Analysis of Local Molecular Structure in Liquid Water Using Machine Learning

Owen Lockwood*

Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

E-mail: lockwo@rpi.edu

Phone: (970) 231 5672

Introduction

Water plays an important role in mediating a number of biological and molecular phenomena.[1–3] These interactions are often dictated by water's structure. Critical to the study and analysis of these phenomena are molecular dynamics (MD) simulation. These simulations are often temporally expensive to design, run, and analyze. Although simulations produce trajectories on the order of nanoseconds, they produce an enormous amount of data; data that is typically analyzed through physics-based calculations. In the case of water simulations, this analysis reveals insight into local structural patterns of water that influence and dictate the many biological phenomena. There are a number of important parameters used to describe and analyze water, the two focused on in this work are cavity formation and tetrahedral order parameter. The use of cavity formation in the analysis of molecular dynamics simulations has shown a great degree of success and has a number of applications.[4–7] Tetrahedral order parameter, first theorized by Chau and Hardwick, has also seen great success in the analysis of molecular simulations.[9–12]

Parallel to the success of molecular dynamics simulations is the impressive developments in recent machine learning research. From achieving impressive levels of performance in difficult tasks ranging from games,[13,14] to linguistics,[15] to music creation,[16] to image analysis and object detection.[17] The three main forms of machine learning are supervised, unsupervised and reinforcement learning, all of which have achieved a great deal of success. Supervised learning is most relevant to this work, as labelled and well structured data is common in molecular simulations. Recently, the partnership between machine learning and the biological and physical sciences has grown considerably to the benefit of both parties. Recent prominent work in the application of machine learning to biological and physical sciences include drug discovery,[18,19] and protein folding.[20] While this partnership is young, it is already proving to be fruitful.

This work utilizes machine learning to analyze the local structure in liquid water simulations. Building upon the work and model presented in DeFever et al., which utilized machine learning for phase analysis, I create machine learning models that are capable of learning local structural order parameters for water. Specifically, two models are created that are capable of learning cavity size and tetrahedral order parameter. This work also shows that a single model with significant shared weights can learn both parameters simultaneously. Finally, it is demonstrated that training artificially (i.e. pseudo-randomly) generated data can perform comparably and potentially better than training on actual simulation data.

Methods

Simulation and Generation of Data

The core simulations for this work were done using GROMACS.[22] Three simulations were run, one at 274 K, one at 298 K and one at 320 K. Other than the temperature difference, the simulations were all run with the same parameters, 50 nanosecond simulation NPT simulation of 4,142 SPC/E[23] water molecules in a 5 nm box. These simulations were analyzed to provide training data and testing data. In order to get the cavity formation training and testing data, gridpoints were sampled from the simulation and the cavity size was calculated by finding the distance to the nearest oxygen atom: $min(\sqrt{(x_{gp} - x_O)^2 + (y_{gp} - y_O)^2 + (z_{gp} - z_O)^2})$. The tetrahedral order parameter was calculated using the formulation from Errington and Debenedetti, specifically, $q = 1 - \frac{3}{8}\sum_{i=1}^{3}\sum_{j=i+1}^{4}(cos(\phi_{ij}) + \frac{1}{3})^2$. For every oxygen atom, the nearest 4 oxygens were found and the angles between them calculated. Then using this formula tetrahedrality of that oxygen atom can be determined, where q = 1 is a perfect tetrahedron.

Model Architecture

This work utilizes two related machine learning models. Each model takes as input a two dimensional array of (x,y,z) coordinates. The general idea these models exploit is the concept of feature extraction. Rather than giving giving the neural network information that it would need to calculate physical properties,[25] the model must learn what the important features are from the coordinates. Each model has a feature extraction base, a set of neural network layers that increase the dimensionality of the data to extract all important information. This is then fed into the prediction head, which a set of layers that decrease in dimensionality, outputing a single number for the prediction. The first model type is called Single Prediction Model (SPM). The SPM model architecture is functionally similar to the architecture presented in DeFever et al. which is a simplified version of the model architecture presented in Qi et al., differing only slightly in hyperparameters. See Figure 1 for a diagram of the model architecture.

I also present the Multi-Prediction Model (MPM), a multiheaded version of the SPM that has shared weights in feature extraction base and two separate prediction heads. See Figure 2 for a diagram of MPM architecture. This is a divergence from the previously referenced architectures and has a significant potential for direct usage, as the concept of a pretrained shared base is common in other fields of machine learning and may be constructed and shared between MD researchers. This will be addressed further in the Results and discussion.

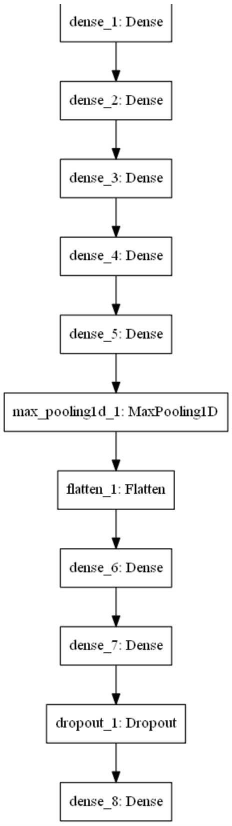

Figure 1: SPM Architecture

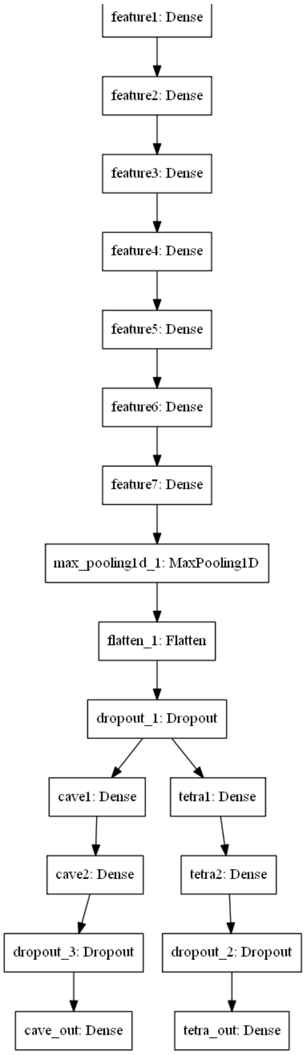

Figure 2: MPM Architecture

Model Training

All models were constructed and trained using TensorFlow.[27] SPM was trained on about 500,000 sets of points with associated labels (i.e. what the value actually is). This training data was exclusively from the 298 K simulation. The mean squared loss is calculated from the predicted value to the actual value, $loss = \frac{1}{N}\sum_{i=0}^{N}(y_{true} - y_{pred})^2$. Gradients were then calculated and applied to the network using the Adaptive Momentum Estimation (ADAM) optimizer.[28] The size of the input varies depending on the task, the input is a 10 by 3 array (that a set of the 10 nearest oxygens) for cavity formation and 4 by 3 array (a set of the 4 nearest oxygen atoms) for tetrahedral order parameter. The model is also successful is a cutoff is applied and the array is zero padded. The input coordinates are centered to zero to ensure the model learns the locational independence of the features. The data is not augmented (unlike DeFever et al., which introduces rotational augmentation) unless otherwise stated, and its success demonstrates that augmentation is not always necessary. The output of each model is a single value representing the cavity size or tetrahedral order parameter. Batch size was 32 and the number of epochs varied between 20-50 (early stopping was used if the model failed to improve). Each model can be trained in a relatively short amount of time (10-30 minutes wall clock time for a single Nvidia GTX 1070 GPU).

The training procedure for the MPM was slighly different. The model architecture is very similar to the previous model. It relies on the same concept of a feature extraction base, it just takes that array that results from pooling and feeds it to two network heads, one for tetrahedrality and one for cavity. The loss and optimizers and other model attributes are the same as before. However, after discovering that the model succeeded with learning tetrahedrality but failed to accurately learn cavity, the importance of loss weighting became apparent. Because tetrahedrality is usually a larger number than cavity size, it produces a larger loss. E.g. if the model is off by 100% for both tetrahedrality and cavity, that would produce a TOP loss of 0.16 (if the actual parameter is 0.8 and the prediction was 0.4), whereas cavity would produce a loss of 0.0001 (if the actual was 0.02 nm and the prediction was 0.01). Thus, when the gradients are calculated and applied the tetrahedrality influences the training significantly more. To account for this, adjustments were made to the loss weighting. The cavity loss is considered 85x the tetrahedral loss to balance it out. This weight significance is arbitrary and was determined experimentally.

Results and Discussion

SPM Results

For the SPM, both cavity formation and tetrahedral order parameter achieve a high degree of success. Figure 2 represents the results for cavity prediction and tetrahedral order parameter. In both cases, the graphs represent the real distribution on a data set of about 400,000 points never before seen by the model, vs what the model actually predicts. These points are from a different time in the simulation and the model has never been trained on them. These graphs clearly demonstrate the success of the model. Note too how the model is able to capture the essential characteristics of the graphs. For example, with the tetrahedral order parameter graph, the real bimodality of the distribution is learned by the model. Figure 3 shows the difference in cavity size predictions and Figure 4 shows the difference in tetrahedral order predictions.

These results can be visualized in a different manner, as 'heatmaps' to see where the differences are inside the actual simulation. These are constructed by taking a very thin 'slice' out of the middle of the simulated box, to reduce one sheet to a single 2D image. Figures 5 and 6, and 7 and 8 represent the actual heatmaps and the predicted heat maps for cavity size and tetrahedral order parameter, respectively.

To more clearly highlight the differences and where exactly the model mis-predicts, see Figure 9 for cavity error and Figure 10 for tetrahedral order parameter error. These represent the exact same points as the figures above, but the scale represents the magnitude of the error. This error helps to reveal where the model might not predict perfectly. Periodic

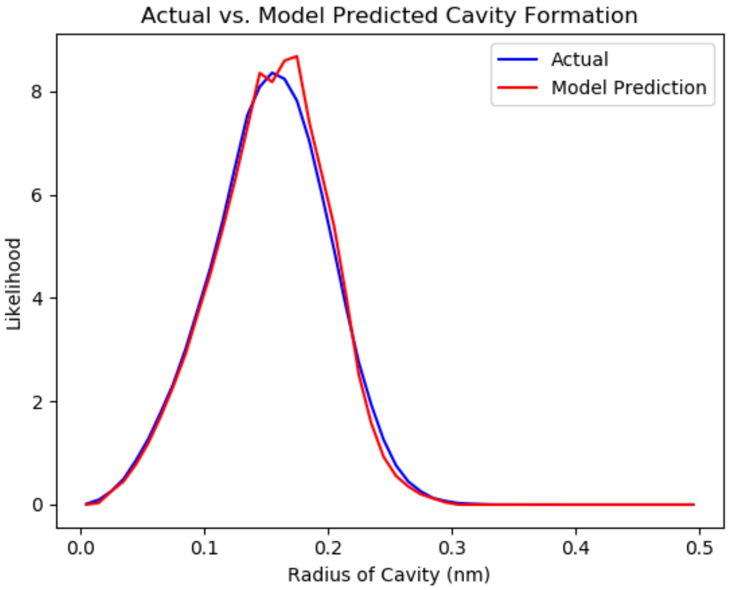

Figure 3: Cavity Size Comparison

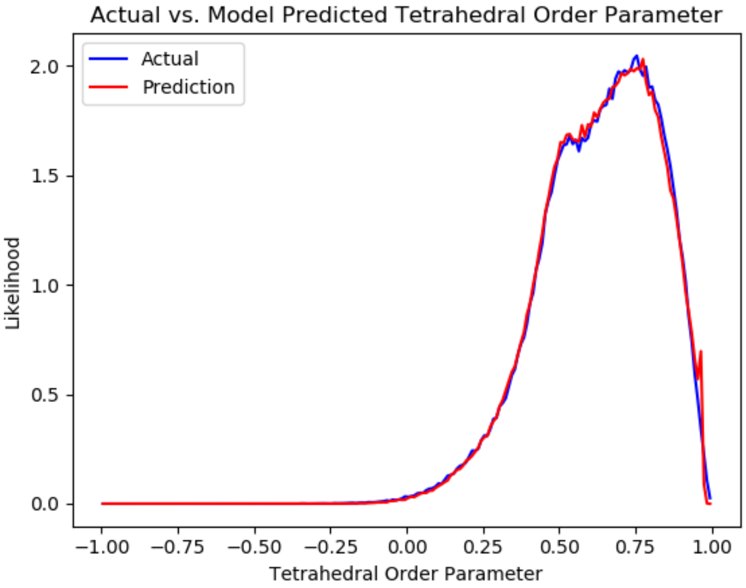

Figure 4: Tetrahedral Order Parameter Comparison

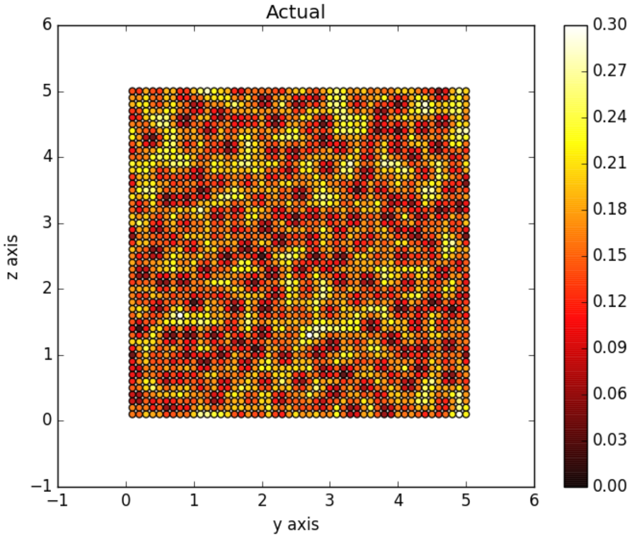

Figure 5: Actual Cavity Size

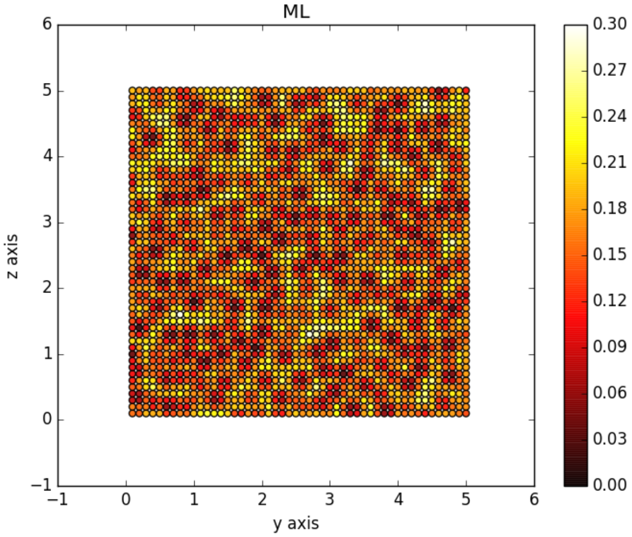

Figure 6: Predicted Cavity Size

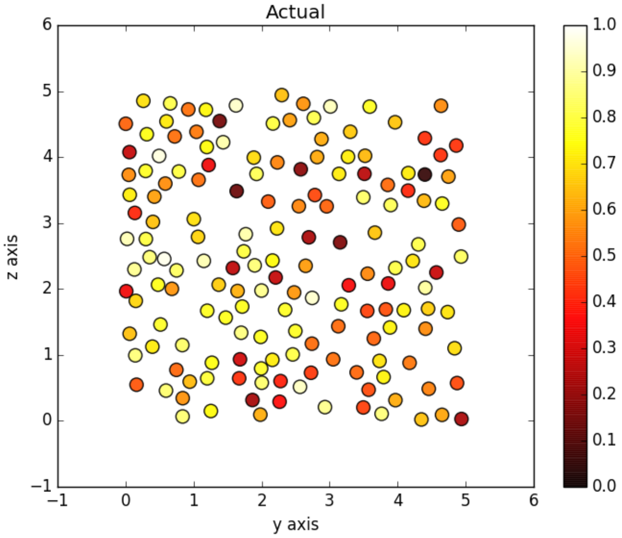

Figure 7: Actual Tetrahedral Order Parameter

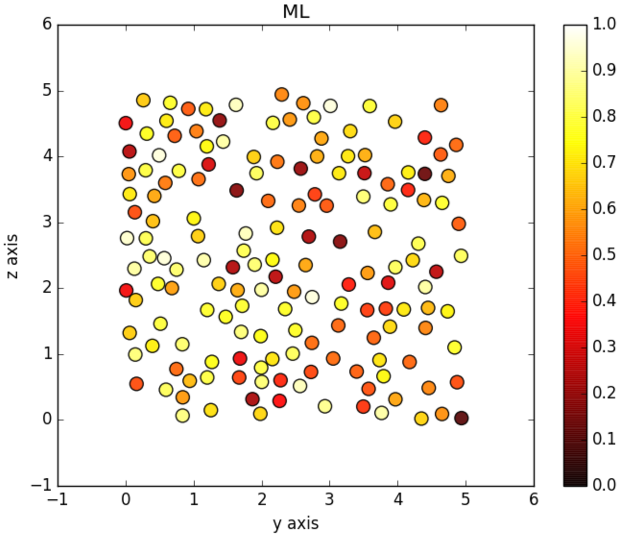

Figure 8: Predicted Tetrahedral Order Parameter

boundary conditions are not removed in the data fed into the network. This means that the network must learn periodic boundary conditions. This was done intentionally, to reduce the human involvement in data preparation as much as possible and to try to test the network in the most robust way possible. Removing the PBC does not require a retraining of the model, our goal was to make the model PBC and non PBC friendly, which succeeded to a certain extend. Although the errors are largely at the edge, the magnitude of the errors is relatively small and not all edges have errors.

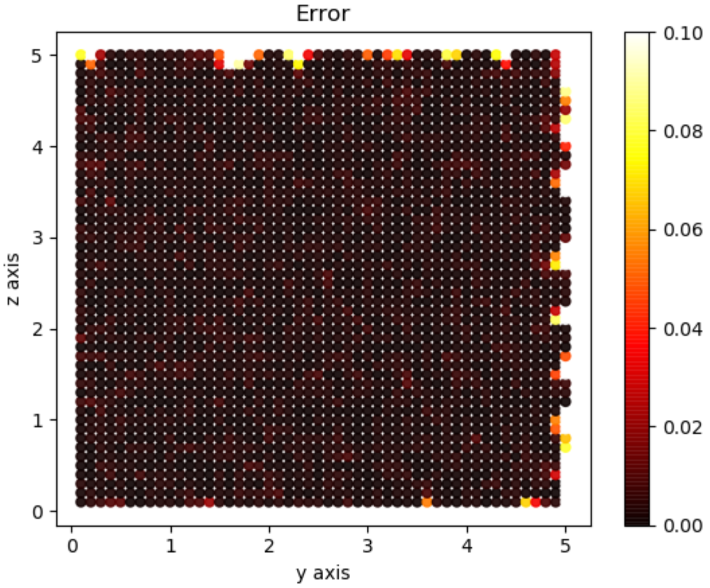

Figure 9: Cavity Size Error

Although the testing data was completely unseen by the model, in order to verify that the model was learning in a sufficiently general way, thet tetrahedral SPM was tested on simulations from other temperatures. See Figure 11 and 12 for the the comparison of actual vs model prediction of tetrahedral order parameters for 274 K and 320 K respectively. Because this model was only trained on 298 K simulation, the success it achieves on the other temperatures shows the model's generalized knowledge within the domain of water structure.

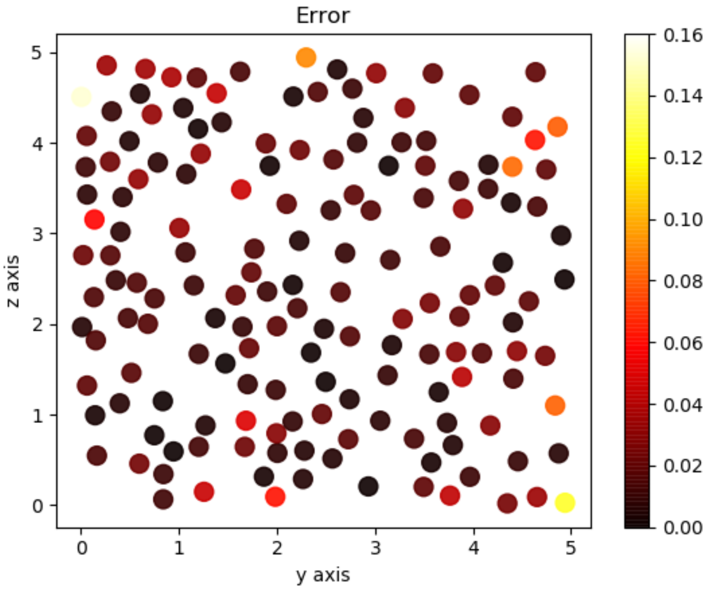

Figure 10: Tetrahedral Order Parameter Error

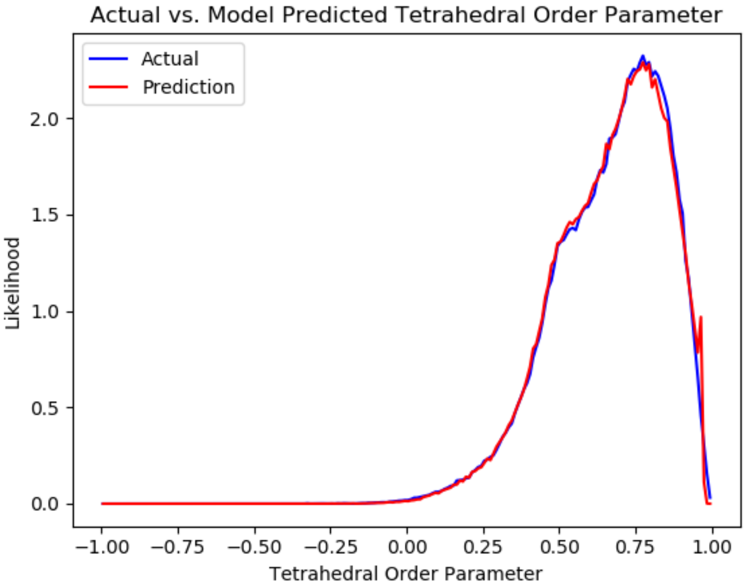

Figure 11: Tetrahedral Order Parameter at 274 K Comparison

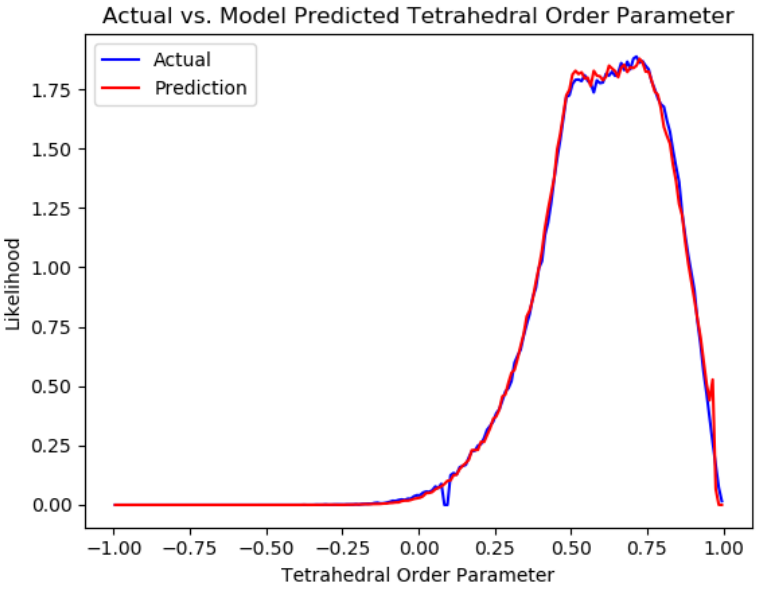

Figure 12: Tetrahedral Order Parameter at 320 K Comparison

MPM Results

MPM also achieved a high degree of success. However, this did require more samples to achieve this level of success. In order to increase the diversity of cavity formation size (as the model was just learning to guess the single number as that due to the tight variance), data augmentation was introduced. Note first, that the cavity data was not the same as true ’cavity formation’. Because the input into the mode had to be coordinates of an oxygen atom, what the ’cavity size’ really shows distribution of O-O distances, which has a much lowered variance. The type of augmentation is atypical for machine learning and capable only in a well-defined physical system, perhaps a better description would be data supplementation. Data augmentation usually relies on modifying the input data in some way so it is numerically different, but functionally the same (e.g. rotating a picture). In this model, the data augmentation was artificially generated data (which will be discussed more in depth in the discussion). Using 400,000 real coordinates with (tetrahedral, cavity) labels, the model was suboptimal. I randomly generated 200,000 points between 0 and 0.7 nanometers (skewed toward <0.25) and calculate the (tetrahedral, cavity) for them. This gives greater diversity to the cavity size. This also impacts the tetrahedral order parameter diversity but does not negatively impact the results. Important to note, these numbers are not reflective of reality and no attempt is made to have them be reflective of the reality of water interactions (e.g. if two points were right on top of each other, which physically wouldn't be allowed, this would be acceptable as physcial laws are not enforced onto the data generation). This increased cavity diversity yielded significantly improved results for cavity, while having little effect on tetrahedrality. Note that again whatever is done to the training data is not done to the testing. All results are on real data, completely unaugmented from a part of a simulation the model has never before seen.

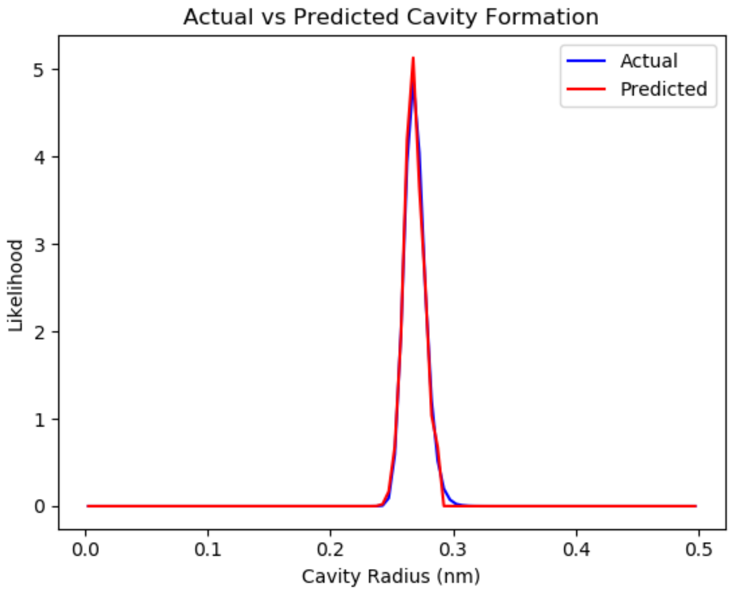

Figure 13: MPM Cavity Prediction

Data Augmentation

Mentioned in the previous section, the use of data augmentation shows promise in improving model results. Here I consider entirely 'augmented data'. That is data purely generated from a pseudo random number generator, rather than data from real simulations. Generating random coordinates (skewed towards the real distribution of coordinates), a SPM model is

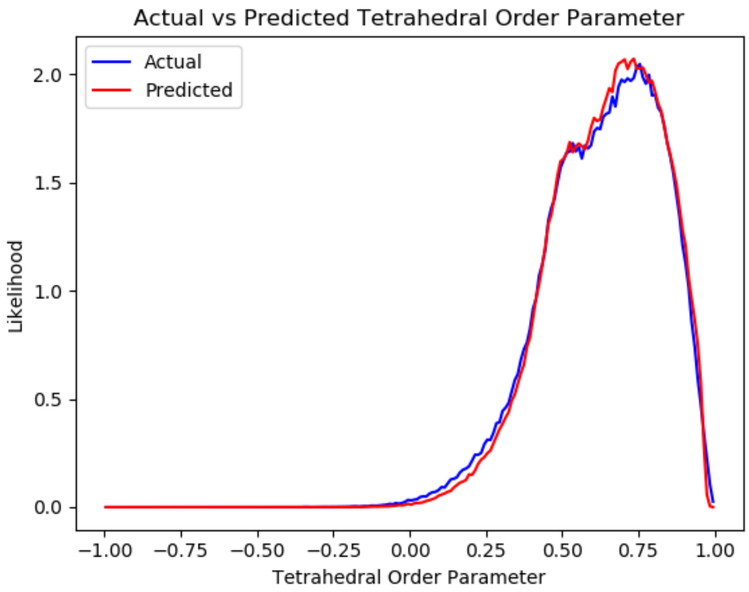

Figure 14: MPM Tetrahedral Order Parameter Prediction

trained to predict cavity size trained only on this ’fake’ data. The model is then test it on real data, as seen in Figure 15. This model performs as well, if not better, than the previous model trained on real data.

Discussion

Machine learning is often not an easy task to begin de novo. Acquiring the quality data, the computational resources and the team to design and implement algorithms is not accessible to everyone. Hence, researchers often make ‘base models’ that are pretrained for a general task and can then be used directly, or as a base for more machine learning for a more specific domain. In natural language processing GPT-3[15] is a model too large for most to train, but very useful for many to use pretrained, other models that come pretrained and can be used by the public as is or for domain specification. Other examples of these pretrained bases include word2vec,[29] and Universal Language Model Fine Tuning for Text Classification (ULMFiT),[30] and in computer vision, VGG16[31] is available. Domain specification means adding complexity to the model to make it more suited to a specific task it was not origi-

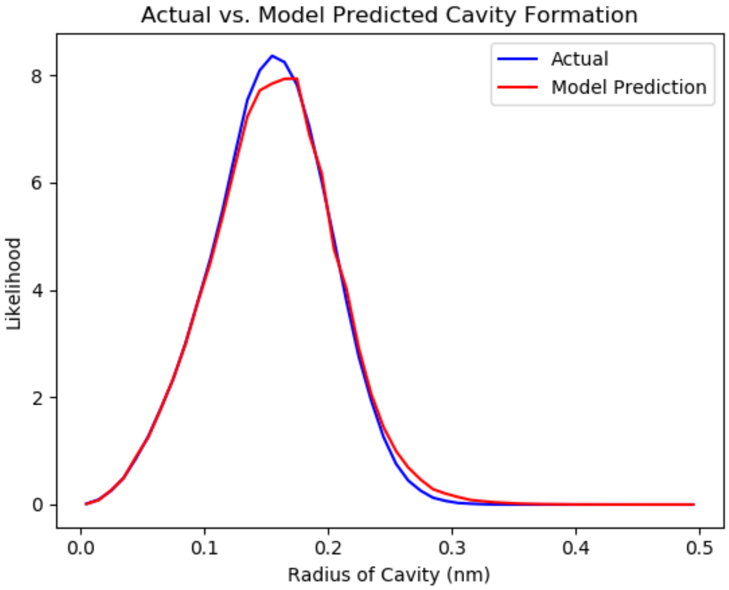

Figure 15: Artificially Generated Data

nally designed for. E.g. using word2vec as an embedding basis, specific to analyze tweets. Word2vec was not designed specifically with twitter in mind, but using this computational basis, one could train additional operations with this purpose in mind. Here the pragmatic value of the MPM becomes apparent. This work demonstrates the potential for a pretrained base. By demonstrating how a single base can be used for both tetrahedral order parameter and cavity formation, this work lays the groundwork for more complex bases to be trained and provided. A single feature extraction base could be trained then domain specified to a researcher's particular need. This is now clearly possible.

These results further indicate the potential and importance for a growing relationship between machine learning and the scientific community, especially the molecular simulation community. These results, in addition to DeFever et al., indicate the vast potential for using machine learning to analyze molecular simulations. Although the success of the results is important, one discovery I found especially interesting was the improvements made with the 'fake' data augmentation. It is called 'fake' only in the sense that if it reflects reality, it is a mere coincidence and adheres to no physical laws. It is near randomly generated. The success of both the data augmentation strategy and the purely ‘fake’ data strategy indicate that this is very important as it could enable a massive speedup in the application of machine learning to analysis. The number of simulations run could be reduce (or potentially entirely eliminated) for the training process. This would allow massive speedup, especially for very complex simulations. The only requirements for this are that the results or the goal of training can be analyzed purely from coordinate points (although perhaps in the future more creative data generation will be created) and that the generation of these points is no more computationally expensive than the actual simulation. Although water simulations are not extremely complex, this concept might be able to be applied to significantly more complex problems, greatly reducing the time complexity of the creation for training data.

Conclusion

I performed MD simulations of water to train multiple machine learning models to analyze two critically important features, cavity formation and tetrahedral order parameter. Our techniques use DeFever et al. as a departure point and build upon it to create unique models with important potential pragmatic use. This work then provides the idea for an interesting methodology for creating ‘fake’ data to either supplement or replace the traditional training data. All results are indicative of success and the model preforms to a high degree. This aligns with what previous works in MD and in machine learning indicate. For the MPM, I hope that laying this groundwork will enable other researchers to create pretrained models that will be extremely useful to molecule dynamics researchers with or without experience in machine learning, as is common in other fields of machine learning. Although the ’fake’ data generation was just breifly introduced, I hope that this concept can be applied to many researchers’ projects. Due to the well structure physical nature nature of machine learning for MD, the creation of random data supplementation/augmentation is a very useful notion.

These approaches should be useful and generalizable to other problems, and I hope to see trained basis including other parameters, such as Steinhardt order parameters,[32] and others. This work is applicable outside just water analysis and can be used in other forms of analysis as the coordinate handling, the multiheaded approach and base, and the data supplementation could all be of great use to any well-defined physical analysis of molecular dynamics simulations.

Acknowledgement

The authors thank Shekhar Garde, Camille Bilodeau, Mayank Vats and Imee Sinha for their assistance in this work.

References

(1) Ball, P. Water is an active matrix of life for cell and molecular biology. *Proceedings of the National Academy of Sciences* **2017**, *114*, 13327–13335.

(2) Levy, Y.; Onuchic, J. N. Water mediation in protein folding and molecular recognition. *Annu. Rev. Biophys. Biomol. Struct.* **2006**, *35*, 389–415.

(3) Papoian, G. A.; Ulander, J.; Wolynes, P. G. Role of water mediated interactions in protein- protein recognition landscapes. *Journal of the American Chemical Society* **2003**, *125*, 9170–9178.

(4) Godawat, R.; Jamadagni, S. N.; Garde, S. Characterizing hydrophobicity of interfaces by using cavity formation, solute binding, and water correlations. *Proceedings of the National Academy of Sciences* **2009**, *106*, 15119–15124.

(5) Xi, E.; Venkateshwaran, V.; Li, L.; Rego, N.; Patel, A. J.; Garde, S. Hydrophobicity of proteins and nanostructured solutes is governed by topographical and chemical context. *Proceedings of the National Academy of Sciences* **2017**, *114*, 13345–13350.

(6) Hummer, G.; Garde, S.; Garcia, A.; Paulaitis, M. E.; Pratt, L. R. Hydrophobic effects on a molecular scale. 1998.

(7) Postma, J. P.; Berendsen, H. J.; Haak, J. R. Thermodynamics of cavity formation in water. A molecular dynamics study. Faraday Symposia of the Chemical Society. 1982; pp 55–67.

(8) Chau, P.-L.; Hardwick, A. A new order parameter for tetrahedral configurations. *Molecular Physics* **1998**, *93*, 511–518.

(9) Duboue-Dijon, E.; Laage, D. Characterization of the local structure in liquid water by various order parameters. *The Journal of Physical Chemistry B* **2015**, *119*, 8406–8418.

(10) Kumar, P.; Buldyrev, S. V.; Stanley, H. E. A tetrahedral entropy for water. *Proceedings of the National Academy of Sciences* **2009**, *106*, 22130–22134.

(11) Pereyra, R. G.; di Lorenzo, A. J. B.; Malaspina, D. C.; Carignano, M. A. On the relation between hydrogen bonds, tetrahedral order and molecular mobility in model water. *Chemical Physics Letters* **2012**, *538*, 35–38.

(12) Radhakrishnan, R.; Trout, B. L. *Handbook of Materials Modeling*; Springer, 2005; pp 1613–1626.

(13) Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M., et al. Mastering the game of Go with deep neural networks and tree search. *nature* **2016**, *529*, 484.

(14) Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P., et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* **2019**, *575*, 350–354.

(15) Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A., et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* **2020**,

(16) Dhariwal, P.; Jun, H.; Payne, C.; Kim, J. W.; Radford, A.; Sutskever, I. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341* **2020**,

(17) Cai, Z.; Fan, Q.; Feris, R. S.; Vasconcelos, N. A unified multi-scale deep convolutional neural network for fast object detection. European conference on computer vision. 2016; pp 354–370.

(18) Kadurin, A.; Aliper, A.; Kazennov, A.; Mamoshina, P.; Vanhaelen, Q.; Khrabrov, K.; Zhavoronkov, A. The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology. *Oncotarget* **2017**, *8*, 10883.

(19) Lavecchia, A. Machine-learning approaches in drug discovery: methods and applications. *Drug discovery today* **2015**, *20*, 318–331.

(20) Senior, A. W.; Evans, R.; Jumper, J.; Kirkpatrick, J.; Sifre, L.; Green, T.; Qin, C.; Žídek, A.; Nelson, A. W.; Bridgland, A., et al. Improved protein structure prediction using potentials from deep learning. *Nature* **2020**, 1–5.

(21) DeFever, R. S.; Targonski, C.; Hall, S. W.; Smith, M. C.; Sarupria, S. A generalized deep learning approach for local structure identification in molecular simulations. *Chemical science* **2019**, *10*, 7503–7515.

(22) Pronk, S.; Páll, S.; Schulz, R.; Larsson, P.; Bjelkmar, P.; Apostolov, R.; Shirts, M. R.; Smith, J. C.; Kasson, P. M.; van der Spoel, D., et al. GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. *Bioinformatics* **2013**, *29*, 845–854.

(23) Berendsen, H.; Grigera, J.; Straatsma, T. The missing term in effective pair potentials. *Journal of Physical Chemistry* **1987**, *91*, 6269–6271.

(24) Errington, J. R.; Debenedetti, P. G. Relationship between structural order and the anomalies of liquid water. *Nature* **2001**, *409*, 318–321.

(25) Geiger, P.; Dellago, C. Neural networks for local structure detection in polymorphic systems. *The Journal of chemical physics* **2013**, *139*, 164105.

(26) Qi, C. R.; Su, H.; Mo, K.; Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017; pp 652–660.

(27) Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M., et al. Tensorflow: A system for large-scale machine learning. 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16). 2016; pp 265–283.

(28) Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* **2014**,

(29) Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* **2013**,

(30) Howard, J.; Ruder, S. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146* **2018**,

(31) Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* **2014**,

(32) Steinhardt, P. J.; Nelson, D. R.; Ronchetti, M. Bond-orientational order in liquids and glasses. *Physical Review B* **1983**, *28*, 784.