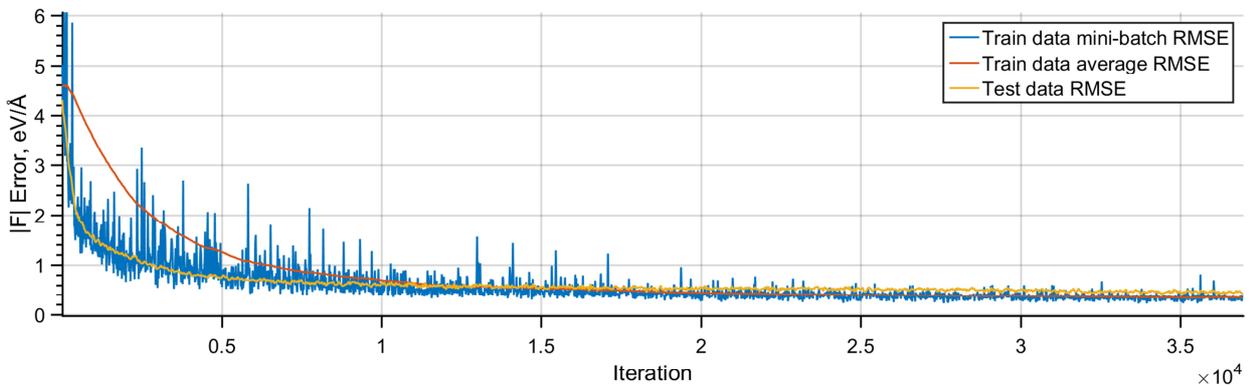


Figure 5. Training process RMSE for train and test data sets

centered basis set defined on a numerical grid. We performed all-electron calculations with a double numeric polarized (DNP) basis set and the gradient-corrected PBE functional [9]. In total, 8168 16-atom clusters were generated, 8000 of which were used for training, and 168 for testing. In total, this accounts for 128000 neural network train samples, and 2688 test samples. The training process root-mean-square error (RMSE) for train and test data sets is shown on Fig. 5.

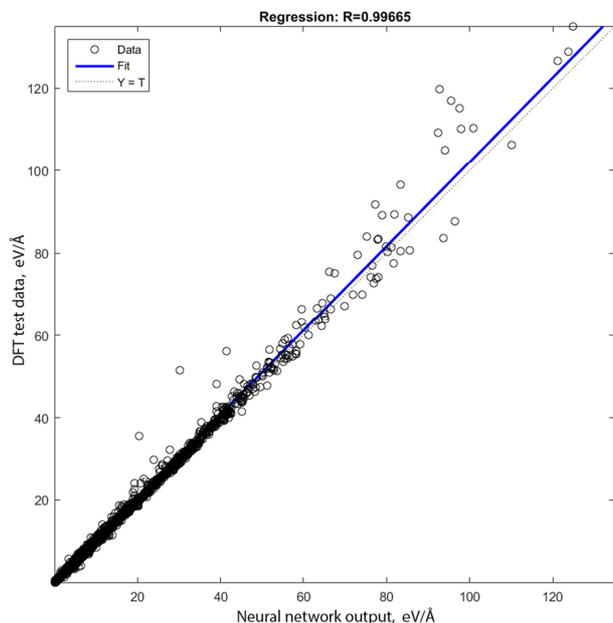


Figure 6. Regression for the approximated force magnitude values for the test data

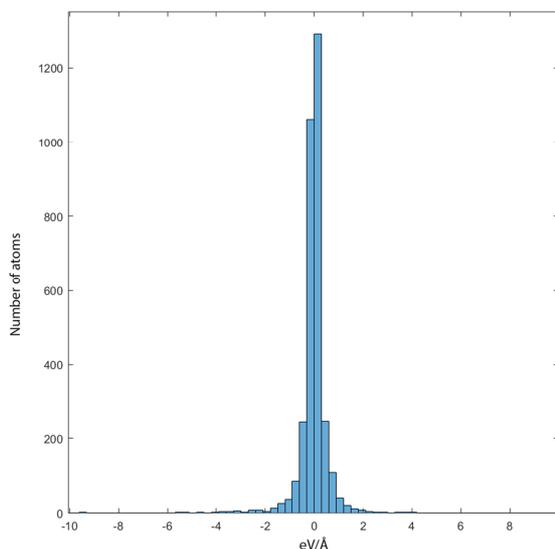


Figure 7. Approximated force magnitude error distribution histogram for the test data

The magnitudes of the DFT calculated forces were ranged from 0.0378 eV/Å to 124.8452 eV/Å, with an average force of 10.7998 eV/Å (Fig. 6).

The final accuracy of the NN force approximation on the test data was about 0.2687 eV/Å, compared to 0.0304 eV/Å [10] and 0.1674 eV/Å [11] by using PES based NN models, the average angle divergence from the DFT force vectors was 6.5294 degrees. Therefore, the neural network on the test data has a relative error of 2.488% (Fig. 7).

Using this NN model several test clusters were optimized, a 480 atom wurtzite ZnO nanocrystal was optimized in less than 5 seconds on the train PC.

## V. CONCLUSIONS

In summary, we have proposed a very fast way to calculate interatomic forces based on neural networks taking into account the positions of all atoms in systems of arbitrary size, and implemented on GPU using the OpenCL API, significantly extending the applicability of NNs to study large structures. The capability of the method was tested on ZnO clusters. Compared to PES based approaches this one is worse accuracy-wise, and faster performance-wise, because it was redesigned to work on GPU's.

## REFERENCES

- [1] J. Behler and M. Parrinello, "Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces," *Phys. Rev. Lett.*, vol. 98, no. 14, p. 146401, Apr. 2007.
- [2] V. V. Gafiychuk, B. K. Ostafiychuk, D. I. Popovych, I. D. Popovych, and A. S. Serednytski, "ZnO nanoparticles produced by reactive laser ablation," *Appl. Surf. Sci.*, vol. 257, no. 20, pp. 8396–8401, Aug. 2011.
- [3] V. M. Zhyrovetsky, D. I. Popovych, S. S. Savka, and A. S. Serednytski, "Nanopowder Metal Oxide for Photoluminescent Gas Sensing," *Nanoscale Res. Lett.*, vol. 12, no. 1, p. 132, Dec. 2017.
- [4] O. V. Bovgyra, R. V. Bovgyra, D. I. Popovych, and A. S. Serednytski, "The density functional theory study of electronic properties of (ZnO)<sub>12</sub> clusters during gas adsorption," *J. Nano- Electron. Phys.*, vol. 7, no. 4, 2015.
- [5] O. V. Bovgyra, R. V. Bovgyra, M. V. Kovalenko, D. I. Popovych, and A. S. Serednytski, "The density functional theory study of structural and electronic properties of zno clusters," *J. Nano- Electron. Phys.*, vol. 5, no. 1, 2013.
- [6] R. Bovhyra, D. Popovych, O. Bovgyra, and A. Serednytski, "Ab Initio Study of Structural and Electronic Properties of (ZnO)<sub>n</sub> 'Magical' Nanoclusters n = (34, 60)," *Nanoscale Res. Lett.*, vol. 12, no. 1, 2017.
- [7] J. Behler, "Constructing high-dimensional neural network potentials: A tutorial review," *Int. J. Quantum Chem.*, vol. 115, no. 16, pp. 1032–1050, Aug. 2015.
- [8] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Dec. 2014, arXiv:1412.6980.
- [9] J.P. Perdew, K. Burke, and M. Ernzerhof, "Generalized gradient approximation made simple," *Phys. Rev. Lett.*, vol. 77, pp. 3865–3868, 1996
- [10] N. Artrith, T. Morawietz, and J. Behler, "High-dimensional neural-network potentials for multicomponent systems: Applications to zinc oxide," *Phys. Rev. B*, vol. 83, no. 15, p. 153101, Apr. 2011.
- [11] N. Artrith, B. Hiller, and J. Behler, "Neural network potentials for metals and oxides - First applications to copper clusters at zinc oxide," *Phys. status solidi*, vol. 250, no. 6, pp. 1191–1203, Jun. 2013.