

Development of Neural Network Models for Prediction of Molecular Properties

John E. Herr

Publication Date

21-04-2020

License

This work is made available under a Public Domain Mark 1.0 (No Copyright) license and should only be used in accordance with that license.

Citation for this work (American Psychological Association 7th edition)

Herr, J. E. (2020). *Development of Neural Network Models for Prediction of Molecular Properties* (Version 1). University of Notre Dame. <https://doi.org/10.7274/j3860577b2v>

This work was downloaded from CurateND, the University of Notre Dame's institutional repository.

For more information about this work, to report or an issue, or to preserve and share your original work, please contact the CurateND team for assistance at curate@nd.edu.

Supplementary information for: Development of neural network models for prediction of molecular properties

John E. Herr

April 21, 2020

Contents

1	Supplementary information for: The many-body expansion combined with neural networks	3
1.1	Radius cutoff of two-body energy and three-body energy	3
1.2	Learning permutation invariance	3
1.3	Molecular dynamic trajectory	5
1.4	Timing	6
1.5	Comparison with AMOEBA09	6
1.6	Code	7
2	Supplementary information for: Metadynamics for generating off-equilibrium geometries	10
2.1	Neural Network MAE and RMSE	10
3	Supplementary information for: TensorMol model with long-range physics	13
3.1	Hyperparameter search	13
3.2	IR spectra from MMFF94 compared to DFT	13
3.3	IR spectra from experimental results	14
3.4	Relative energies of six conformers of the water hexamer	15
3.5	Comparison of TensorMol with and without long-range physics included	16
3.6	Timing	16
3.7	Data	17
3.8	Procedures for using TensorMol 0.1	18
4	Supplementary information for: Fully transferable high-dimensional neural network potentials	19
4.1	Code	19
4.2	Physical data used to train the autoencoder	19

5	Supplementary information for: Stokes shifts in lead halide perovskites	23
5.1	Projected density of states	23
5.2	Spin-Orbit Coupling	23
5.3	Exciton Binding Energies	24

1 Supplementary information for: The many-body expansion combined with neural networks

1.1 Radius cutoff of two-body energy and three-body energy

In order to choose a reasonable radius cutoff, the convergence of two-body and three-body energies with respect to the radius cutoff is studied. Figure 1 shows the two-body and three-body energies with respect to the cutoff of a cluster of 108 methanol molecules which is randomly sampled from an MD trajectory at 330 K. As it is shown in the Figure 1, the convergence of the two-body energy is not achieved until the cutoff reaches 8 Å. The three-body energy requires a larger cutoff to converge than the two-body energy and a cutoff of at least 10 Å is required.

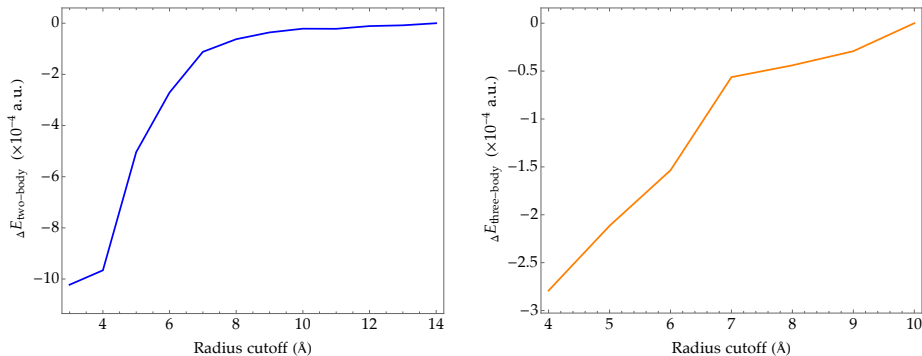


Figure 1: Change in energy per molecule of the two-body (left panel) and three-body (right panel) contributions to the total energy as the cutoff distance is increased in a random cluster of 108 methanol molecules.

1.2 Learning permutation invariance

The permutation invariance is learned by augmenting all the possible permutations for the samples into the training data set. For one-body cases, there are six possible permutations since the three hydrogen atoms on a methyl group are equivalent. The methanol dimer has two methyl groups and two interchangeable molecules, therefore it has 72 possible permutations. For the same reason, the methanol trimer has 1296 possible permutations, and it is difficult to include all of them. Therefore, we combine the chemically inert hydrogen atoms on a methyl group into one imaginary atom by taking the average of the coordination of the three hydrogen atoms, which eliminates the permutations of hydrogen atoms on a methyl group and left 6 possible permutations caused by the interchange of methanol molecules in a methanol trimer.

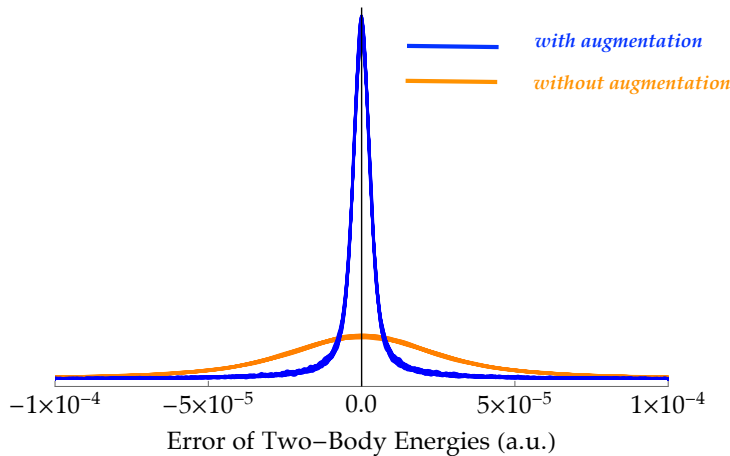


Figure 2: Histogram of the two-body energy error of the neural network trained with data augmentation (blue) and the neural network trained without data augmentation (orange). The errors of the neural network trained without augmentation is much more spread over the axis, while the one trained with data augmentation has a much smaller spread of errors.

The histogram of the two-body energy error of the the neural network that is trained with augmentation and without augmentation is shown in Figure 2. The test data set includes all the 72 possible permutations of the coulomb matrix of the original test data set. Error of the two-body energy is examined here since it has the most possible permutations of all of the many-body energy terms. One can see that the error of the neural network trained without augmentations is spread over the axis, while the error of the neural network trained with augmentation mostly populates in the region between -10^{-5} a.u. and 10^{-5} a.u.. The change of the total two-body energy, predicted by the neural network when one methanol is pulled away from the other two in a methanol trimer is shown in the left and center panels of Figure 3. Each line in the two panels shows the two-body energy that is predicted by neural network based on one possible permutation of the Coulomb matrix. The right panel of Figure 3 shows the standard deviation of the energies predicted by the neural network based on all the 72 possible coulomb matrix. One can see that, compared with the model trained without data augmentation, the neural network trained with data augmentation predicts a much more consistent energy for the 72 different permuted Coulomb matrices, which shows the neural network learns the permutation invariance from the augmented training data.

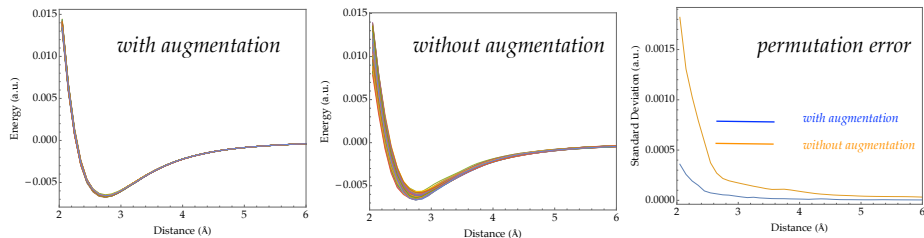


Figure 3: Left and middle panels: Change of the total two-body energy as one methanol is pulled away from the other two in a methanol trimer, calculated by the neural network trained with data augmentation (left) and without data augmentation (middle). Each of the possible 72 permutations is plotted on both panels. Right panel: Standard deviation of the 72 energies predicted by the neural network. One can see the neural network trained with data augmentation predicts a much more consistent energy for the different permuted Coulomb matrices.

1.3 Molecular dynamic trajectory

Figure 4 shows the potential energy calculated by MP2-MBE and the NN-MBE. One can see that the curves overlap with each other well with only small discrepancies at any point. The energy difference between MP2-MBE and NN-MBE is smaller than 10^{-3} a.u. throughout the trajectory.

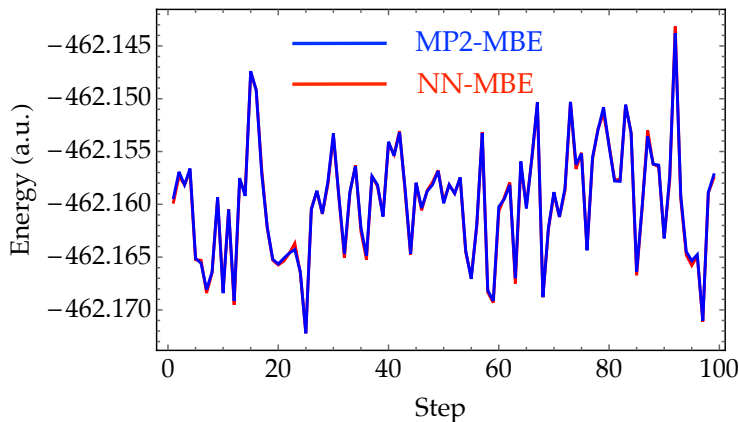


Figure 4: PES calculated by MP2-MBE (blue) and NN-MBE (red) of 4 methanol molecules along 100 steps of a molecular dynamic trajectory at 330 K. The NN-MBE predicts energies consistent with the MP2-MBE and the energy difference between MP2-MBE and NN-MBE is smaller than 10^{-3} a.u..

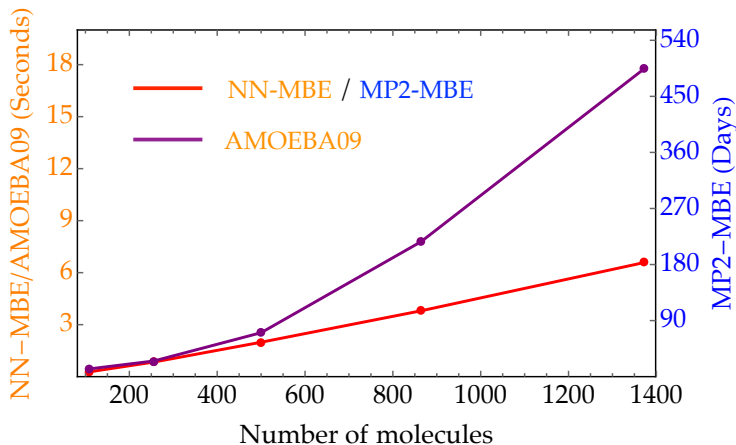


Figure 5: Total wall time for the NN-MBE (red curve, left orange y-axis), MP2-MBE (red, curve, right blue y-axis) AMOEBA09 (purple curve, left orange y-axis) to calculate the energies of methanol clusters of different size. One can see that NN-MBE has a more than two million times speed up over MP2-MBE. The NN-MBE calculations were done using one Tesla K80 GPU, MP2-MBE calculations were done with one 24-core CPU and AMOEBA09 calculation was performed on one CPU using *TINKER* package.

1.4 Timing

Figure 5 shows the total wall time comparison of MP2-MBE, the NN-MBE, and AMOEBA09. A cutoff of 10 Å is used for MP2-MBE and NN-MBE, so both methods will be near linear scaling for large clusters. The NN-MBE has a speed up of more six orders of magnitude relative to MP2-MBE, which enables us to use the NN-MBE to calculate the energy of large clusters of thousands of molecules in seconds with *ab initio* accuracy, which would take months for an MP2-MBE calculation. The AMOEBA09 calculation is done with the *TINKER* package. The total wall time of the NN-MBE is similar to AMOEBA09.

1.5 Comparison with AMOEBA09

Figure 6 compares the potential energy for a methanol trimer binding curve calculated with MP2, the NN-MBE, and AMOEBA09. Both the NN-MBE and AMOEBA09 get the shape of the binding energy curve and the minimum energy distance correct, but NN-MBE is shown to be more accurate compared to AMOEBA09.

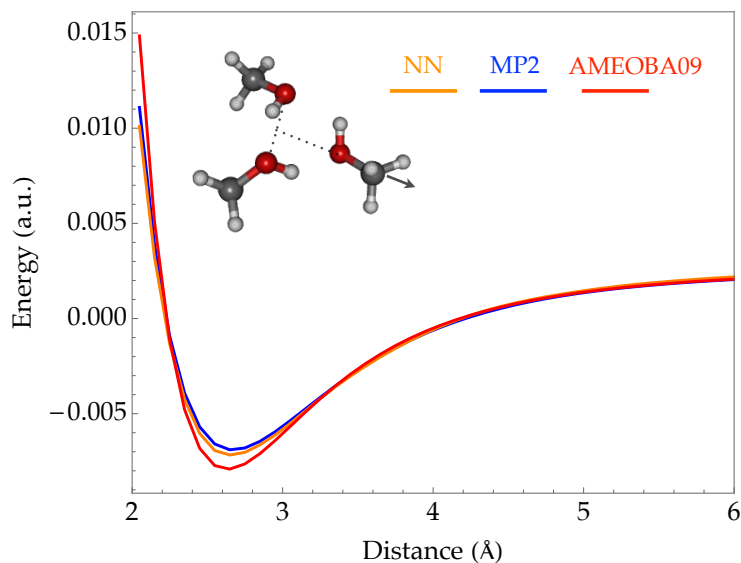


Figure 6: Change of the total energy when one methanol is pulled away from the other two in a methanol trimer, calculated by NN-MBE, MP2 and AMOEBA09. One can see that NN-MBE has better agreement with MP2 than AMOEBA09.

1.6 Code

Code to reproduce the NN-MBE code is available in TensorMol at https://github.com/jparkhill/TensorMol_MBE.git. The code to produce a depth map from given a set of coordinates and atomic numbers is given below.

Depth map code:

```
double abin = 0.143;
double reabin = 1/abin;
int xa, ya, za;
double x, y, z, dist;
double testpos[3];

double vdWC = 1.85/3;
double vdWH = 1.2/3;
double vdWO = 1.40/3;

int posmax = (vdWC*reabin);

for(int m=0; m<18;m++) {
    xa = xyz[m*3]*reabin;
    ya = xyz[m*3+1]*reabin;
    za = xyz[m*3+2]*reabin;
```



```

for (int j=-(posmax+4);j<(posmax+5);j++)
  for (int k=-(posmax+4);k<(posmax+5);k++)
    for (int l=-(posmax+4);l<(posmax+5);l++) {

      testpos[0] = (xa+j)*abin;
      testpos[1] = (ya+k)*abin;
      testpos[2] = (za+l)*abin;

      x = testpos[0]-xa*abin;
      y = testpos[1]-ya*abin;
      z = testpos[2]-za*abin;

      dist = (x*x)+(y*y)+(z*z);
      dist = sqrt(dist);
      if (deppl_[(xa+j)*width+(ya+k)] == 0) {
        if ( idxlist_ [m] == 1) {
          if (dist <= vdWC) {
            deppl_[(xa+j)*width+(ya+k)] =
              (za+l)*abin;
            break;
          }
        }
        if ( idxlist_ [m] == 2) {
          if (dist <= vdWO) {
            deppl_[(xa+j)*width+(ya+k)] =
              (za+l)*abin;
            break;
          }
        }
        if ( idxlist_ [m] == 0) {
          if (dist <= vdWH) {
            deppl_[(xa+j)*width+(ya+k)] =
              (za+l)*abin;
            break;
          }
        }
      }
    }
  if (deppl_[(xa+j)*width+(ya+k)] != 0) {

    if ( idxlist_ [m] == 1) {
      if (dist <= vdWC) {
        if ((za+l)*abin <
            deppl_[(xa+j)*width+(ya+k)]) {
          deppl_[(xa+j)*width+(ya+k)] =
            (za+l)*abin;
        }
      }
    }
  }
}

```

```

        break;
    }
}
}
if ( idxlist_ [m] == 2) {
    if ( dist <= vdWO) {
        if ((za+1)*abin <
            deppl_[(xa+j)*width+(ya+k)]) {
            deppl_[(xa+j)*width+(ya+k)] =
                (za+1)*abin;
            break;
        }
    }
}
if ( idxlist_ [m] == 0) {
    if ( dist <= vdWH) {
        if ((za+1)*abin <
            deppl_[(xa+j)*width+(ya+k)]) {
            deppl_[(xa+j)*width+(ya+k)] =
                (za+1)*abin;
            break;
        }
    }
}
}
}
}
}
}

```

2 Supplementary information for: Metadynamics for generating off-equilibrium geometries

2.1 Neural Network MAE and RMSE

The calculated MAE and RMSE for each of the total 18 networks trained are presented in Tables 1-6. Each table compares models trained with the same amount of training data. MAE and RMSE values are reported for each model across the independent test sets from each sampling method.

Table 1: Energy errors (kcal/mol) for networks trained on 2000 geometries cross-evaluated on all data generation methods.

Training Data	Evaluation Data					
	AIMD		MetaMD		Vibration	
	<i>MAE</i>	<i>RMSE</i>	<i>MAE</i>	<i>RMSE</i>	<i>MAE</i>	<i>RMSE</i>
AIMD	0.301	0.424	67.008	69.668	48.881	66.141
MetaMD	2.935	3.641	2.525	3.434	7.290	9.974
Vibration	15.739	18.029	10.185	13.080	2.847	3.916

Table 2: Energy errors (kcal/mol) for networks trained on 4000 geometries cross-evaluated on all data generation methods.

Training Data	Evaluation Data					
	AIMD		MetaMD		Vibration	
	<i>MAE</i>	<i>RMSE</i>	<i>MAE</i>	<i>RMSE</i>	<i>MAE</i>	<i>RMSE</i>
AIMD	0.136	0.186	79.116	82.070	58.898	79.077
MetaMD	4.091	4.693	1.236	1.747	7.584	9.053
Vibration	23.098	24.386	14.202	17.241	3.966	4.720

Table 3: Energy errors (kcal/mol) for networks trained on 8000 geometries cross-evaluated on all data generation methods.

Training Data	Evaluation Data					
	AIMD		MetaMD		Vibration	
	<i>MAE</i>	<i>RMSE</i>	<i>MAE</i>	<i>RMSE</i>	<i>MAE</i>	<i>RMSE</i>
AIMD	0.052	0.068	82.547	85.633	63.166	84.819
MetaMD	5.329	5.899	0.471	0.635	7.361	8.520
Vibration	9.063	10.143	9.117	11.300	1.306	1.940

Table 4: Energy errors (kcal/mol) for networks trained on 16000 geometries cross-evaluated on all data generation methods.

Training Data	Evaluation Data					
	AIMD		MetaMD		Vibration	
	<i>MAE</i>	<i>RMSE</i>	<i>MAE</i>	<i>RMSE</i>	<i>MAE</i>	<i>RMSE</i>
AIMD	0.032	0.051	87.578	90.682	62.904	84.041
MetaMD	6.563	7.048	0.310	0.393	8.490	9.655
Vibration	7.160	8.485	7.973	10.184	1.090	1.559

Table 5: Energy errors (kcal/mol) for networks trained on 32000 geometries cross-evaluated on all data generation methods.

Training Data	Evaluation Data					
	AIMD		MetaMD		Vibration	
	<i>MAE</i>	<i>RMSE</i>	<i>MAE</i>	<i>RMSE</i>	<i>MAE</i>	<i>RMSE</i>
AIMD	0.018	0.024	90.387	93.469	65.654	87.704
MetaMD	6.833	7.399	0.180	0.226	9.474	10.595
Vibration	7.161	8.485	6.439	8.336	0.995	1.392

Table 6: Energy errors (kcal/mol) for networks trained on 40000 geometries cross-evaluated on all data generation methods.

Training Data	Evaluation Data					
	AIMD		MetaMD		Vibration	
	<i>MAE</i>	<i>RMSE</i>	<i>MAE</i>	<i>RMSE</i>	<i>MAE</i>	<i>RMSE</i>
AIMD	0.015	0.019	90.705	93.778	64.762	86.284
MetaMD	7.267	7.865	0.175	0.220	9.950	11.103
Vibration	9.235	10.589	6.733	8.894	1.040	1.317

3 Supplementary information for: TensorMol model with long-range physics

3.1 Hyperparameter search

A hyperparameter search was conducted to determine suitable parameters of TensorMol. Hyperparameters searched over include activation function, number of hidden layers, and the number of neurons per hidden layer. Results are presented in Table 7.

Table 7: Test RMSE of each learning target for water networks trained with different hyperparameters. The unit of energy, gradient, and dipole RMSEs is kcal/mol/atom, kcal/mol/Å per atom and Debye per atom, respectively.

Hidden layers	Neurons per layer	Activation function	Energy	Gradient	Dipole
3	500	Softplus ($\alpha = 100$)	0.054	0.49	0.0082
3	100	Softplus ($\alpha = 100$)	0.058	0.56	0.0090
3	200	Softplus ($\alpha = 100$)	0.059	0.52	0.0086
3	1000	Softplus ($\alpha = 100$)	0.066	0.48	0.0082
1	500	Softplus ($\alpha = 100$)	0.065	0.69	0.0086
2	500	Softplus ($\alpha = 100$)	0.093	0.54	0.0085
4	500	Softplus ($\alpha = 100$)	0.054	0.50	0.0083
3	500	Softplus ($\alpha = 10$)	0.089	0.80	0.0090
3	500	Softplus ($\alpha = 1$)	0.61	3.4	0.011
3	500	Tanh	0.24	1.1	0.010
3	500	Sigmoid	0.38	2.6	0.011
3	500	Guassian	0.075	0.66	0.0098

3.2 IR spectra from MMFF94 compared to DFT

IR spectra were also calculated for morphine, aspirin, tyrosine, caffeine, and cholesterol using MMFF94.[1] IR spectra for the five molecules are shown in Figures 7 and 8. Calculated spectra from DFT are shown for comparison. The spectra calculated with MMFF94 are significantly less accurate than TensorMol as compared to DFT.

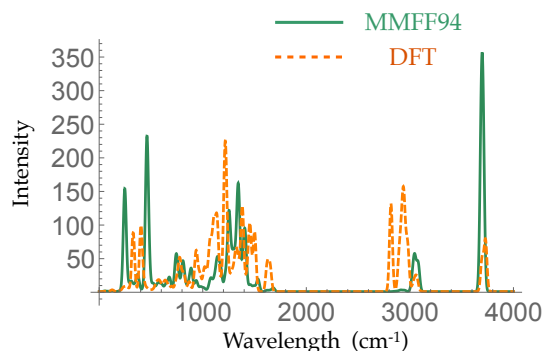


Figure 7: Harmonic IR spectrum of morphine simulated by ω B97X-D/6-311G** (dashed orange line) and MMFF94 functions (solid green line). MMFF94 frequencies are calculated using RDKit [1]. DFT frequencies are scaled by a factor of 0.957 [2]. The MAE of MMFF94 frequencies is 28.4 cm^{-1} .

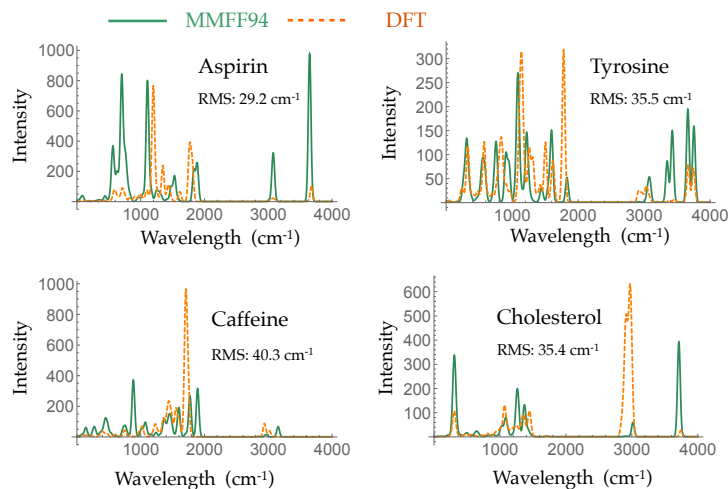


Figure 8: Harmonic IR spectrum of four different molecules simulated by ω B97X-D/6-311G** (dashed orange line) and MMFF94 (solid green line). MMFF94 frequencies are calculated using RDKit [1]. DFT frequencies are scaled by a factor of 0.957 [2].

3.3 IR spectra from experimental results

For comparison, the experimentally observed IR spectra is given for morphine from the NIST Chemistry WebBook.[3] The calculated frequencies and intensities from DFT and TensorMol appear to match the experimental result well.

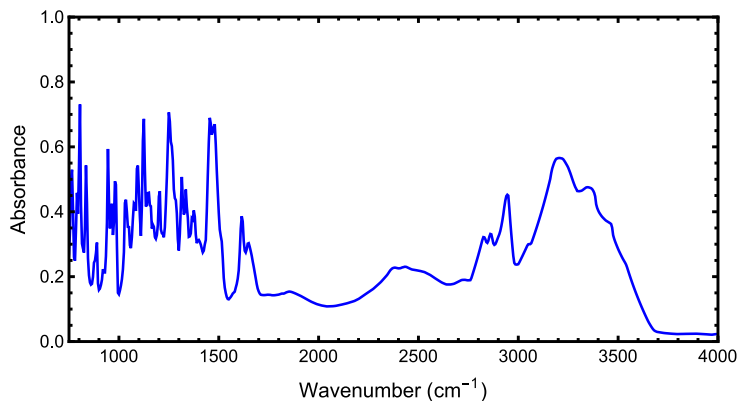


Figure 9: Experimental IR spectrum for morphine, given in absorbance. Data taken from NIST.[3]

3.4 Relative energies of six conformers of the water hexamer

Water hexamers exhibit six conformers which are close in energy.[4] The energy of each conformer is calculated with ω B97X-D/6-311G** and TensorMol and results are shown in Figure 10. The relative ordering of the lowest energy conformers are correctly predicted by TensorMol.

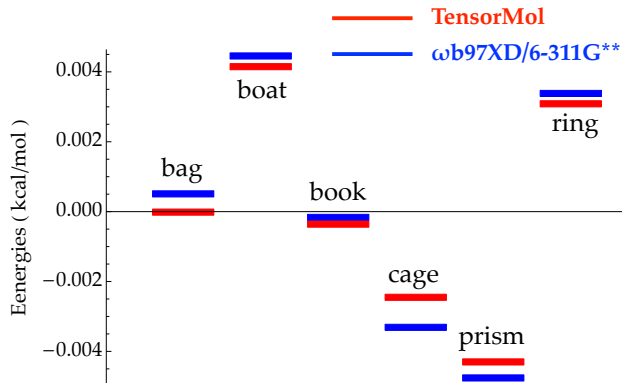


Figure 10: Relative energies of different conformers of water hexamer cluster[4]. TensorMol predicts the same order of energies as the target method, ω B97X-D/6-311G** and the MAE of the relative energies predicted by TensorMol is 0.44 kcal/mol.

3.5 Comparison of TensorMol with and without long-range physics included

The inclusion of Coulomb and dispersion energies are necessary to achieve the correct long-range behavior. A second version of TensorMol was trained without including Coulomb or dispersion interactions directly. Each version was used to optimize a system of a hydronium, water, and hydroxyl aligned vertically, shown in Figure 11. Optimizing with the full version exhibits the correct behavior by transferring the proton from the hydronium to the water, and then another proton transfer from water to hydroxyl. The version without long-range physics included does not reproduce this behavior.

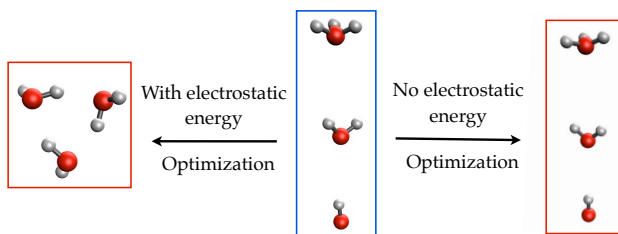


Figure 11: Geometry optimization of a water trimer that contains OH^- and H_3O^+ . The neural network that includes electrostatic interaction successfully bring the cluster to the correct global minimal while the one that does not consider electrostatic interaction is stuck at local minimal because of the lack of long range interaction.

3.6 Timing

A comparison of the wall time required to calculate energy, forces, and dipole moments with TensorMol and with Q-Chem using $\omega B97X-D/6-311G^{**}$ for increasing sizes of water clusters is shown in Figure 12. TensorMol is about five orders of magnitude faster than Q-Chem.

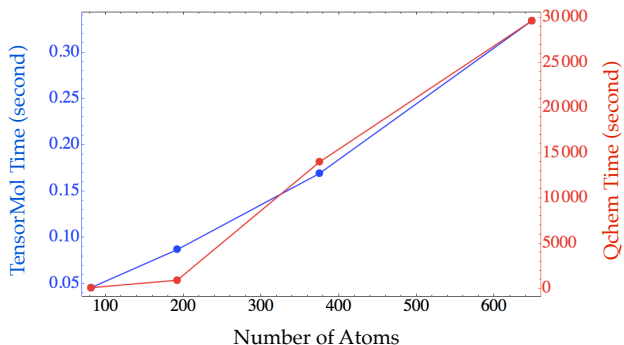


Figure 12: Timing of TensorMol force field and Qchem for different size of water clusters. The GPU timing is measured on single Nvidia K40 GPU and CPU timing is measured on two 16 thread Intel Xeon CPU E5-2667 v4.

TensorMol can be run on GPU or CPU depending on hardware available. Wall times to calculate energies, forces, and dipoles for TensorMol with CPU and GPU are shown in Figure 13. The advantages of running TensorMol on GPU becomes greater for larger systems. The neighborlist implemented in TensorMol is only implemented on CPU, so both timings include CPU time for the neighborlist build.

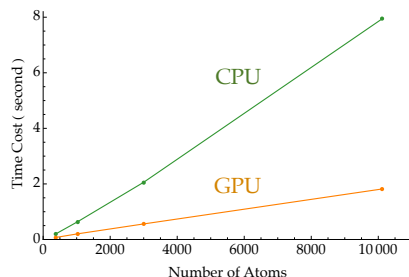


Figure 13: Aperiodic timings of an energy, charge, force call for cubic water clusters at a density of 1 gm/cm without considering neighbor list building which is not implemented in GPU yet. The GPU timing is measured on single Nvidia K40 GPU and CPU timing is measured on a 8 thread Intel Xeon CPU E5-1620 v2. The GPU is 3 times to 4 times faster than CPU per call.

3.7 Data

Both the water and ChempSpider datasets are available at https://drive.google.com/drive/folders/1IfWPs7i5kfmErIRyuhGv95dSVtNF0e_?usp=sharing. Datasets are in a format which can be loaded in the TensorMol package.

3.8 Procedures for using TensorMol 0.1

- Install Tensorflow, Python.
- Download TensorMol from <https://github.com/jparkhill/TensorMol>. Checkout master branch. Download the trained water network and chemspider network from https://drive.google.com/drive/folders/1IfWPs7i5kfmErIRyuhGv95dSVtNFo0e_?usp=sharing. Copy the trained networks (network.tar.bz2) into TensorMol folder. Unzip trained networks.
- Copy the test script test_tensormol01.py in folder ”./samples” to into the Tensormol folder. Run the script test the geometry optimization, molecular dynamic, harmonic IR spectrum and realtime IR spectrum.
- Demo of training a neural network force field using TensorMol: Copy the training script training_sample.py into the tensormol folder. Run the script. This will train a network force field for water.

4 Supplementary information for: Fully transferable high-dimensional neural network potentials

4.1 Code

Three separate repositories contain the code essential to reproducing this work. The autoencoder is available at <https://github.com/jeherr/element-encoder>. The model used to train on the elpasolite formation energies is available at <https://github.com/jeherr/Elpasolite-Formation-Energy-Predictor>. The new TensorMol model is available in the most recent updates of TensorMol available at <https://github.com/jeherr/tensormol>. A standalone version of the model code is also available at <https://github.com/jeherr/emode-hdnp>.

4.2 Physical data used to train the autoencoder

Physical property data used to train the autoencoder is reported in Table 8. Values are normalized across a column before being fed into the autoencoder for training to help the autoencoder to train easier.

Table 8: Physical property data used to train the autoencoder network.

Symbol	Atomic number	Standard atomic mass (amu)	# s elec.	# p elec.	# d elec.	Electro- negativity	Radius (pm)	Ionization energy (kJ/mol)	Electron affinity (kJ/mol)	Polarizability (a.u.)
H	1.0	1.0079	1.0	0.0	0.0	2.3	53.0	1312.0	0.754195	4.4923
He	2.0	4.0026	2.0	0.0	0.0	4.16	31.0	2372.3	-0.52	1.3831
Li	3.0	6.941	1.0	0.0	0.0	0.912	167.0	520.2	0.618049	164.0
Be	4.0	9.0121	2.0	0.0	0.0	1.576	112.0	899.5	-0.52	37.71
B	5.0	10.811	2.0	1.0	0.0	2.051	87.0	800.6	0.279723	20.53
C	6.0	12.0107	2.0	2.0	0.0	2.544	67.0	1086.5	1.2621226	11.26
N	7.0	14.0067	2.0	3.0	0.0	3.066	56.0	1402.3	-0.000725	7.26
O	8.0	15.9994	2.0	4.0	0.0	3.61	48.0	1313.9	1.4611136	5.24
F	9.0	18.9984	2.0	5.0	0.0	4.193	42.0	1681.0	3.4011898	3.7
Ne	10.0	20.1797	2.0	6.0	0.0	4.787	38.0	2080.7	-1.2	2.67
Na	11.0	22.9897	1.0	0.0	0.0	0.869	190.0	495.8	0.547926	162.7
Mg	12.0	24.305	2.0	0.0	0.0	1.293	145.0	737.7	-0.415	70.89
Al	13.0	26.9815	2.0	1.0	0.0	1.613	118.0	577.5	0.43283	55.4
Si	14.0	28.0855	2.0	2.0	0.0	1.916	111.0	786.5	1.3895212	37.31
P	15.0	30.9737	2.0	3.0	0.0	2.253	98.0	1011.8	0.746607	24.93
S	16.0	32.065	2.0	4.0	0.0	2.589	88.0	999.6	2.0771042	19.37
Cl	17.0	35.453	2.0	5.0	0.0	2.869	79.0	1251.2	3.612724	14.57
Ar	18.0	39.948	2.0	6.0	0.0	3.242	71.0	1520.6	-1.0	11.07
K	19.0	39.0983	1.0	0.0	0.0	0.734	243.0	418.8	0.501459	290.6
Ca	20.0	40.078	2.0	0.0	0.0	1.034	194.0	589.8	0.02455	155.9
Sc	21.0	44.9559	2.0	0.0	1.0	1.19	184.0	633.1	0.188	142.28
Ti	22.0	47.867	2.0	0.0	2.0	1.38	176.0	658.8	0.084	114.34
V	23.0	50.9415	2.0	0.0	3.0	1.53	171.0	650.9	0.52766	97.34

Table 8: *Continued*

Symbol	Atomic number	Standard atomic mass (amu)	# s elec.	# p elec.	# d elec.	Electro- negativity	Radius (pm)	Ionization energy (kJ/mol)	Electron affinity (kJ/mol)	Polarizability (a.u.)
Cr	24.0	51.9961	1.0	0.0	5.0	1.65	166.0	652.9	0.67584	78.4
Mn	25.0	54.9380	2.0	0.0	5.0	1.75	161.0	717.3	-0.52	66.8
Fe	26.0	55.845	2.0	0.0	6.0	1.8	156.0	762.5	0.153236	62.65
Co	27.0	58.9331	2.0	0.0	7.0	1.84	152.0	760.4	0.66226	57.71
Ni	28.0	58.6934	2.0	0.0	8.0	1.88	149.0	737.1	1.15716	51.1
Cu	29.0	63.546	1.0	0.0	10.0	1.85	145.0	745.5	1.23578	40.7
Zn	30.0	65.38	2.0	0.0	10.0	1.59	142.0	906.4	-0.62	38.8
Ga	31.0	69.723	2.0	1.0	10.0	1.756	136.0	578.8	0.43	51.4
Ge	32.0	72.64	2.0	2.0	10.0	1.994	125.0	762.0	1.2326764	39.43
As	33.0	74.9216	2.0	3.0	10.0	2.211	114.0	947.0	0.8048	29.8
Se	34.0	78.96	2.0	4.0	10.0	2.424	103.0	941.0	2.0206047	26.24
Br	35.0	79.904	2.0	5.0	10.0	2.685	94.0	1139.9	3.363588	21.03
Kr	36.0	83.798	2.0	6.0	10.0	2.966	88.0	1350.8	-0.62	17.075
Rb	37.0	85.4678	1.0	0.0	0.0	0.706	265.0	403.0	0.485916	318.8
Sr	38.0	87.62	2.0	0.0	0.0	0.963	219.0	549.5	0.05206	186.0
Y	39.0	88.9058	2.0	0.0	1.0	1.12	212.0	600.0	0.307	153.0
Zr	40.0	91.224	2.0	0.0	2.0	1.32	206.0	640.1	0.4333	121.0
Nb	41.0	92.9063	1.0	0.0	4.0	1.41	198.0	652.1	0.91740	106.0
Mo	42.0	95.96	1.0	0.0	5.0	1.47	190.0	684.3	0.7473	72.5
Tc	43.0	98.0	2.0	0.0	5.0	1.51	183.0	702.0	0.55	80.4
Ru	44.0	101.07	1.0	0.0	7.0	1.54	178.0	710.2	1.04638	65.0
Rh	45.0	102.9055	1.0	0.0	8.0	1.56	173.0	719.7	1.14289	58.0
Pd	46.0	106.42	0.0	0.0	10.0	1.58	169.0	804.4	0.56214	32.0
Ag	47.0	107.8682	1.0	0.0	10.0	1.87	165.0	731.0	1.30447	52.5
Cd	48.0	112.411	2.0	0.0	10.0	1.52	161.0	867.8	-0.725	46.9

Table 8: *Continued*

Symbol	Atomic number	Standard atomic mass (amu)	# s elec.	# p elec.	# d elec.	Electro- negativity	Radius (pm)	Ionization energy (kJ/mol)	Electron affinity (kJ/mol)	Polarizability (a.u.)
In	49.0	114.818	2.0	1.0	10.0	1.656	156.0	558.3	0.3	68.7
Sn	50.0	118.71	2.0	2.0	10.0	1.824	145.0	708.6	1.112070	42.4
Sb	51.0	121.76	2.0	3.0	10.0	1.984	133.0	834.0	1.047401	42.55
Te	52.0	127.6	2.0	4.0	10.0	2.158	123.0	869.3	1.970875	37.0
I	53.0	126.9044	2.0	5.0	10.0	2.359	115.0	1008.4	3.0590465	34.6
Xe	54.0	131.293	2.0	6.0	10.0	2.582	108.0	1170.4	-0.83	27.815
Cs	55.0	132.9054	1.0	0.0	0.0	0.659	298.0	375.7	0.471630	401.0
Ba	56.0	137.327	2.0	0.0	0.0	0.881	253.0	502.9	0.14462	268.0
Lu	71.0	174.9668	2.0	0.0	1.0	1.09	217.0	523.5	0.346	148.0
Hf	72.0	178.49	2.0	0.0	2.0	1.16	208.0	658.5	0.017	109.0
Ta	73.0	180.9478	2.0	0.0	3.0	1.34	200.0	761.0	0.323	88.0
W	74.0	183.84	2.0	0.0	4.0	1.47	193.0	770.0	0.81626	75.0
Re	75.0	186.207	2.0	0.0	5.0	1.60	188.0	760.0	0.060396	65.0
Os	76.0	190.23	2.0	0.0	6.0	1.65	185.0	840.0	1.1	57.0
Ir	77.0	192.217	2.0	0.0	7.0	1.68	180.0	880.0	1.56436	51.0
Pt	78.0	195.084	1.0	0.0	9.0	1.72	177.0	870.0	2.12510	44.0
Au	79.0	196.9665	1.0	0.0	10.0	1.92	174.0	890.1	2.308610	36.1
Hg	80.0	200.592	2.0	0.0	10.0	1.76	171.0	1007.1	-0.52	34.15
Tl	81.0	204.382	2.0	1.0	10.0	1.789	156.0	589.4	0.377	52.3
Pb	82.0	207.2	2.0	2.0	10.0	1.854	154.0	715.6	0.356743	46.96
Bi	83.0	208.9804	2.0	3.0	10.0	2.01	143.0	703.0	0.942362	50.0

5 Supplementary information for: Stokes shifts in lead halide perovskites

5.1 Projected density of states

Projected density of states (PDOS) were calculated to confirm the contributions to molecular orbitals matched that reported in the literature.[5, 6] Figure 14 shows an example from a nanocrystal model with an edge length of $l = 2.64$ nm. The top of the valence band stems from the antibonding interaction of the Pb(6s)-Br(4p) orbitals while the bottom of the conduction band corresponds to the Pb(6p)-Br(4p) antibonding interaction.

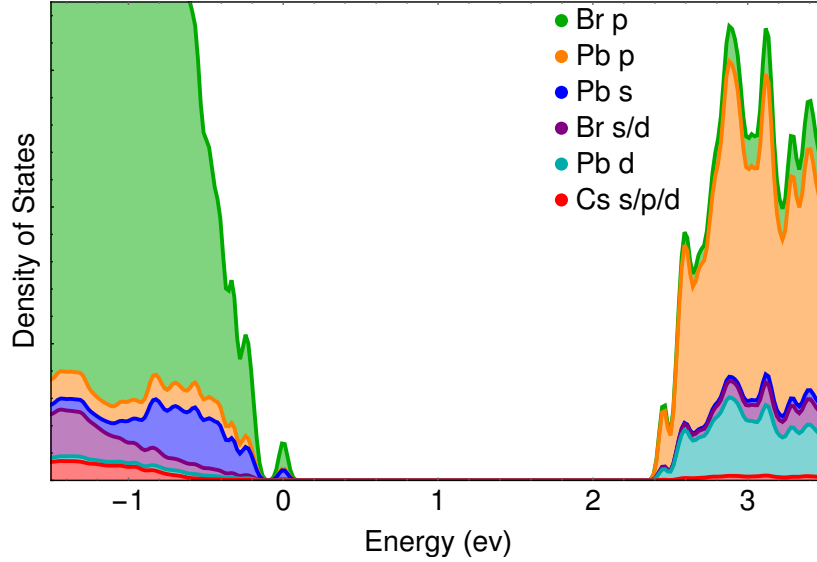


Figure 14: Projected density of states for a $l = 2.64$ -nm model. The Fermi energy is shifted to 0 eV. The VB edge is dominated by Br p -orbital and Pb s -orbital antibonding character. The CB edge results mainly from coupling of Pb p -orbitals.

5.2 Spin-Orbit Coupling

The spin-orbit interaction between CHS→CBES spin states: singlet spin $m_s=0$ ($|S_{m_s=0}\rangle$), triplet $m_s=0$ ($|T_{m_s=0}\rangle$), and triplets $m_s=\pm 1$ ($|T_{m_s=\pm 1}\rangle$) was computed by employing the Breit-Pauli Hamiltonian[7–9] within a small matrix approximation calculate the spin-orbit (SO) coupling between relevant spin states:

$$\hat{H}_{SO} = -\frac{\alpha_0^2}{2} \sum_{i,A} \frac{Z_A}{r_{iA}^3} (\mathbf{r}_{iA} \times \mathbf{p}_i) \cdot \mathbf{s}_i \quad (1)$$

where i denotes electrons, A denotes nuclei, $\alpha_0 = 137.037^{-1}$ is the fine structure constant. Z_A is the bare positive charge on nucleus A . s_i represents the spin of electron i . The term $\mathbf{r}_{iA} \times \mathbf{p}_i$ denotes the angular momentum operator of electron i calculated with respect to nucleus A at the position R_A , with r_{iA} being the distance between them. A basis of spin-adapted many electron states was built from excitations between the CHS and CBES, and the matrix elements of the Breit-Pauli Hamiltonian between these spin-adapted states were calculated by numerical quadrature and the usual Slater-Condon rules. The resulting Hamiltonian was diagonalized to obtain SO coupled fine structure states. The wavefunctions themselves are dominated by low-angular momentum waves, leading to very small fine-structure splitting and coupled states which are close to spin-eigenfunctions.

5.3 Exciton Binding Energies

The Coulombic binding energy between a hole in the CHS and an electron in the lowest CB state ranges from 148 meV in the $l = 2.64$ nm model to 30 meV in the $l = 3.82$ nm model. This compares favorably with a 40 meV exciton binding energy determined using an effective mass approximation.[10] Such small binding energies validate our assumption that the effect of polaronic lattice relaxation on the Stokes shift is small compared to the electronic degrees of freedom.

References

- (1) Tosco, P.; Stiefl, N.; Landrum, G. *J. Cheminform.* **2014**, *6*, 37.
- (2) NIST Precomputed vibrational scaling factors, <http://cccbdb.nist.gov/vibscale.asp>, Accessed: 2017-12-23.
- (3) *Infrared Spectra*, NIST Mass Spectrometry Data Center In *NIST Chemistry WebBook, NIST Standard Reference Database Number 69*, Linstrom, P., Mallard, W., Eds., 2018.
- (4) Dahlke, E. E.; Olson, R. M.; Leverentz, H. R.; Truhlar, D. G. *The Journal of Physical Chemistry A* **2008**, *112*, 3976–3984.
- (5) Ten Brinck, S.; Infante, I. *ACS Energy Letters* **2016**, *1*, 1266–1272.
- (6) Brandt, R. E.; Stevanović, V.; Ginley, D. S.; Buonassisi, T. *Mrs Communications* **2015**, *5*, 265–275.
- (7) Bethe, H. A.; Salpeter, E. E., *Quantum Mechanics of One- and Two-Electron Atoms*; Springer: 1957.
- (8) Sayfutyarova, E. R.; Chan, G. K.-L. *J. Chem. Phys.* **2016**, *144*, 234301.
- (9) Shao, Y. et al. *Mol. Phys.* **2015**, *113*, 184–215.
- (10) Protesescu, L.; Yakunin, S.; Bodnarchuk, M. I.; Krieg, F.; Caputo, R.; Hendon, C. H.; Yang, R. X.; Walsh, A.; Kovalenko, M. V. *Nano Letters* **2015**, *15*, 3692–3696.