JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

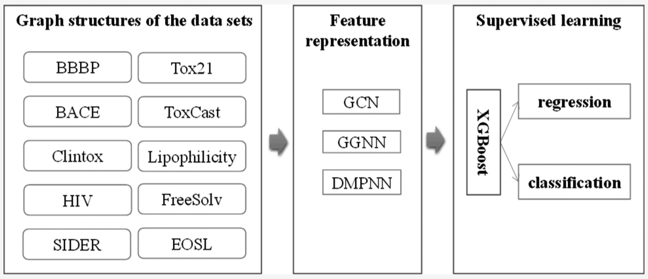
pubs.acs.org/jcim Article

XGraphBoost: Extracting Graph Neural Network-Based Features for a Better Prediction of Molecular Properties

Daiguo Deng, Xiaowei Chen, Ruochi Zhang, Zengrong Lei, Xiaojian Wang,* and Fengfeng Zhou*

Cite This: J. Chem. Inf. Model. 2021, 61, 2697−2705 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information



ABSTRACT: Determining the properties of chemical molecules is essential for screening candidates similar to a specific drug. These candidate molecules are further evaluated for their target binding affinities, side effects, target missing probabilities, etc. Conventional machine learning algorithms demonstrated satisfying prediction accuracies of molecular properties. A molecule cannot be directly loaded into a machine learning model, and a set of engineered features needs to be designed and calculated from a molecule. Such hand-crafted features rely heavily on the experiences of the investigating researchers. The concept of graph neural networks (GNNs) was recently introduced to describe the chemical molecules. The features may be automatically and objectively extracted from the molecules through various types of GNNs, e.g., GCN (graph convolution network), GGNN (gated graph neural network), DMPNN (directed message passing neural network), etc. However, the training of a stable GNN model requires a huge number of training samples and a large amount of computing power, compared with the conventional machine learning strategies. This study proposed the integrated framework XGraphBoost to extract the features using a GNN and build an accurate prediction model of molecular properties using the classifier XGBoost. The proposed framework XGraphBoost fully inherits the merits of the GNN-based automatic molecular feature extraction and XGBoost-based accurate prediction performance. Both classification and regression problems were evaluated using the framework XGraphBoost. The experimental results strongly suggest that XGraphBoost may facilitate the efficient and accurate predictions of various molecular properties. The source code is freely available to academic users at https://github.com/chenxiaowei-vincent/XGraphBoost.git.

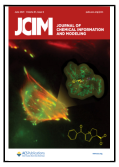
■ INTRODUCTION

Determining the various properties of a molecule is a critical step in drug discovery. A series of complicated biochemical reactions are usually carried out to get the properties of a given molecule.[1] The quantities of the already accumulated chemical molecules and the rapidly emerging novel molecules render it an impossible mission to experimentally determine a specific property of all the molecules.[2,3] The computer-aided drug design technique has been introduced to computationally predict the molecular properties, which has become one of the major bioinformatics research trends.[4,5]

The conventional strategies to extract the molecular descriptors (or features) are usually calculated based on the molecules' three-dimensional structures, and the molecular properties may be predicted by the quantitative structure−activity relationship (QSAR) models.[6,7] Machine learning algorithms have been utilized to achieve very good prediction performance of chemical molecular properties.[8−13] Various popular supervised classifiers are used to predict chemical molecular properties, including support vector machine (SVM),[14] random forest (RF),[15] artificial neural network (ANN),[16] etc.

Deep learning algorithms have recently become very successful in various biochemical prediction areas. Deep learning algorithms may replace the conventional machine learning algorithms and perform the supervised learning tasks. A deep neural network (DNN) was used to build a molecular property prediction model using the precalculated molecular descriptors (or features) and demonstrated performance improvements compared with the machine learning algorithms.[17,18] Deep learning algorithms may also serve as the feature extractors for the biomarker development and drug discovery problems.[19] Three-dimensional convolutional neural networks (3D-CNNs)[20] and graph convolution neural networks (GCNs)[21,22] have been successfully utilized in extracting molecular descriptors (or features), but the challenging requirements for a large number of training samples and intensive computing powers remain to be resolved for the deep learning-based molecular descriptor extractions and property prediction.

Received: December 26, 2020
Published: May 19, 2021



https://doi.org/10.1021/acs.jcim.0c01489
J. Chem. Inf. Model. 2021, 61, 2697−2705

This study proposed the framework XGraphBoost to integrate the GCN-based feature extractions and the conventional classifier XGBoost for the molecular property prediction problem. An extensive evaluation procedure was carried out on the 12 data sets from Wu et al. and Mayr et al.[23] The experimental results evaluated the proposed framework XGraphBoost using both classification and regression problems. This proof-of-principal study strongly suggested the necessity of optimizing both molecular feature extraction and the classifications for the molecular property prediction problems.

Drug development is a highly costly and time-consuming procedure, and it is essential to reduce the computing requirement and error rate of the computational candidate drug screening. Both machine learning[24] and deep learning[25,26] have been successfully utilized in the drug development. This facilitates the rapid innovations of the virtual screening (VS) of candidate drugs.[27,28] Various computational algorithms have been successfully deployed, including Fingerprinting,[29,30] Scoring Function,[31,32] Docking Method,[33] SMILES,[34] Molecule Generation,[35] Property Prediction,[36,37] etc.

The molecular property prediction is one of the most important problems in the drug development procedure, and quite a few conventional machine learning algorithms have been utilized in predicting molecular properties. Researchers calculated the Morgan fingerprints of the given molecules[38] and then used the classifiers Support Vector Machine (SVM)[39] or random forest (RF)[40] to train the prediction models.

The graph neural networks (GNNs) represent one of the major breakthroughs in learning the interatom connections. Various GNN subtypes demonstrate their efficacies of capturing the internode relationships via message passing between graph nodes, including graph convolutional network (GCN),[41−43] gated graph neural network (GGNN), and direct MPNN (DMPNN).[36]

■ MATERIALS AND METHODS

Data Set Summary. The proposed framework XGraphBoost was comprehensively evaluated using the 10 molecular property data sets from Wu et al.[44] and Mayr et al.[45] The details of the 10 data sets were summarized in Table 1. The number of molecules in these data sets ranges between 600 and 42,000. The investigated molecular properties include quantum mechanics, physical chemistry, biophysics, and physiology.

Table 1. Summary of the Data Set Details[a]

data set	task type	metric	compounds
ESOL	regression	RMSE	1128
FreeSolv	regression	RMSE	642
Lipophilicity	regression	RMSE	4200
HIV	classification	AUC	41127
BACE	classification	AUC	1513
BBBP	classification	AUC	2039
Tox21	classification	AUC	7831
ToxCast	classification	AUC	8575
SIDER	classification	AUC	1427
Clintox	classification	AUC	1478

[a] Each data set is denoted by an abbreviation in the column "data set". The column "task type" denotes whether this data set is a regression or classification problem. The last two columns are the performance metric ("metric") and the number of compounds ("compounds"), respectively.

The molecules of all 10 data sets are encoded in the SMILES strings.[46] Both classification and regression problems are investigated, as shown in Table 1.

Each data set is randomly split by the stratified strategy into the train, validation, and test data sets by the proportions of 0.8, 0.1, and 0.1, respectively. So, the sample distributions in the train, validation, and test data sets were the same for one original data set. All the experiments were repeated three times using different random seeds to minimize the effect of sample distributions on the model performances. The evaluated models are trained on the train data set, tuned for the parameter choices with better model performances on the validation data set, and tested for the final model performances on the test data set. Different task types are evaluated by different performance metrics, as shown in Table 1.

Graph Neural Networks. Graph neural networks (GNNs) have been used to learn the representation of the molecular structures. Each graph is composed of nodes and edges. A node is described by the atomic type, atomic element, the number of additional hydrogen atoms, the number of valence, aromatic properties, and the other properties. These descriptors of each node are encoded by the one-hot strategy. The adjacency matrix represents the connectivity between the atomic pairs regardless of the single or double bonds. This study used the tool RDKit[47] to process these SMILES encoding compounds to get the molecular graphs and Morgan fingerprints,[38] which will be used in the GNNs and XGBoost.

The states of the graph nodes are updated using the node embedding method, which is described as

$$H_i^t = U(H_i^{t-1}, m_i^t)$$

This formula describes the ith node updated by the previous node state h_i^{t-1} and a message state of the interaction term m_i^t with its neighboring node. The graph convolutional neural network (GCN) is the simplest version of a messaging neural network that uses convolutional operations.

The gated graph neural network (GGNN) utilizes the gate recurrent unites (GRUs) in the propagation step.[48] While the message passing neural network (MPNN) unifies and generalizes various existing GNNs,[49] the directed version of MPNN (DMPNN) propagates information via the directed bonds.[50] DMPNN demonstrates superior prediction performances of molecular properties.

XGBoost for Classification and Regression. The extreme gradient boosting algorithm (XGBoost) is an efficient implementation of the gradient boosting strategy.[51,52] The gradient boosting decision tree (GBDT) is an ensemble supervised learning algorithm that summarizes the results of multiple weak learners like decision trees. Chen et al. proposed this very fast implementation XGBoost that greatly facilitated the applications of the gradient boosting strategy.[53] XGBoost improves the conventional gradient boosting strategy via an advanced regularization as in the following formula

$$\tilde{y}_l = \varnothing(x_i) = \sum_{k=1}^{K} f_k(x_i), f_k \in F$$

The variable F is a set of all the base decision trees, and the function f_k at each of the k steps converts the descriptor values of x_i to the output. The function f_k is what needs to be learned during the training procedure. XGBoost proposed a minor modification in the regularization objective function.

$$L(\varnothing) = \sum_i l(\tilde{y}_l, y_i) + \sum_k \Omega(f_k)$$

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \| \omega \|^2$$

XGBoost for Classification and Regression. This study was carried out in three major steps, as shown in Figure 1.

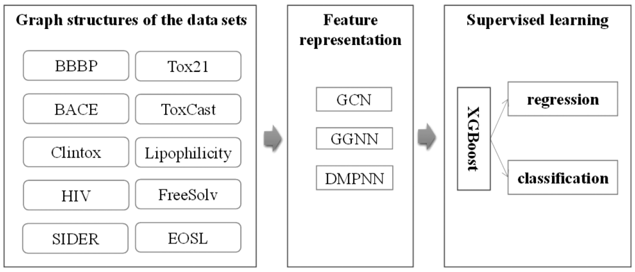


Figure 1. The overall workflow of building an efficient and accurate predictor for molecular properties. The descriptors (or features) of the molecules are extracted by a GNN model and are then loaded to the supervised learner XGBoost for classification or regression.

First, the raw molecular data was formatted as a graph structure using a similar method as in ref 36. Second, the proposed framework XGraphBoost learned the graph representations of the molecular features using the networks GCN, GGNN, and DMPNN. Third, the graph representations were loaded as the sample features to the supervised learner XGBoost.

The proposed framework XGraphBoost orchestrates the merits of both the graph representation of molecules and the data-driven supervised learner XGBoost. The three graph neural networks have been widely used in the molecular property prediction studies and demonstrated their capabilities in representing the chemical and physical features of both the atoms and the interatom bonds, while the supervised learner XGBoost is a popular and very efficient machine learning algorithm for various biochemical pattern prediction problems.

Performance Metrics. This study evaluates the proposed framework XGraphBoost with both classification and regression problems. Their respective performance metrics are defined as follows.

A binary classification problem may be evaluated using the metrics specificity (Sp), sensitivity (Sn), accuracy (Acc), and area under receiver operating characteristics curve (AUC). A binary classification problem has two classes of samples, i.e., positive and negative samples. The positive samples are separated as true positives and false negatives if they are correctly and incorrectly predicted by a binary classifier, respectively. The numbers of true positives and false negatives are defined as TN and FP. The negative samples are separated as true negative and false positive samples, and the numbers of these two subgroups are defined as TN and FP. The correct prediction rates of the positive and negative samples are defined as sensitivity (Sn = TP/(TP+FN)) and specificity (Sp = TN/(TN+FP)), respectively, and the overall accuracy is defined as Acc = (TP+TN)/(TP+FN+TN+FP).

The receiver operating characteristics (ROC) curve is defined as the connected curve of the points (true positive rate, false positive rate). These two rates are defined as TPR = TP/(TP+FN) and FPR = FP/(FP+FN). The area under the ROC curve is defined as the classification performance metric AUC. AUC ranges between 0 and 1, and a binary classifier usually achieves AUC between 0.5 and 1.0. A larger value of this parameter-independent metric AUC suggests a better classification model.

A regression model predicts a continuous value as the molecular property, and it needs a different performance metric. The root-mean-square error (RMSE) is a popularly used metric to evaluate a regression model.[54,55] RMSE is the standard deviation of the residuals (or prediction errors) and measures how spread out the residuals are from the best fit.

Implementation and Running Environments. All the experiments were implemented using the programming language Python version 3.7.5, using the packages pytorch version 1.3.1 and py-xgboost-gpu version 0.9.0. The experiments were conducted using a fat computing node with 1 CPU nodes (4 cores per CPU), 1 GPU cards (Nvidia 2018Ti) (10GB memory per card), and 16 GB system memory.

■ RESULTS AND DISCUSSION

Implementation and Running Environments. All three graph neural networks (GNNs) demonstrated satisfying prediction performances on both classification and regression problems, as shown in Figure 2 and Supplementary Figure 1. The network DMPNN achieved the best loss rates on both the training samples and validation samples. DMPNN also demonstrated the fastest converging speed. These three GNNs converged on the loss rates within 200 epochs on both the training and validation samples. GCN had the

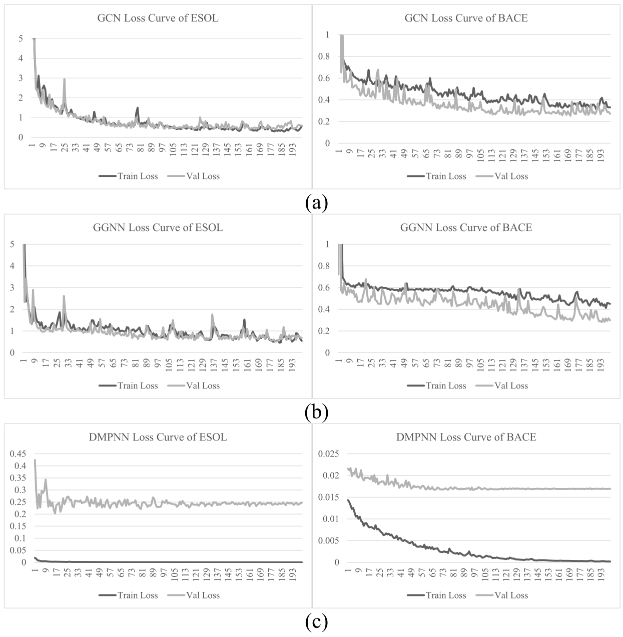


Figure 2. The loss curves of the three GNNs on the classification and regression problems. The representative classification data set BACE and regression data set ESOL were evaluated for the networks (a) GCN, (b) GGNN, and (c) DMPNN. The horizontal axis gave the epochs, and each network was evaluated for its loss curves on the training (train_loss) and validation (val_loss) samples. These three networks were evaluated on all 10 data sets, and the visualization plots are shown in Supplementary Figure S1.

https://doi.org/10.1021/acs.jcim.0c01489

Table 2. Performance Comparison of the Three GNNs[a]

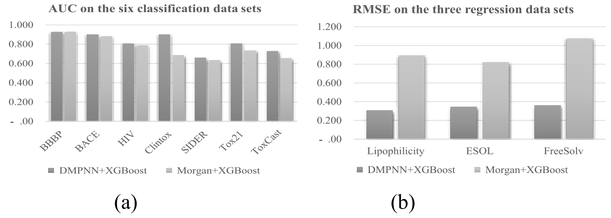
data set	metric	DMPNN	GGNN	GCN
ESOL	RMSE	0.329 ± 0.029	1.026 ± 0.198	1.470 ± 0.285
FreeSolv	RMSE	0.287 ± 0.064	1.725 ± 0.30	3.499 ± 0.111
Lipophilicity	RMSE	0.453 ± 0.011	1.005 ± 0.060	1.918 ± 0.653
BACE	AUC	0.866 ± 0.030	0.880 ± 0.010	0.774 ± 0.011
BBBP	AUC	0.932 ± 0.012	0.893 ± 0.015	0.857 ± 0.014
Clintox	AUC	0.898 ± 0.060	0.756 ± 0.043	0.627 ± 0.018
HIV	AUC	0.801 ± 0.016	0.629 ± 0.006	0.682 ± 0.035
Tox21	AUC	0.819 ± 0.005	0.699 ± 0.040	0.818 ± 0.003
ToxCast	AUC	0.797 ± 0.028	0.725 ± 0.021	0.453 ± 0.019
SIDER	AUC	0.655 ± 0.033	0.611 ± 0.009	0.503 ± 0.005

[a] The three networks DMPNN, GGNN, and GCN were evaluated on the 10 data sets. The supervised learning algorithm was XGBoost. The regression models were evaluated using the metric RMSE, and the classification models were evaluated by AUC. Ten random runs were performed, and each model had the averaged value and standard deviations of the performance metric.

relatively smallest difference between the loss rates of training and validation samples. So, the GCN-trained models tended to have the smallest possibility of overfitting. GCN also dropped the slowest in the loss curves and demonstrated large fluctuations. Similar observations may be found for the three GNNs on all 10 data sets in Supplementary Figure S1.

The three GNNs were evaluated on the 10 data sets for their performances using XGBoost as the supervised learning models, as shown in Table 2. DMPNN outperformed the other two GNNs on 9 out of the 10 data sets, suggesting that DMPNN extracted features with more significant associations with the molecular properties. DMPNN achieved an averaged AUC = 0.866, which was slightly smaller than that (averaged AUC = 0.880) of the GGNN model. GCN performed the worst on 8 out of the 10 data sets, suggesting that the GCN-extracted features alone may not deliver a satisfying performance for the molecular property prediction problem, but GCN achieved the second-best performance (averaged AUC = 0.818), which was very close to the best model based on the DMPNN-extracted features (averaged AUC = 0.819). So, it is essential to evaluate how each of the three GNN-based feature extraction algorithms performs on a specific prediction problem (or data set) before further deployment of the prediction models.

Comparison with the Popular Morgan Fingerprint Features. This study further compared the duet of DMPNN features and the supervised learner XGBoost with that of the popular Morgan fingerprint features and XGBoost on all the data sets, as shown in Figure 3. The DMPNN-extracted features achieved the best performances on 9 out of the 10 data sets in the above section. So, the rest of this study used DMPNN as the default feature extraction algorithm.



(a)

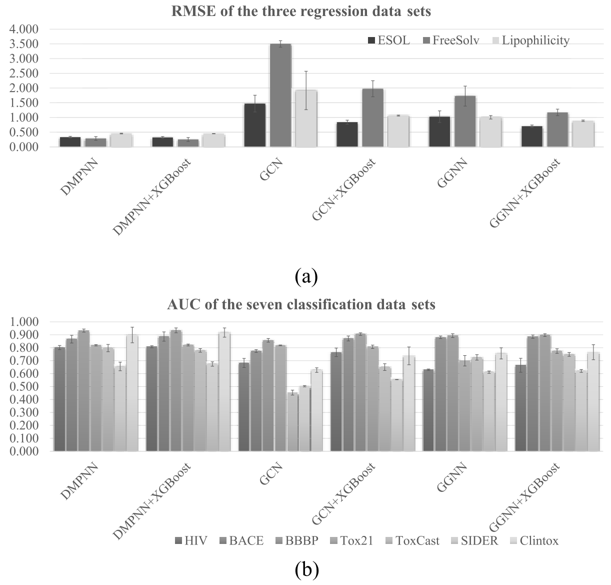
(b)

Figure 3. Performance comparison of the XGBoost models using DMPNN features and Morgan fingerprints. The horizontal axis gave the list of data sets, and the vertical axis gave (a) the detailed AUC values for the six classification data sets and (b) the RMSE values for the three regression data sets.

Figure 3 demonstrated that the DMPNN-extracted features outperformed the Morgan fingerprints on all 10 data sets. A good supervised learning model tends to have a large AUC for the classification problem and a small RMSE value for the regression problem, and the largest improvement in the classification AUC value (0.215) was achieved on the Clintox data set. The overall AUC of the Clintox data set was improved from 0.685 using the Morgan fingerprints to 0.899 using the DMPNN-extracted features, as shown in Figure 3(a). The DMPNN-extracted features achieved smaller RMSE values than the Morgan fingerprints, and the ratios between them are 0.344, 0.421, and 0.335 for the three data sets Lipophilicity, ESOL, and FreeSolv, respectively.

So, it is necessary to use the GNNs like DMPNN to extract the molecular features for a better supervised learning model.

Evaluating the Necessity of Using the Supervised Learner XGBoost. Figure 4(a) showed that all three



(a)

(b)

Figure 4. Performance comparison between the supervised learner XGBoost and the original output layer of the GNN. The horizontal axis gave the model architectures, and the vertical axis gave the values of (a) RMSE for the regression data sets and (b) AUC for the classification data sets. The three GNNs with the original output layers were denoted as DMPNN, GCN, and GGNN, and the three models with XGBoost as the output layers were denoted as DMPNN +XGBoost, GCN+XGBoost, and GGNN+XGBoost.

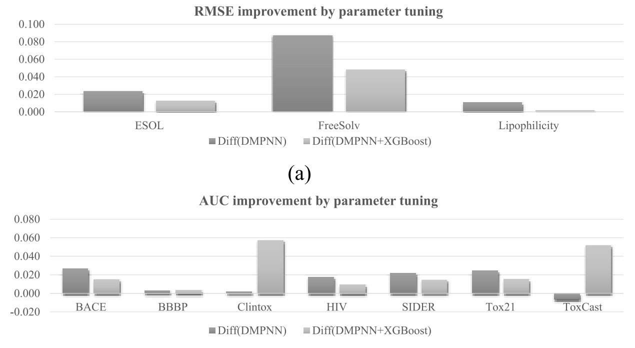
https://doi.org/10.1021/acs.jcim.0c01489

regression data sets received improved RMSE values if the output layers of the three GNNs were replaced by the supervised learner XGBoost. The largest improvement was achieved for the GCN model on the FreeSolv data set, the RMSE value was reduced from RMSE = 3.499 (the GCN model) to 1.975 (the GCN+XGBoost model), and the standard deviations of these two models were 0.111 and 0.274, respectively.

Almost all the classification models were improved by simply replacing the output layers of these GNNs with the supervised learner XGBoost, as shown in Figure 4(b). XGBoost achieved worse AUC values on only two cases, i.e., the DMPNN +XGBoost model on the ToxCast data set and the GCN +XGBoost model on the Tox21 data set. XGBoost reduced the AUC values of the DMPNN algorithm from AUC = 0.797 (the DMPNN model) to 0.779 (the DMPNN+XGBoost model) on the ToxCast data set. The classifier XGBoost caused the minor AUC decrease of 0.011 on another data set Tox21. All the other GNN models were improved by replacing their output layers with XGBoost, and the largest improvement of 0.197 in AUC was achieved for the GCN-extracted features on the ToxCast data set.

Summarized from the data in this section, the duet of the DMPNN-extracted features and the supervised learner XGBoost achieved the best performances for 9 out of 10 data sets. The DMPNN-XGBoost performed (AUC = 0.779) slightly worse than the DMPNN model (AUC = 0.797) on the ToxCast data set.

Tuning the Parameters for a Better Performance. The three XGBoost parameters were tuned for the best value choices, i.e., the learning rate, max depth, and minimum child weight, as shown in Figure 5. Five values of the learning rate [0.01, 0.05, 0.1, 0.15, 0.2] were evaluated. Five values[29,8,10] were also evaluated for the other two parameters: maximum tree depth and the minimum child weight.



(a)

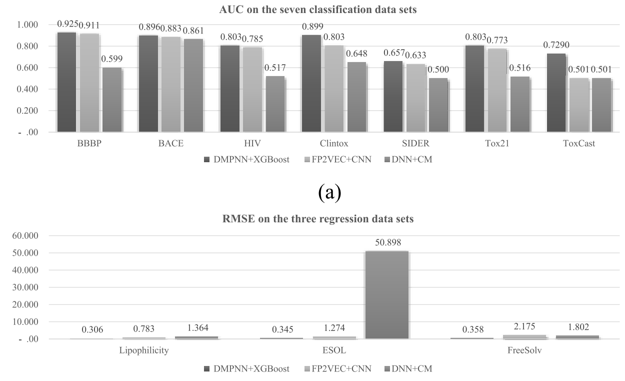
(b)

Figure 5. The performance improvements in AUC or RMSE by tuning the parameters of the supervised learner XGBoost. The horizontal axis listed the data sets, and the vertical axis gave the difference before and after the parameter tuning. (a) A good regression model tended to have a small RMSE value. So, the RMSE values of the DMPNN models and the DMPNN+XGBoost models were decreased by the RMSE values of the DMPNN+XGBoost models with the tuned parameters as “Diff(DMPNN)” and “Diff-(DMPNN+XGBoost)”, respectively. (b) A good classification model tended to have a large AUC value. So, the AUC values of the DMPNN+XGBoost model with the tuned parameters were decreased by those of the DMPNN models as “Diff(DMPNN)” and the DMPNN+XGBoost models as “Diff(DMPNN+XGBoost)”. So, the models were improved if they had positive values in both (a) and (b).

Except for the ToxCast data set, the duet model of DMPNN +XGBoost achieved the best performances on 9 out of 10 data sets, as shown in Figure 5. This is similar to the above observation that the ToxCast data set seems to be difficult to be improved by the classifier XGBoost and its parameter tuning. The DMPNN+XGBoost with the tuned parameters performed slightly worse than the DMPNN model, with a minor decrease of 0.007 in the AUC value.

The AUC value of the DMPNN model may be improved by as large as 0.027 on the BACE data set using the DMPNN +XGBoost model with the tuned parameters, and the largest improvement of 0.057 in the AUC value was achieved by simply tuning the parameters of the DMPNN+XGBoost model on the Clintox data set. The parameter tuning improved the regression FreeSolv data set with the largest improvements. The improvements of 0.087 and 0.048 in the RMSE values were achieved for the DMPNN model and the DMPNN +XGBoost model, respectively.

Comparison with Two Studies. Two studies were evaluated on the same data sets in this study, as shown in Figure 6. Jeo and Kim proposed an efficient molecular feature learning algorithm FP2VEC, inspired by the representation learning strategy in the field of natural language processing.[56] The final FP2VEC features were used to train a CNN model and achieved competitive results on the QSAR (quantitative structure−activity relationship) task. We reproduced this study using the released open-source codes and denoted this study as FP2VEC+CNN.



(a)

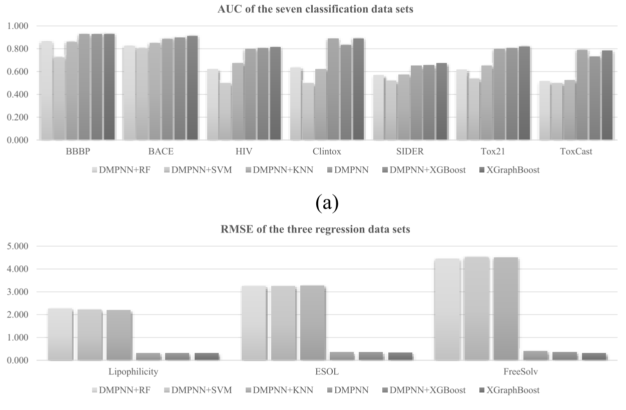
(b)

Figure 6. The performance comparison of this study (DMPNN +XGBoost) with the two studies FP2VEC+CNN and DNN+CM. The horizontal axis listed the data sets, and the vertical axis gave the performance metrics: (a) AUC values of seven classification data sets and (b) RMSE values of the three regression data sets.

Hou et al. investigated the molecular property prediction problems using multiple neural networks, including a single-layer neural network (SLNN), a multilayer deep neural network (DNN), and a convolutional neural network (CNN). The authors used the Coulomb matrix (CM) format to represent the molecules and trained the prediction models using this input data format.[57] A three-layer DNN with the CM data input achieved the best result, which was denoted as DNN+CM, which was reproduced on the same data sets in this study.

The proposed model in this study was DMPNN+XGBoost, and it outperformed these two studies FP2VEC+CNN and DNN+CM on both classification and regression models of the molecular property prediction problems, as shown in Figure 6. The largest improvement of 45.4% in AUC over the FP2VEC +CNN algorithm was achieved on the data set ToxCast, and this study achieved an improvement of 2.175 in RMSE for the regression data set FreeSolv. The DNN+CM algorithm achieved the worst average performances for the molecular prediction data sets, as shown in Figure 6. DNN+CM only achieved RMSE = 50.898 on the regression data set ESOL, while the other two algorithms DMPNN+XGBoost and DNN +CM achieved 0.345 and 1.274 in RMSE, respectively.

Evaluation of Different Supervised Learners. This study replaced the output layer of the original DMPNN models with a conventional supervised learner XGBoost and tuned the parameters of the integrated framework for the molecular property prediction problems, as shown in Figure 7.



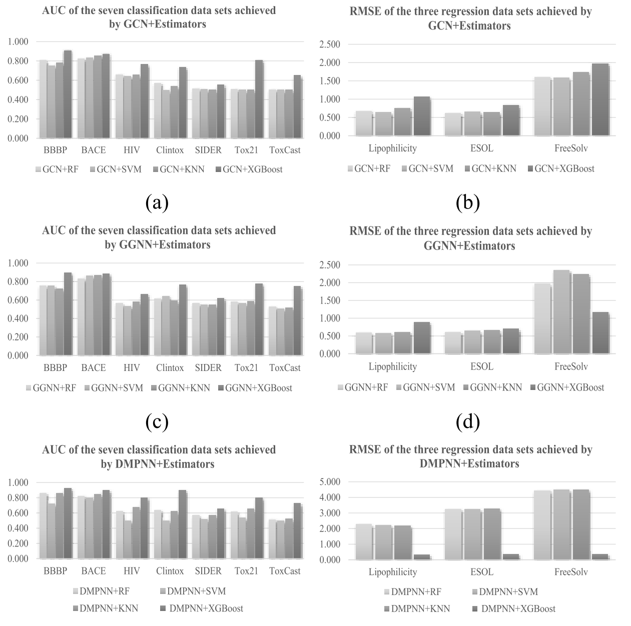
(a)

(b)

Figure 7. The performance comparison with different supervised learners. The horizontal axis listed the data sets, and the vertical axis gave the performance metrics: (a) AUC values of seven classification data sets and (b) RMSE values of the three regression data sets. The DMPNN-extracted features were evaluated using different supervised learners, including Random Forest (RF), Support Vector Machine (SVM), k nearest neighbor (KNN), and XGBoost. The notation "DMPNN" used the output layer of the original DMPNN model, and the notation "XGraphBoost" gave the results of the DMPNN-extracted features and the supervised learner XGBoost and the parameter tuning.

The original DMPNN algorithm outperformed the other graph networks like GCN and GGNN for the molecular prediction problem, but this study hypothesized that the output layer of the original DMPNN model may be further improved by a conventional supervised learner. Figure 7 showed that the conventional algorithms RF, SVM, and KNN did not improve the model performances of the DMPNN models. The supervised learner XGBoost only achieved the performance improvements on 7 out of 10 data sets, but after further tuning the parameters of the framework DMPNN+XGBoost, the proposed algorithm XGraphBoost achieved performance improvements compared with the DMPNN models on all 10 data sets.

Collaborations between Graph Representations and Supervised Learners. The collaborating duets between the graph representation algorithms and the supervised learners were evaluated in Figure 8. We may observe that except for the duet GGNN+XGBoost on the classification data set ToxCast, the best performing duet was always DMPNN+XGBoost, but the duet GGNN+XGBoost only achieved 0.664 in AUC on the classification data set HIV, compared with the AUC = 0.803 for the duet DMPNN+XGBoost.



(a) (b)

(c) (d)

(e) (f)

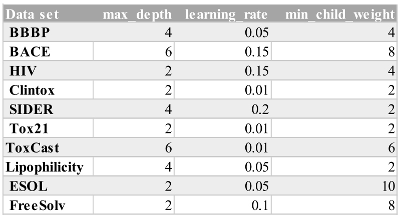
Figure 8. How different supervised learners impact the molecular property prediction models using different graph representations. The horizontal axis listed the data sets, and the vertical axis gave the performance metrics, i.e., AUC values of seven classification data sets or RMSE values of the three regression data sets. Four supervised learners were evaluated, including Random Forest (RF), Support Vector Machine (SVM), k nearest neighbor (KNN), and XGBoost. The GCN-extracted features were evaluated for (a) the classification and (b) regression data sets. The GGNN-extracted features were also evaluated for (c) the classification and (d) regression data sets.

So overall, the collaboration between the DMPNN-extracted features and the supervised learner XGBoost serves as a good molecular property prediction framework (denoted as XGraphBoost in this study).

■ CONCLUSION

A proof-of-principle study was carried out about the integration of the graph neural network (GNN) and XGBoost for the molecular property prediction problem. A comprehensive evaluation was conducted to demonstrate that the conventional Morgan fingerprint features may be improved by the GNN-extracted features, while the best duet was the integration of the DMPNN-extracted features and the supervised learner XGBoost. The DMPNN+XGBoost model may be further improved by tuning the parameters of the algorithm XGBoost. We believe that the proposed framework XGraphBoost may efficiently improve the prediction problems of various molecular properties, which may be of interest to both academic and industrial communities. In order to ensure the replicability of this study, the parameter values of the supervised learner XGBoost for the data sets were given in Figure 9.

The main contribution of this study was the utilization of the supervised learner XGBoost in replacing the output layer of the DMPNN model. The delicate tuning of the model parameters also showed performance improvements in predicting the molecular properties.



Data set	max_depth	learning_rate	min_child_weight
BBBP	4	0.05	4
BACE	6	0.15	8
HIV	2	0.15	4
Clintox	2	0.01	2
SIDER	4	0.2	2
Tox21	2	0.01	2
ToxCast	6	0.01	6
Lipophilicity	4	0.05	2
ESOL	2	0.05	10
FreeSolv	2	0.1	8

Figure 9. Optimized values of the three parameters of the 10 data sets. The columns "max_depth", "learning_rate", and "min_child_weight" gave the values of these three parameters for each data set.

The proposed algorithm XGraphBoost used the SMILES codes to represent molecules, so it also has the limitation induced by the SMILES codes, e.g., the limited capability in describing three-dimensional conformations.

This study demonstrated that conventional machine learning models may improve the prediction performances of graph neural networks. Due to the limited numbers of samples in the individual molecular property prediction data sets, the graph neural networks may be further improved by the strategies of transfer learning[58] or few-shot learning,[59,60] etc. For example, a natural language processing network ChemBERTa was recently released as a pretrained model.[58] X-MOL achieved state-of-the-art results on all these molecular analysis tasks and proved the beneficial effect of utilizing the large-scale pretraining models on learning molecular representations and downstream tasks.[61]

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.0c01489.

Visualization plots generated on all 10 data sets for three graph neural networks (GCN, GGNN, and DMPNN) (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Fengfeng Zhou − *College of Computer Science and Technology, and Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, Jilin 130012, P.R. China;* orcid.org/0000-0002-8108-6007; Phone: +86-431-8516-6024; Email: FengfengZhou@gmail.com, ffzhou@jlu.edu.cn; Fax: +86-431-8516-6024; http://www.healthinformaticslab.org/

Xiaojian Wang − *State Key Laboratory of Bioactive Substances and Functions of Natural Medicines, Institute of Materia Medica, Peking Union Medical College and Chinese Academy of Medical Sciences, Beijing 100050, P.R. China;* orcid.org/0000-0002-1856-8820; Email: wangxiaojian@imm.ac.cn

Authors

Daiguo Deng − *Fermion Technology Co., Ltd., Guangzhou, Guangdong 510000, P.R. China*

Xiaowei Chen − *Fermion Technology Co., Ltd., Guangzhou, Guangdong 510000, P.R. China*

Ruochi Zhang − *Fermion Technology Co., Ltd., Guangzhou, Guangdong 510000, P.R. China; College of Computer Science and Technology, and Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, Jilin 130012, P.R. China*

Zengrong Lei − *Fermion Technology Co., Ltd., Guangzhou, Guangdong 510000, P.R. China*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.0c01489

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This study was not supported by any grants. The insightful comments from the anonymous reviewers are greatly appreciated.

■ REFERENCES

(1) Burrage, K.; Tian, T.; Burrage, P. A multi-scaled approach for simulating chemical reaction systems. *Prog. Biophys. Mol. Biol.* **2004**, *85* (2−3), 217−34.

(2) Xue, J.; Guijas, C.; Benton, H. P.; Warth, B.; Siuzdak, G. METLIN MS(2) molecular standards database: a broad chemical and biological resource. *Nat. Methods* **2020**, *17* (10), 953−954.

(3) Capecchi, A.; Awale, M.; Probst, D.; Reymond, J. L. PubChem and ChEMBL beyond Lipinski. *Mol. Inf.* **2019**, *38* (5), 1900016.

(4) Deb, P. K. Recent updates in the computer aided drug design strategies for the discovery of agonists and antagonists of adenosine receptors. *Curr. Pharm. Des.* **2019**, *25* (7), 747−749.

(5) Velmurugan, D.; Pachaiappan, R.; Ramakrishnan, C. Recent Trends in Drug Design and Discovery. *Curr. Top. Med. Chem.* **2020**, *20* (19), 1761−1770.

(6) Cherkasov, A.; Muratov, E. N.; Fourches, D.; Varnek, A.; Baskin, I. I.; Cronin, M.; Dearden, J.; Gramatica, P.; Martin, Y. C.; Todeschini, R.; Consonni, V.; Kuz'min, V. E.; Cramer, R.; Benigni, R.; Yang, C.; Rathman, J.; Terfloth, L.; Gasteiger, J.; Richard, A.; Tropsha, A. QSAR modeling: where have you been? Where are you going to? *J. Med. Chem.* **2014**, *57* (12), 4977−5010.

(7) Tropsha, A. Best Practices for QSAR Model Development, Validation, and Exploitation. *Mol. Inf.* **2010**, *29* (6−7), 476−88.

(8) Tao, L.; Zhang, P.; Qin, C.; Chen, S. Y.; Zhang, C.; Chen, Z.; Zhu, F.; Yang, S. Y.; Wei, Y. Q.; Chen, Y. Z. Recent progresses in the exploration of machine learning methods as in-silico ADME prediction tools. *Adv. Drug Delivery Rev.* **2015**, *86*, 83−100.

(9) Ma, J.; Sheridan, R. P.; Liaw, A.; Dahl, G. E.; Svetnik, V. Deep neural nets as a method for quantitative structure-activity relationships. *J. Chem. Inf. Model.* **2015**, *55* (2), 263−74.

(10) Mitchell, J. B. Machine learning methods in chemoinformatics. *Wiley. Interdiscip. Rev. Comput. Mol. Sci.* **2014**, *4* (5), 468−481.

(11) Dudek, A. Z.; Arodz, T.; Galvez, J. Computational methods in developing quantitative structure-activity relationships (QSAR): a review. *Comb. Chem. High Throughput Screening* **2006**, *9* (3), 213−28.

(12) Olayan, R. S.; Ashoor, H.; Bajic, V. B. DDR: efficient computational method to predict drug−target interactions using graph mining and machine learning approaches. *Bioinformatics* **2018**, *34* (7), 1164−1173.

(13) Watson, O. P.; Cortes-Ciriano, I.; Taylor, A. R.; Watson, J. A. A decision-theoretic approach to the evaluation of machine learning algorithms in computational drug discovery. *Bioinformatics* **2019**, *35* (22), 4656−4663.

(14) Yao, X. J.; Panaye, A.; Doucet, J. P.; Zhang, R. S.; Chen, H. F.; Liu, M. C.; Hu, Z. D.; Fan, B. T. Comparative study of QSAR/QSPR correlations using support vector machines, radial basis function neural networks, and multiple linear regression. *J. Chem. Inf. Comput. Sci.* **2004**, *44* (4), 1257−66.

(15) Svetnik, V.; Liaw, A.; Tong, C.; Culberson, J. C.; Sheridan, R. P.; Feuston, B. P. Random forest: a classification and regression tool for compound classification and QSAR modeling. *J. Chem. Inf. Comput. Sci.* **2003**, *43* (6), 1947−58.

(16) Maleki, A.; Daraei, H.; Alaei, L.; Faraji, A. Comparison of QSAR models based on combinations of genetic algorithm, stepwise multiple linear regression, and artificial neural network methods to predict Kd of some derivatives of aromatic sulfonamides as carbonic anhydrase II inhibitors. *Bioorg. Khim.* **2014**, *40* (1), 70−84.

(17) Gong, P.; Zhang, C.; Chen, M. Editorial: Deep Learning for Toxicity and Disease Prediction. *Front. Genet.* **2020**, *11*, 175.

(18) Lavecchia, A. Deep learning in drug discovery: opportunities, challenges and future prospects. *Drug Discovery Today* **2019**, *24* (10), 2017−2032.

(19) Mamoshina, P.; Vieira, A.; Putin, E.; Zhavoronkov, A. Applications of Deep Learning in Biomedicine. *Mol. Pharmaceutics* **2016**, *13* (5), 1445−54.

(20) Francoeur, P. G.; Masuda, T.; Sunseri, J.; Jia, A.; Iovanisci, R. B.; Snyder, I.; Koes, D. R. Three-Dimensional Convolutional Neural Networks and a Cross-Docked Data Set for Structure-Based Drug Design. *J. Chem. Inf. Model.* **2020**, *60* (9), 4200−4215.

(21) Wang, X.; Li, Z.; Jiang, M.; Wang, S.; Zhang, S.; Wei, Z. Molecule Property Prediction Based on Spatial Graph Embedding. *J. Chem. Inf. Model.* **2019**, *59* (9), 3817−3828.

(22) Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular graph convolutions: moving beyond fingerprints. *J. Comput.-Aided Mol. Des.* **2016**, *30* (8), 595−608.

(23) St. John, P. C.; Phillips, C.; Kemper, T. W.; Wilson, A. N.; Guan, Y.; Crowley, M. F.; Nimlos, M. R.; Larsen, R. E. Message-passing neural networks for high-throughput polymer screening. *J. Chem. Phys.* **2019**, *150* (23), 234111.

(24) Jiang, J.; Wang, R.; Wang, M.; Gao, K.; Nguyen, D. D.; Wei, G.-W. Boosting tree-assisted multitask deep learning for small scientific datasets. *J. Chem. Inf. Model.* **2020**, *60* (3), 1235−1244.

(25) Dasoulas, G.; Santos, L. D.; Scaman, K.; Virmaux, A. Coloring graph neural networks for node disambiguation. 2019, *arXiv preprint arXiv:1912.06058*. https://arxiv.org/abs/1912.06058 (accessed 2021-05-02).

(26) Shi, Q.; Chen, W.; Huang, S.; Wang, Y.; Xue, Z. Deep learning for mining protein data. *Brief. Bioinform.* **2021**, *22*, 194−218.

(27) Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. 2014, *arXiv preprint arXiv:1409.1556*. https://arxiv.org/abs/1409.1556 (accessed 2021-05-02).

(28) Roy, A.; Skolnick, J. LIGSIFT: an open-source tool for ligand structural alignment and virtual screening. *Bioinformatics* **2015**, *31* (4), 539.

(29) Abdo, A.; Chen, B.; Mueller, C.; Salim, N.; Willett, P. Ligand-based virtual screening using bayesian networks. *J. Chem. Inf. Model.* **2010**, *50* (6), 1012.

(30) Ewing, T.; Baber, J. C.; Feher, M. Novel 2D fingerprints for ligand-based virtual screening. *J. Chem. Inf. Model.* **2006**, *46* (6), 2423.

(31) Lu, J.; Hou, X.; Wang, C.; Zhang, Y. Incorporating explicit water molecules and ligand conformation stability in machine-learning scoring functions. *J. Chem. Inf. Model.* **2019**, *59* (11), 4540.

(32) Raha, K.; Merz, K. M. Large-scale validation of a quantum mechanics based scoring function: predicting the binding affinity and the binding mode of a diverse set of protein− ligand complexes. *J. Med. Chem.* **2005**, *48* (14), 4558.

(33) McGaughey, G. B.; Sheridan, R. P.; Bayly, C. I.; Culberson, J. C.; Kreatsoulas, C.; Lindsley, S.; Maiorov, V.; Truchon, J.-F.; Cornell, W. D. Comparison of topological, shape, and docking methods in virtual screening. *J. Chem. Inf. Model.* **2007**, *47* (4), 1504.

(34) Zheng, S.; Yan, X.; Yang, Y.; Xu, J. Identifying structure−property relationships through SMILES syntax analysis with self-attention mechanism. *J. Chem. Inf. Model.* **2019**, *59* (2), 914.

(35) Olivecrona, M.; Blaschke, T.; Engkvist, O.; Chen, H. Molecular de-novo design through deep reinforcement learning. *J. Cheminf.* **2017**, *9* (1), 48.

(36) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M. Analyzing learned molecular representations for property prediction. *J. Chem. Inf. Model.* **2019**, *59* (8), 3370.

(37) Wu, Z.; Lei, T.; Shen, C.; Wang, Z.; Cao, D.; Hou, T. ADMET evaluation in drug discovery. 19. Reliable prediction of human cytochrome P450 inhibition using artificial intelligence approaches. *J. Chem. Inf. Model.* **2019**, *59* (11), 4587.

(38) Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **2010**, *50* (5), 742.

(39) Jorissen, R. N.; Gilson, M. K. Virtual screening of molecular databases using a support vector machine. *J. Chem. Inf. Model.* **2005**, *45* (3), 549.

(40) Bruce, C. L.; Melville, J. L.; Pickett, S. D.; Hirst, J. D. Contemporary QSAR classifiers compared. *J. Chem. Inf. Model.* **2007**, *47* (1), 219.

(41) Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. *Advances in Neural Information Processing Systems 28 (NIPS 2015)* **2015**, 2224−2232.

(42) Lin, X.; Quan, Z.; Wang, Z.-J.; Ma, T.; Zeng, X. Kgnn: Knowledge graph neural network for drug-drug interaction prediction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*; 2020; pp 2739−2745,.

(43) Li, B.; Wang, W.; Sun, Y.; Zhang, L.; Ali, M. A.; Wang, Y. *GraphER: Token-Centric Entity Resolution with Graph Convolutional Neural Networks* **2020**, *34*, 8172−8179.

(44) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: a benchmark for molecular machine learning. *Chemical science* **2018**, *9* (2), 513−530.

(45) Mayr, A.; Klambauer, G.; Unterthiner, T.; Steijaert, M.; Wegner, J. K.; Ceulemans, H.; Clevert, D.-A.; Hochreiter, S. Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chemical science* **2018**, *9* (24), 5441−5451.

(46) Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Model.* **1988**, *28* (1), 31.

(47) Landrum, G. *RDKit: Open-source cheminformatics*; 2006.

(48) Ruiz, L.; Gama, F.; Ribeiro, A. Gated graph recurrent neural networks. 2020, *arXiv preprint arXiv:2002.01038*. https://arxiv.org/abs/2002.01038 (accessed 2021-05-02).

(49) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural message passing for quantum chemistry. 2017, *arXiv preprint arXiv:1704.01212*. https://arxiv.org/abs/1704.01212 (accessed 2021-05-02).

(50) Swanson, K. *Message passing neural networks for molecular property prediction*; Massachusetts Institute of Technology, 2019.

(51) Friedman, J.; Hastie, T.; Tibshirani, R. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *Annals of Statistics* **2000**, *28* (2), 337−407.

(52) Friedman, J. H. Greedy function approximation: a gradient boosting machine. *Annals of Statistics* **2001**, *29*, 1189−1232.

(53) Chen, T.; He, T.; Benesty, M.; Khotilovich, V.; Tang, Y. Xgboost: extreme gradient boosting. *R Package Version 0.4-2*; 2015; pp 1−4.

(54) Wang, J.; Cao, D.; Tang, C.; Chen, X.; Sun, H.; Hou, T. Fast and Accurate Prediction of Partial Charges Using Atom-Path-Descriptor-based Machine Learning. *Bioinformatics* **2020**, *36*, 4721.

(55) Wei, Z.; Ding, S.; Duan, M.; Liu, S.; Huang, L.; Zhou, F. FeSTwo, a two-step feature selection algorithm based on feature engineering and sampling for the chronological age regression problem. *Comput. Biol. Med.* **2020**, *125*, 104008.

(56) Jeon, W.; Kim, D. FP2VEC: a new molecular featurizer for learning molecular properties. *Bioinformatics* **2019**, 35 (23), 4979−4985.

(57) Hou, F.; Wu, Z.; Hu, Z.; Xiao, Z.; Wang, L.; Zhang, X.; Li, G. Comparison Study on the Prediction of Multiple Molecular Properties by Various Neural Networks. *J. Phys. Chem. A* **2018**, *122* (46), 9128−9134.

(58) Chithrananda, S.; Grand, G.; Ramsundar, B. ChemBERTa: Large-Scale Self-Supervised Pretraining for Molecular Property

Prediction. 2020, *arXiv preprint arXiv:2010.09885*. https://arxiv.org/abs/2010.09885 (accessed 2021-05-02).

(59) Guo, Z.; Zhang, C.; Yu, W.; Herr, J.; Wiest, O.; Jiang, M.; Chawla, N. V. Few-Shot Graph Learning for Molecular Property Prediction. 2021, *arXiv preprint arXiv:2102.07916*. https://arxiv.org/abs/2102.07916 (accessed 2021-05-02).

(60) Pappu, A.; Paige, B. Making Graph Neural Networks Worth It for Low-Data Molecular Machine Learning. 2020, *arXiv preprint arXiv:2011.12203*. https://arxiv.org/abs/2011.12203 (accessed 2021-05-02).

(61) Xue, D.; Zhang, H.; Xiao, D.; Gong, Y.; Chuai, G.; Sun, Y.; Tian, H.; Wu, H.; Li, Y.; Liu, Q. X-MOL: large-scale pre-training for molecular understanding and diverse molecular analysis. 2021, *bioRxiv*. https://www.biorxiv.org/content/10.1101/2020.12.23.424259v2.full (accessed 2021-05-02).