Journal Name

ARTICLE TYPE

Cite this: DOI: 00.0000/xxxxxxxxxx

Introducing CARATE: Finally speaking chemistry†

Julian M. Kleber,*[a]

Received Date
Accepted Date

DOI: 00.0000/xxxxxxxxxx

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 in the context of medicinal chemistry 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.

Numerous methods try to tackle multiple scales 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. Current representations of chemicals lack the descriptiveness necessary for modern modelling questions.

This work introduces a novel representation of small and large molecules. The representation is obtained by the *biochemical and pharmaceutical encoder* (CARATE). Furthermore, the regression and classification abilities of the learned representation by CARATE are evaluated against benchmarking datasets (ZINC, ALCHEMY, MCF-7, MOLT-4, YEAST, Enzymes, Proteins) and compared to other baseline approaches. CARATE outperforms on classification tasks of large biomolecules as well as on quantum chemical regression tasks of small molecules. .

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].

The problem of multiple scales shows us that our theories do not handle the concept of multiple scales well enough for our computational abilities, such that each multiscale modelling is doomed using our technology alongside our current state of theoretical physics describing the dynamics of the system.

For quantum chemistry in any scale, we want to solve the time-dependent Schrödinger equation when considered with the dynamic behavior of the system.

$$H\psi(t,r) = E\psi(r,t) \tag{1}$$

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.

Yet, the current representation of chemicals (SMILES, InChI and Lewis-like structures) can't be suitable for modelling because they are in themselves abstractions.

Even worse, the original SMILES algorithm was claimed to be unique[2], but was proven to be non-unique later.[3] It is impossible for fingerprints and SMILES strings to represent a wave function properly and thus methods using these representations are doomed to fail.

The recent surge in application of SMILES for use in DL[4–9] leaves doubts on the applicability of the derived models. On the contrary, large matrix representations are mostly not accurate while being resource intense.[10]

To accurately approximate the physical (simulated) properties of a given compound, a more accurate, uniform representation is mandatory.

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

[a] *Berlin, Germany.; E-mail: julian.m.kleber@gmail.com*

† Electronic Supplementary Information (ESI) available: [https://codeberg.org/sail.black/carate].

molecule with a receptor, one needs to combine quantum chemistry (QC) with classical molecular mechanics (MM) at the moment.

The computations performed by the QM/MM approximation still have the QC calculations as a bottleneck. However, recent advances in DL make the prediction of a particular simulation scale accessible[11,12]. Recent prediction of quantum chemical properties with deep learning methods became accurate[13] leading to MD simulations using learned coupled-cluster potentials[14].

However, current machine learning methods, including modern deep learning methods, mostly do only solve the time-independenent Schrödinger equation.

$$H\psi(r) = E\psi(r) \tag{2}$$

Also, this work is considered with solutions of the time-independent Schrödinger equation.

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[15].

Yet, graph-based methods do not reach the accuracy necessary for accurate simulation because all methods neglect self-interaction, a foundational quantum mechanical principle. Either the graph-based algorithm ignores self-interaction, or the given input neglects self-interaction.[9] The current graph-based methods thus lead to an already flawed ground truth and are doomed to fail.

Up to now, however, to the best of my knowledge, no algorithm with short training times and competitive prediction abilities has been reported. For example, *Chen et al.* report a similar algorithm to CARATE developed during the same time, but omit results on large datasets, which they used to pretrain their algorithm[9]. Thus, a novel auto-encoder for obtaining a wave function to the eigenvalue equations **??** might offer a route to an ad-hoc adaptive model for multiscale simulations.

Presently, it is barely possible to perform simulations of decent size and complexity on edge devices or personal computers. Typically, the equipment involved consists of a couple of nodes on a high-performance cluster. Only insiders can access high-performance clusters, giving rise to a more and more arcane elite.

A novel method equipped with outstanding algorithms may lead to a new era of edge simulations to enable scientists and accelerate as well as democratize progress in the natural and life sciences.

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 depends upon the molecular representation.[16].

The recent advances in natural language processing (NLP) introduced by the self attention mechanism in bidirectional auto encoders[17] showed that respective algorithms (BERT technology) can learn semantic properties of the input.[18]. Especially, the BERT algorithm has abilities to organize and structure information.[19].

Because molecular descriptors can be learned in a data-driven way[20] 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[21].

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[22].

However, one-shot-learning based on self-attention LSTM with encoded molecular graphs as inputs showed more decent results[23]. 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)[21].

In 2017, the group around *Schütt et al.* proposed *SchNet* and their benchmark dataset *ISO17* . The key improvement in *SchNet* was to get hold of the molecular geometry as a graph while using convolutions to craft weighted feature importance in the molecule.

Another way to achieve better structure recognition is to apply message passing networks[25]. Moreover, multi-headed self attention has been applied to GNN encoded molecular structures through a self attention mechanism[26].

This work completes the series on the application of attention-based algorithms to chemical problems. This work investigates a multi-headed graph self-attention model (GAT)[27] with graph-convolutions as an encoder for chemical classification and regression tasks, with significant results.

Chen et al. introduced the idea of bidirectional transformers to chemistry on graphs; however, they still work on SMILES and did also not publish their results on large benchmark datasets[9]. The idea of obtaining a better representation through graph-deep learning is becoming more and more popular, is however still working on SMILES representations[28].

3 Methods

The whole code is open source and available on Codeberg a German association for the advancement of open-source software (`https://codeberg.org/sail.black/carate`). The package can be obtained via PyPi and pip, too (`https://pypi.org/project/carate/`).

For the reader with historical interest, the notebooks containing the first experiments are also published under my private profile (`https://codeberg.org/cap_jmk/carate-notebooks`)

In generatl, though the training on GPU is slightly faster, the CPU was chosen as the reference chip.

The accuracy of the model is sometimes slightly lower and less reproducible across different CUDA versions and GPUs at different times.

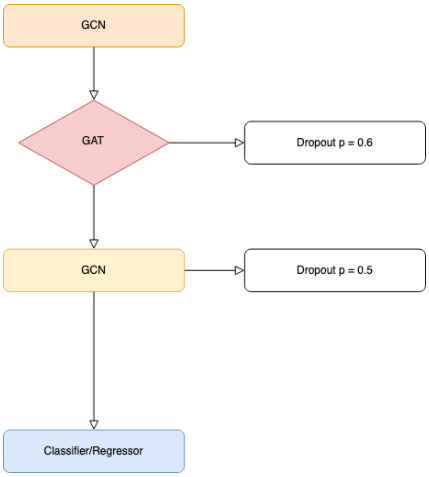

Fig. 1 General architecture of the CARATE network used for classification and regression tasks.

Both methods, regression and classification, 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.

The datasets were obtained from TUDataset[29], and MoleculeNet[30] through the PyTorchGeometric API[31]. 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[31].

All GAT and GNN implementations were applied from or achieved with the PytorchGeometric[31] library. Other deep learning building blocks were implemented using the PyTorch[32] and Numpy[33] 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 prediction, the encoder-predictor model was tested against physical regression tasks on the ALCHEMY[34] and ZINC data sets obtained from TUDatasets[29].

For each task, no more than 50 training iterations are necessary for a good convergence. The convergence is thus quite fast.

In fact, early stopping was beneficial in most cases, such that longer training results may lead to less accurate results. The degradation of the convergence behavior is indicating some overfitting behavior.

The desired targets were normalized to ensure smooth convergence behavior during training.

The architecture for the regression tasks differs from the classification to obtain multitask regression results, instead of multitask classification only by the output function applied to the network. The results are compared to GIN-ε.[35,36]The GINE-ε network uses batch normalization as well as zero mean and unit variance normalization.

Table 1 Regression performance compared to baseline GINE-ε[35].

Method	ZINC	ALCHEMY
CARATE	0.0036	0.00108
GINE-ε[35]	0.084	0.103

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 minimal and may not be descriptive for the classification problem they are intended for.

4 Results

The outperformance of other state-of-the-art models indicates 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[37]. The MOLT-4 dataset shows similar behavior 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.

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 large biomolecules alike with satisfying results. The CARATE encoder enables multi-class classification and regression tasks.

Moreover, the multi-headed self-attention mechanism allows efficient learning and prediction in low data regimes. Besides, the CARATE algorithm may detect flawed data in given data sets

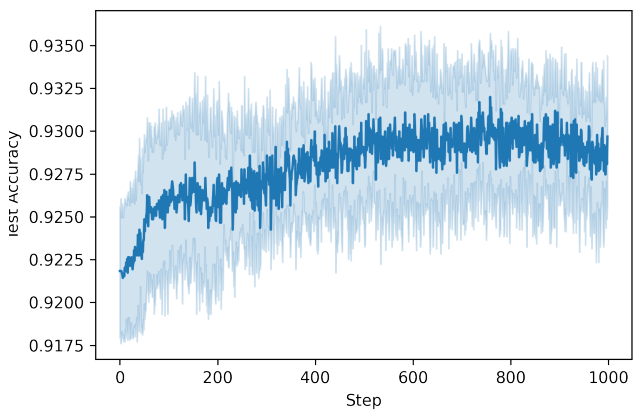

Fig. 2 Test accuracy over the course of training of CARATE for the MOLT-4 dataset.

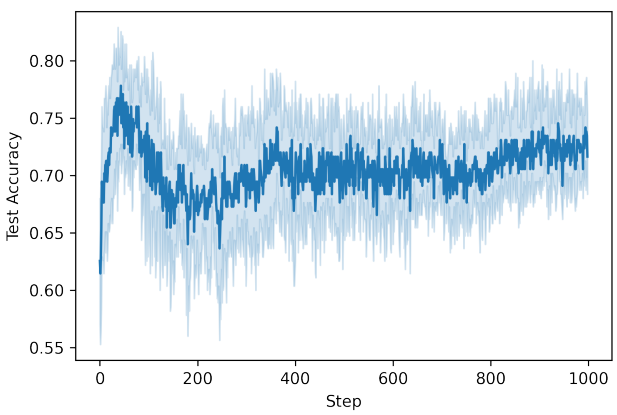

Fig. 3 Test accuracy over the course of training of CARATE for the Proteins dataset.

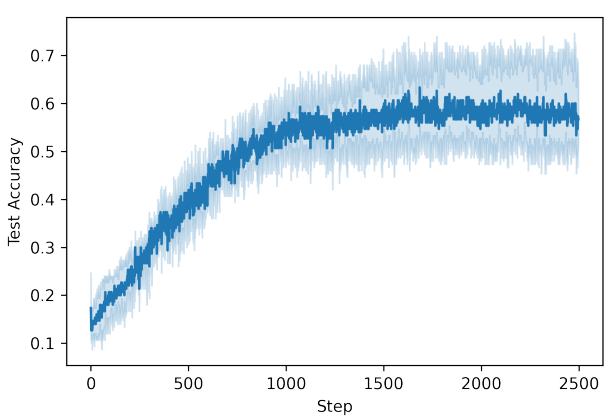

Fig. 4 Test accuracy over the course of training of CARATE for the Enzymes dataset.

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 behavior, however, is expected, as GAT units are inserted flexibly to detect chemical patterns.

If the important pattern is not well represented in the data set, 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.

In fact, the performance of the algorithm might as well be increased by modifying the architecture in subsequent work via adjusting the network stack with batch-normalization or a slightly different dropout-rate or dropout localization.

Both regressions are multitasking, and the MAE may not be a good estimate of the performance on quantum chemical calculations. On the other hand, the MAE is used in the general Hartree-Fock scheme, making it a physically sound metric.

6 Conclusions

Therefore, the accurate results on quantum chemical datasets indicate that the algorithm is learning a wave function ψ to solve the Schrödinger equation via the last regression layer such that the general Eigenvalue equation

$$\hat{A}\psi = \lambda\psi \tag{3}$$

can be applied to understand the calculation. When considered with multitask problems, the network solves

$$\hat{A}_i\psi = \lambda_i\psi \tag{4}$$

The high accuracy for most of the classification and regression tasks suggests that the multi head GAT units are decent encoders for molecular structures.

When considering the initial motivation of solving the eigenvalue problem of an operated acting on a wave function, we must conclude that the encoding of the molecule through the CARATE unit represents indeed a wave function and the operator acting on the wave function is represented by the final classification or regression layer of the network.

The principle must by definition be self-replicating such that one can add layer after layer and would still satisfy the eigenvalue equations.

Especially, if presented with a descriptive data regime, CARATE shows low training times while achieving high accuracies for biomolecules and small molecules alike.

The idea of CARATE is similar to the SchNet[38] but incorporating a self-interaction term such that high accuracies can be achieved. By incorporating more physical knowledge into the model, the encoding was framed as a Markov graph in the wider sense.

Even though there is a high dropout probability and one dropout layer is stacked as the last layer, accuracies are high. The

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

high accuracies contradict common deep learning knowledge.

Simulations utilizing CARATE seem promising. However, to verify the results, in later studies more experiments are necessary. 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 on large datasets such as ZINC and ALCHEMY.

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[27], and flawed or fraudulent data detection in chemical data sets.

Conflicts of interest

There are no conflicts to declare. Though for researching the patent system, a patent was filed in 2021 in the US. The patent was removed in 2023 by their government and would have required refiling. The author refused to refile the patent, as he did not believe in patents anyway.

Acknowledgements

Special thanks go to my family and friends, the Berlin AI/ML community - especially to all members of BLISS e.V - and my mentors at the different universities, companies, and research institutes in Berlin.

Without you, I would not be where I am today. Also, the paper would not have advanced to this stage.

The work started as a small project. Still, the potential to improve the current way of doing quantum chemistry presented itself early during the first experiments.

The more time passed, the more work was done to finally prove more related theorems through different approaches. I had more time to study other perspectives on the relationship of deep learning and quantum physics.

This work needed a lot of reflection and questioning, even though the results were clear right from the beginning.

During this time, the software advanced to a reproducible state and the paper is more complete than the first preprint I wrote back 2021.

It was not easy to cut off the work to not change the paper from its original state and leave open-ended questions for later research. Also, it was intended to avoid distortion of the historical context of the research in hindsight.

I tried to freeze the historical context by publishing as much of my research even though it is mostly in preprint stage or using other formats than classical research papers. You can refer to my Research Gate profile to obtain the documents.

The distortion might happen easily and especially during mid-2022 through 2023, a lot of research in the wider academic circle of mine was influenced and took influence through dialogues and discussions.

I want to make the point clear, that the whole theoretical and scientific implications of solving the eigenvalue problem related to wave functions via DL were not understood at the point of the first experiments of this work.

I hope could provide some insights about my reasoning behind the comparably simple algorithm to tackle this gap.

In this sense, I also want to thank especially the RSC who encouraged me to finish this work on my own.

I want to clarify that I do not prefer any one chip company over another, and would have preferred using my own open-source chips if I had any suitable ones.

Also, this version is not yet complete and serves to me as a reminder where to stop the theoretical part to not distort the historical events.

There will be a final version at the end with the new data and some extra references on the relation of the Schrödinger equation and quantum mechanics.

Notes and references

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

2 D. Weininger, A. Weininger and J. L. Weininger, *Journal of Chemical Information and Computer Sciences*, 1989, **29**, 97–101.

3 G. Neglur, R. Grossman and B. Liu, author, 2005, pp. 145–157.

4 C. Li, J. Feng, S. Liu and J. Yao, *Computational Intelligence and Neuroscience*, 2022, **2022**, year.

5 X. Li, X. Yan, Q. Gu, H. Zhou, D. Wu and J. Xu, *Journal of Chemical Information and Modeling*, 2019, **59**, 1044–1049.

6 B. Ramsundar, B. Liu, Z. Wu, A. Verras, M. Tudor, R. P. Sheridan and V. Pande, *Journal of Chemical Information and Modeling*, 2017, **57**, 2068–2076.

7 W. Ahmad, E. Simon, S. Chithrananda, G. Grand and B. Ramsundar, 2022.

8 E. N. Feinberg, D. Sur, Z. Wu, B. E. Husic, H. Mai, Y. Li, S. Sun, J. Yang, B. Ramsundar and V. S. Pande, *ACS Central Science*, 2018, **4**, 1520–1530.

9 D. Chen, K. Gao, D. D. Nguyen, X. Chen, Y. Jiang, G. W. Wei and F. Pan, *Nature Communications*, 2021, **12**, 1–9.

10 T. Engel, *Representation of Chemical Compounds*, 2003, `https://onlinelibrary.wiley.com/doi/abs/10.1002/3527601643.ch2`.

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

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

13 R. Nagai, R. Akashi and O. Sugino, *npj Computational Materials*, 2020, **6**, year.

14 J. S. Smith, B. T. Nebgen, R. Zubatyuk, N. Lubbers, C. Devereux, K. Barros, S. Tretiak, O. Isayev and A. E. Roitberg, *Nature Communications*, 2019, **10**, 1–8.

15 L. Böselt, M. Thürlemann and S. Riniker, *Journal of Chemical Theory and Computation*, 2021.

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

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

18 A. Rogers, O. Kovaleva and A. Rumshisky, *arXiv*, 2020.

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

20 Y. Yang, Z. Feng, M. Song and X. Wang, *arXiv*, 2020.

21 S. Lim and Y. O. Lee, *arXiv*, 2020.

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

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

24 2020, 1–12.

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

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

27 P. Velicković, G. Cucurull, A. Casanova, A. Romero, P. Liò and Y. Bengio, *arXiv*, 2017, 1–12.

28 X. Zang, X. Zhao and B. Tang, *Communications Chemistry*, 2023, **6**, year.

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

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

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

32 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 and S. Chintala, *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035.

33 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 and T. E. Oliphant, *Nature*, 2020, **585**, 357–362.

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

35 C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel and M. Neumann, *arXiv*, 2020.

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

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

38 K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko and K.-R. Müller, *The Journal of Chemical Physics*, 2018, **148**, year.