INTRODUCING CARATE: FINALLY SPEAKING CHEMISTRY.

A PREPRINT

Julian M. Kleber*
Freie Universität Berlin

Stettiner Straße, 13357 Berlin
julian.kleber@fu-berlin.de

February 16, 2022

ABSTRACT

Computer-Aided Drug Design is advancing to a new era. Recent developments in statistical modelling, including *Deep Learning*, *Machine Learning* and high throughput simulations, enable workflows and deductions not achievable 20 years ago. The key interaction for many small molecules is via bio-molecules. The interaction between a small molecule and a biological system therefore manifests itself at multiple time and length scales. While the human chemist quite intuitively grasps the concept of multiple scales, most of the computer technologies do not relate multiple scales easily. Therefore, numerous methods in the realm of computational sciences have been developed. However, up to now it was not clear that the problem of multiple scales is not only a mere matter of computational abilities but even more a matter of accurate representation. Because of the amount of already performed simulations at various scales, deep learning (DL) becomes a viable approach to speed up computations by approximating simulations in a data-driven way. However, to accurately approximate the physical (simulated) properties of a given compound, an accurate, uniform representation is mandatory. Therefore, the *biochemical and pharmaceutical encoder* (CARATE) is introduced. Furthermore, the regression and classification abilities of CARATE are evaluated against benchmarking datasets (ZINC, ALCHEMY, MCF-7, MOLT-4, YEAST, Enzymes, Proteins) and compared to other baseline approaches.

Keywords Multi Head Self Attention, Graph Neural Networks, Regression, Quantum Chemistry, Computer Aided Drug Design, Computational Science, Deep Learning

1 Introduction

The wish to accurately predict the outcome of a drug development process is as old as the regulated pharmaceutical industry itself. However, the task of simulating a drug interaction in the body is a multiscale problem operating on many scales. Up to now, no accurate method has been found to model the whole life cycle of an API inside a mammal. The work on multiscale problems focuses on different aspects of the systems at multiple scales and tries to represent the important aspects of a particular scale accurately by the means of that particular scale. The simulator decides on unimportant parts of a scale and approximates them more efficiently with methods of a larger scale[1].

When different length and time scales are involved, computational expensive methods for the small length scale may become infeasible. For example, to model the interaction of a small molecule with a receptor, one needs to combine quantum chemistry (QC) with classical molecular mechanics (MM). The computations performed by the QM/MM approximation still have the QC calculations as a bottleneck. However, recent advances in deep learning make the prediction of a particular simulation scale accessible[2, 3]. Also, the prediction of quantum chemical properties with deep learning methods are accessible[4]. To speed up the computation, of a QM/MM simulations, the QC computations should be approximated by predicting the desired properties via machine learning methods[5].

*Master Student at Freie Universität Berlin

Up to now, however, to the best of my knowledge no algorithm with short training times and competitive prediction abilities has been reported. However, when the training of accurate predictive models required severe computation, then for investigating new interactions between small molecules and bio molecules they are not really suitable as the model may have to adapt to new circumstances frequently.

2 Related Work

Predicting physical properties of chemical compounds is indeed a well-known topic in the literature. Recent work comparing the representations of molecules for the QM-9 prediction performance of a feed forward neural network (FNN) indeed showed that the quality of predictions of physical properties highly depend upon the molecular representation. [6]. Moreover, the recent advances in natural language processing (NLP) introduced by the self attention mechanism in bidirectional auto encoders[7] showed that respective algorithms (BERT technology) can learn semantic properties of the input.[8]. Especially, the BERT algorithm has abilities to organize and structure information.[9]. Because molecular descriptors can be learned in a data-driven way[10] the self-attention mechanism made room for innovation around self-attention deep neural networks in computational chemistry working on the SMILES or similar text representations of molecules with good classification and regression results[11].

A fundamentally different, yet equally promising approach to a text-based encoding is to use numerical representations such as fingerprints, substructures, or graphs. Indeed, there is a recent study investigating the potential of unified fingerprints for small and biomolecules[12]. However, one-shot-learning based on self-attention LSTM with encoded molecular graphs as inputs showed more decent results[13]. Many approaches to model the molecular structure with graph neural networks (GNNs) have been applied. To increase computational efficiency, a self-attention mechanism has been rudimentary applied to graph convolutional networks as well (GCNs)[11]. Another way to achieve better structure recognition is to apply message passing networks[14]. Moreover, multi-headed self attention has been applied to GNN encoded molecular structures through a self attention mechanism[15]. This work completes the series on the subsequent of attention-based algorithms to chemical problems. This work investigates a multi-headed graph self attention model (GAT)[16] as an encoder for classification and regression tasks, with significant results.

3 Methods

Both methods use the basic encoder structure and apply specific architectures on top of the encoder structure for each specific task. The training was performed in batches with no cross validation to model the low data regime within a multiscale simulation properly. Unless otherwise stated, each encoder combined graph convolutional network (CARATE-GCN) was trained with the encoder for 50 steps. The two GAT layers with a dropout probability of 0.6 a GCN and two linear layers were stacked on top of the GAT architecture. Unless stated otherwise, a batch size of 64 samples was applied. The datasets were obtained from TUDataset[17], and MoleculeNet.[18] Each dataset was used as is, to see how CARATE performs on raw data. The only preprocessing applied was rescaling to the target's unit, as would be done during a simulation.

If the data set was given in SMILES format, the SMILES format was converted to a molecular graph representation implemented in PytorchGeometric.[19] Moreover, all GAT and GNN implementations were applied from or achieved with the PytorchGeometric[19] library. Other deep learning building blocks were implemented with help of the PyTorch[20] and Numpy[21] library.

The CARATE network consists of an input layer consisting of a GCN and GAT layer. To use the encoder in regression and classification tasks, another GCN layer followed by task-specific layers are added to the stack (compare Figure 1)

To test the encoding to property relationship, the encoder-predictor model was tested against physical regression tasks on the ALCHEMY[22] and ZINC data set obtained from TUDatasets[17]. For each task, no more than 50 training iterations are necessary for a good fit. The convergence is thus quite fast. The desired targets were normalized to ensure smooth convergence behaviour during training. The architecture for the regression tasks differs from the classification to obtain multitask regression results, as opposed to multitask classification only by the output function applied to the network. The results are compared to GIN-ϵ.[23, 24] The GINE-ϵ network uses batch normalization as well as zero mean and unit variance normalization.

To test the classification abilities of the model we tested it against the drug discovery datasets Enzymes, Proteins, YEAST, MOLT-4, MCF-7 obtained from the TUDataset without any further preparation (Table 2). For each example, five-fold stratified cross-validation for statistical significance of the results was performed. The number of hidden units was reduced for the Proteins and Enzymes dataset to avoid overfitting, as the datasets are quite small and may not be really descriptive for the classification problem they are intended for.

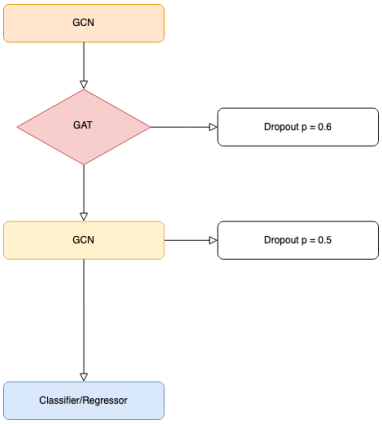

Figure 1: General architecture of the CARATE network used for classification and regression tasks.

Table 1: Regression performance compared to baseline GINE-ϵ[23].

Method	ZINC	ALCHEMY
CARATE	0.0036	0.00108
GINE-ϵ[23]	0.084	0.103

4 Results

The model yields good results, indicating that the GAT-Convolutional neural network can classify molecules with empirical data accurately. The model does not require long training times, and in fact reaches satisfactory convergence usually after less than 500 iterations. Moreover, the model is prone to overfitting even when reducing the number of units in the hidden layer for the Enzymes and Proteins dataset. For the Enzyme dataset the performance is limited to overfitting, but the accuracy is around 70%. The Proteins dataset is fitted nicely with 83% accuracy. The test accuracy for the Proteins dataset is shown in Figure 3 and emphasizes that early stopping after around 50 iterations yields a good fit. The variance for the Proteins dataset, however, is quite high. The MCF-7 dataset, in contrast, is not fitted nicely with an accuracy of just 93.6%. The fit of MCF-7 is better than most comparison algorithms, except the 1-WL algorithm[25]. The MOLT-4 dataset shows similar behaviour to the MCF-7 dataset but is fitted with higher accuracy than most reference methods (compare Table 2). Moreover, the variance for fitting the MCF-7 dataset is quite low. The training accuracy shown in Figure 2 shows slow convergence with low variance.

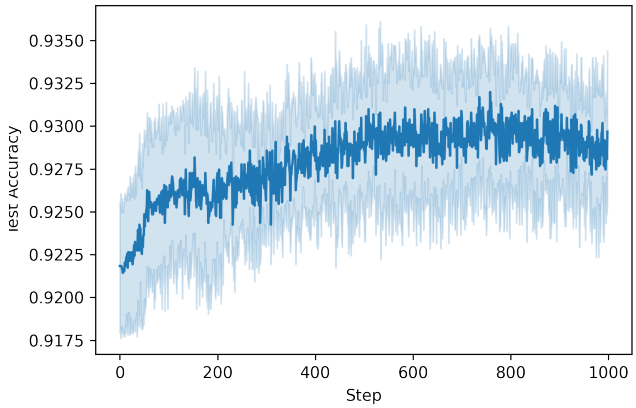

Figure 2: Test accuracy over the course of training of CARATE for the MOLT-4 dataset.

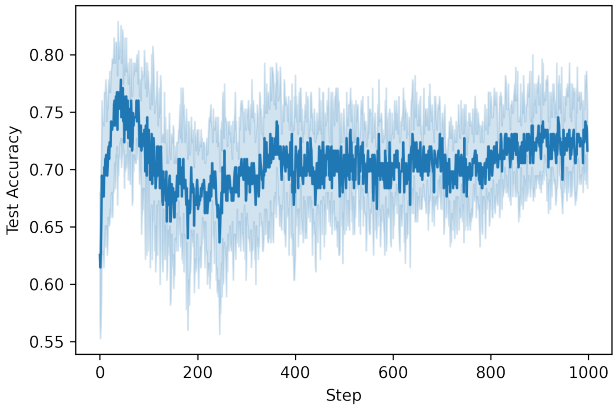

Figure 3: Test accuracy over the course of training of CARATE for the Proteins dataset.

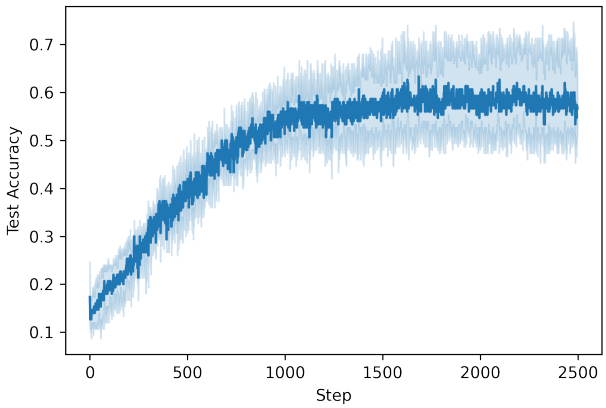

Figure 4: Test accuracy over the course of training of CARATE for the Enzymes dataset.

Table 2: Classification performance of CARATE as compared to other state of the art GNNs and kernel classifiers.

Method	MCF-7	MOLT-4	YEAST	Enzymes	Proteins
CARATE	93.6 ± 0.2	93.5 ± 0.4	89.1 ± 0.5	$(7 \pm 1) \cdot 10^1$	83 ± 2
1-WL	94.5	94.6	89.2	50.8	72.6
WL-OA	-	-	88.2	56.4	73.4
GR	91.7	92.1	88.2	29.5	71.6
SP	91.7	92.1	88.3	39.3	75.6
GIN-e	92	92.4	88.3	38.7	72.2
GIN-e-JK	91.8	92.2	88.2	39.3	72.2

The CARATE regressor converged usually fast and satisfactory accuracies may be achieved with early stopping. The relative errors decreased below experimental accuracy for each class in the ALCHEMY dataset and converged to the p.p.m region for the ZINC dataset.

5 Discussion

The aim of efficiently learning a decent molecular to property representation from a dataset has been achieved. The proposed architecture can be applied to small and bio molecules alike with satisfying results. The CARATE encoder enables multi class classification and regression tasks. Therefore, application of the CARATE technology in multiscale simulations is feasible. Moreover, the multi-headed self attention mechanism allows efficient learning and prediction in the low data regimes. Besides, the CARATE algorithm may detect flawed data in given data sets or uncover statistically unrelated data, thus giving hints on where to improve the quality of the data in a data lake.

On the contrary, in the highly distorted data regime, the CARATE algorithm may be inferior to classical methods based on supported vector machines or random forest classifiers. The behaviour, however, is expected, as GAT units are inserted to flexibly recognize chemical patterns. If the important pattern is not well represented in the dataset, the attention mechanism may forget the pattern due to the dropout probability of 0.6. To overcome this issue, highly distorted data sets can be modelled by adjusting the learning or dropout rate and training for more epochs and achieve similar to or better than baseline results.

Still, the high accuracy for most of the classification and regression tasks suggest that the multi head GAT units are decent encoders for molecular structures. Especially, if presented with a descriptive data regime, CARATE shows low training times while achieving high accuracies for biomolecules and small molecules alike. Therefore, simulations utilizing CARATE seem promising. However, to verify the results, in later studies more experiments have to be conducted. It would be interesting how CARATE performs classification on larger datasets of biomolecules and small molecules because the fitting abilities of the proposed algorithm appear to be very high. Based on the accurate regression results, the GAT encoding enables the subsequently stacked layers to model a given, well described problem accurately if the regression task is manageable in terms of scales. Future work will also focus on applying CARATE to multiscale simulation problems, many class multitask learning, transfer learning[16], and flawed or fraudulent data detection in chemical data sets.

References

[1] L. D. Site, *Entropy* **2014**, *16*, 23–40.

[2] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, K. R. Müller, *Science Advances* **2017**, *3*, 1–6.

[3] H. Xiao, J.-L. Wu, J.-X. Wang, R. Sun, C. Roy, *Journal of Computational Physics* **2016**, *324*, 115–136.

[4] R. Nagai, R. Akashi, O. Sugino, *npj Computational Materials* **2020**, *6*, DOI `10.1038/s41524-020-0310-0`.

[5] L. Böselt, M. Thürlemann, S. Riniker, *Journal of Chemical Theory and Computation* **2021**, DOI `10.1021/acs.jctc.0c01112`.

[6] G. A. Pinheiro, J. Mucelini, M. D. Soares, R. C. Prati, J. L. F. Da Silva, M. G. Quiles, *The Journal of Physical Chemistry A* **2020**, *124*, 9854–9866.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, *Advances in Neural Information Processing Systems* **2017**, *2017-December*, 5999–6009.

[8] A. Rogers, O. Kovaleva, A. Rumshisky, *arXiv* **2020**, DOI `10.1162/tacl_a_00349`.

[9] I. Tenney, P. Xia, B. Chen, A. Wang, A. Poliak, R. T. McCoy, N. Kim, B. V. Durme, S. Bowman, D. Das, E. Pavlick in International Conference on Learning Representations, **2019**.

[10] Y. Yang, Z. Feng, M. Song, X. Wang, *arXiv* **2020**.

[11] S. Lim, Y. O. Lee, *arXiv* **2020**.

[12] A. Capecchi, D. Probst, J. L. Reymond, *Journal of Cheminformatics* **2020**, *12*, 1–15.

[13] H. Altae-Tran, B. Ramsundar, A. S. Pappu, V. Pande, *ACS Central Science* **2017**, *3*, 283–293.

[14] B. Tang, S. T. Kramer, M. Fang, Y. Qiu, Z. Wu, D. Xu, *Journal of Cheminformatics* **2020**, *12*, 1–9.

[15] A. E. Bair, *Molecular Graph Self Attention and Graph Convolution for Drug Discovery*, Massachusetts Institute of Technology, **2019**, pp. 1–57.

[16] P. Velicković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, *arXiv* **2017**, 1–12.

[17] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, M. Neumann in ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020), **2020**.

[18] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, V. Pande, *Chemical Science* **2018**, *9*, 513–530.

[19] GitHub - rusty1s/pytorch_geometric: Geometric Deep Learning Extension Library for PyTorch, `https://github.com/rusty1s/pytorch_geometric`, (Accessed on 05/23/2021).

[20] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala in *Advances in Neural Information Processing Systems 32*, (Eds.: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett), Curran Associates, Inc., **2019**, pp. 8024–8035.

[21] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T. E. Oliphant, *Nature* **2020**, *585*, 357–362.

[22] G. Chen, P. Chen, C.-Y. Hsieh, C.-K. Lee, B. Liao, R. Liao, W. Liu, J. Qiu, Q. Sun, J. Tang, R. Zemel, S. Zhang, *arXiv* **2019**.

[23] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, M. Neumann, *arXiv* **2020**.

[24] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, S. Jegelka, *arXiv* **2018**, 1–14.

[25] N. Shervashidze, P. Schweitzer, E. J. Van Leeuwen, K. Mehlhorn, K. M. Borgwardt, *Journal of Machine Learning Research* **2011**, *12*, 2539–2561.