A Comparative Study of Marginalized Graph Kernel and Message Passing Neural Network

Yan Xiang[a], Yu-Hang Tang[b], Guang Lin*[c], Huai Sun*[a]

[a]*School of Chemistry and Chemical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*

[b]*Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720, United States*

[c]*Department of Mathematics & School of Mechanical Engineering, Purdue University, West Lafayette, Indiana 47907, United States*

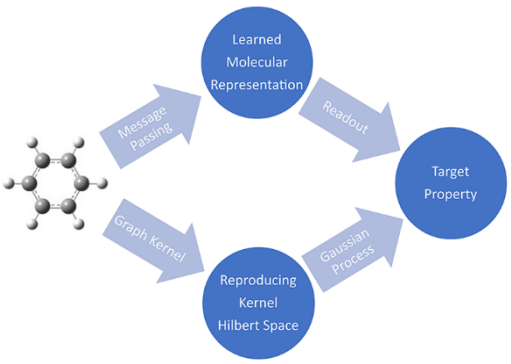
AUTHOR INFORMATION

* Corresponding Authors

E-mail address: huaisun@sjtu.edu.cn, guanglin@purdue.edu

Abstract

This work proposes a state-of-the-art hybrid kernel to calculate molecular similarity. Combining with Gaussian process models, the performance of the hybrid kernel in predicting molecular properties is comparable to that of the Directed Message Passing Neural Network (D-MPNN). The hybrid kernel consists of a marginalized graph kernel (MGK) and a radial basis function (RBF) kernel that operates on molecular graphs and global molecular features, respectively. Bayesian optimization was used to get the optimal hyperparameters for both models. The comparisons are performed on 11 publicly available data sets. Our results show that the predictions of both models are correlated with similar performance, and the ensemble prediction of both models performs better than either of them. Through principal component analysis, we found that the features extracted by the hybrid kernel are similar to those extracted by D-MPNN. The advantage of D-MPNN lies in computational efficiency, while the advantage of the graph kernel models lies in the inherent uncertainty quantification and accurate uncertainty quantification.



I. INTRODUCTION

Predicting molecular properties is one of the central topics of cheminformatics that has attracted widespread attention for decades. This field is rejuvenated due to the advances in Graph neural networks (GNNs) recently[1]. Numerous methods have been developed for decades. To evaluate the quality of different methods, Wu et al. introduced a large-scale benchmark for molecular property predictions, *MoleculeNet,*[2] which provides multiple public data sets, data splitting, as well as the implementation of popular algorithms of molecular featurization and learning algorithms. Their results demonstrated that graph-based models outperform molecular fingerprints methods in most data sets. GNNs have achieved impressive success in predicting quantum mechanical properties, physicochemical properties, biological activity, and toxicity.[3–13]

Nevertheless, Yang et al. showed that a hybrid molecular representation that combines Directed Message Passing Neural Network (D-MPNN) and expert-crafted descriptors is superior to using either one model alone in extensive comparisons on 19 public and 16 proprietary data sets.[14] Loukas demonstrated that when the amount of data is sufficient, the depth and width of the message passing neural networks (MPNNs) need to increase at least polynomially with the size of the graph to distinguish the graphs.[15] However, in predicting molecular properties the optimal performance is usually achieved within a few message-passing steps. Therefore, the molecular representations

learned through message passing are fundamentally local, and it is beneficial to introduce features that describe the molecular global features. We speculate that the gap lies in the fact that the real data sets are always insufficient. In the results of the 2021 KDD Cup Large-scale Challenge (OGB-LSC), the depth GNNs perform better for the PCQM4M-LSC data set containing about 4 million molecules.[16]

Like GNNs, graph kernel is a branch of graph-based machine learning methods.[17–26] Marginalized graph kernel (MGK) is a random walk graph kernel based on the Weisfeiler-Lehman isomorphism test. Compared to GNNs, graph kernel has received less attention due to the computational cost and programming difficulty. Recently, Tang et al. developed the GraphDot software package,[27] which uses GPUs to compute MGK matrix efficiently.[28] Using GraphDot, Tang and de Jong presented an MGK for molecular atomization energy prediction.[29] Xiang et al. developed normalized marginalized graph kernels (nMGK) for predictions of thermodynamic properties of pure organic liquids.[30]

Naturally, it is interesting to compare the two different Weisfeiler-Lehman approaches, *i.e.*, MPNNs and MGK, to understand their advantages and disadvantages. In this work, we evaluate the performance of MGK coupled with Gaussian process regression and classification (GP-MGK) using the data sets commonly used in benchmark studies.[2] The D-MPNN[14] is used as the comparison model. In both GP-MGK and D-MPNN, global molecular features

are incorporated and Bayesian optimization is used to optimize the hyperparameters. We compared and analyzed the performances of both methods on 7 regression and 4 classification data sets. We also provide suggestions for practical applications of GP-MGK and D-MPNN.

II. METHODS

Marginalized Graph Kernel Method

In MGK, molecules are represented by undirected labeled graphs, where vertices represent atoms, and edges represent chemical bonds. The MGK, which computes the molecular similarity, consists of five parts, namely atom microkernels, bond microkernels, a starting probability distribution, a stopping probability distribution, and a transition probability matrix. The atom and bond microkernels are further composed of elementary kernels, which act on individual features, using rules such as addition, tensor product, and R-convolution.[31]

The atom and bond features are listed in Tables 1 and 2. For features that are discrete variables, the associated elementary kernel is an elevated Kronecker delta function:

$$\delta(\phi_1, \phi_2) = \begin{cases} 1 & , \phi_1 = \phi_2 \\ h \in (0,1), \text{otherwise.} \end{cases} \quad (1)$$

For features that are a list of discrete variables with variable lengths, the associated elementary kernel is a sequence convolution of Kronecker deltas:

$$C(l_1, l_2) = \frac{f(l_1, l_2)}{\sqrt{f(l_1, l_1) f(l_2, l_2)}}, \tag{2}$$

where

$$f(l_1, l_2) = \sum_{\phi_1 \in l_1} \sum_{\phi_2 \in l_2} \delta(\phi_1, \phi_2)\,. \tag{3}$$

Here, l_1, l_2 are two variable-length feature vectors, and *h* is a hyperparameter that determines the tolerance of different feature values. If *h* is too small, MGK will be too strict, and the return value of two different molecules will be close to 0. If *h* is too large, the difference between atoms and bonds will be ignored, for example, propanol and butane cannot be distinguished.

The microkernel for atoms or bonds used in this study is a weighted addition of elementary kernels between individual features:

$$\kappa_{\mathrm{v}}(v, v') = \frac{\sum_j c_j \mu_j \left(\phi_j(v), \phi_j(v')\right)}{\sum_j c_j}, \tag{4}$$

$$\kappa_{\mathrm{e}}(e, e') = \frac{\sum_j c_j \mu_j \left(\phi_j(e), \phi_j(e')\right)}{\sum_j c_j}, \tag{5}$$

where μ_j is the elementary kernel for the *j*th feature ϕ_j, and c_j is a weight hyperparameter that determines the importance of the feature.

The starting probability of an atom is a weighted addition of elementary probabilities:

$$p_{\mathrm{s}}(v) = 1.0 + \sum_{k \in \mathcal{K}} p_{\mathrm{k}}(v)\,, \tag{6}$$

$$p_{\mathrm{k}}(v) = \begin{cases} p, v \text{ in group k} \\ 0, \quad \text{otherwise.} \end{cases} \tag{7}$$

where p_{k} is the elementary probability for the group k and *p* is the hyperparameter that determines the importance of this group. Groups can be

defined arbitrarily, and we use atom types $\mathcal{K} = \{\mathrm{B}, \mathrm{C}, \mathrm{N}, \mathrm{O}, \mathrm{F}, \mathrm{Si}, \mathrm{P}, \mathrm{S}, \mathrm{Cl}, \mathrm{Br}, \mathrm{I}\}$ in practice.

The stopping probability is set to be a hyperparameter p_q which is the same for all elements. The transition probability is set to $1/n$ where *n* is the number of neighbors to the current atom.

The MGK computes the expectation of path similarities from a simultaneous random walk process on a pair of graphs *G* and *G'*:

$$K(G,G') = \sum_{\ell=1}^{\infty} \sum_{\mathbf{h}} \sum_{\mathbf{h}'} \left[\begin{array}{l} p_{\mathrm{s}}(h_1) p'_{\mathrm{s}'}(h'_1) \kappa_{\mathrm{v}}\left(v_{h_1}, v'_{h'_1}\right) p_{\mathrm{q}}(h_\ell) p'_{\mathrm{q}}(h'_\ell) \times \\ \left(\prod_{i=2}^{\ell} p_{\mathrm{t}}(h_i | h_{i-1}) \right) \left(\prod_{j=2}^{\ell} p'_{\mathrm{t}}(h'_i | h'_{i-1}) \right) \times \\ \left(\prod_{k=2}^{\ell} \kappa_{\mathrm{v}}\left(v_{h_k}, v'_{h'_k}\right) \kappa_{\mathrm{e}}\left(e_{h_k h_{k-1}}, e'_{h'_k h'_{k-1}}\right) \right) \end{array} \right], \qquad (8)$$

Where **h** and **h**' are the random walk paths of length l. A linear algebra transformation allows the fast numerical evaluation of eq 8 using $O(|h||h'|)$ time.

The MGK can be normalized with an exponential factor:

$$\bar{K}(G,G') = F \frac{K(G,G')}{\sqrt{K(G,G)K(G',G')}} \exp\left[-\frac{\left(K(G,G) - K(G',G')\right)^2}{\lambda^2} \right], \qquad (9)$$

where *F* and λ are the hyperparameters.

For more detailed information about MGK, we refer a reader to references.[18,28–30] Gaussian processes are used for regression and classification tasks.[32]

Directed Message Passing Neural Network

The D-MPNN is used as the comparison model in this work. Herein, we briefly introduce the model.[14]

The initial atom features x_v and bond features e_{vw} used in D-MPNN are listed in Tables 3 and 4. The initial edge hidden states are:

$$h_{vw}^0 = \tau\big(W_i \,\mathrm{cat}(x_v, e_{vw})\big), \tag{10}$$

where $\mathrm{cat}(x_v, e_{vw})$ is the concatenated vector of the atom features x_v and the bond features e_{vw}, W_i is a learned matrix, and τ is the ReLU activation function.[33]

The message passing update equations are

$$m_{vw}^{t+1} = \sum_{k \in \{N(v) \setminus w\}} h_{kv}^t, \tag{11}$$

$$h_{vw}^{t+1} = \tau(h_{vw}^0 + W_m m_{vw}^{t+1}), \tag{12}$$

where $N(v)$ are the neighbors of *v*, W_m is a learned matrix. The learned atom hidden states are

$$m_v = \sum_{w \in N(v)} h_{vw}^T, \tag{13}$$

$$h_v = \tau\big(W_a \,\mathrm{cat}(x_v, m_v)\big). \tag{14}$$

The molecular representation is the mean of atom hidden states

$$h = \frac{1}{|G|} \sum_{v \in G} h_v. \tag{15}$$

The final property is obtained through a feed-forward neural network $f(\cdot)$,

$$\hat{y} = f(h). \tag{16}$$

By training several copies of D-MPNN with different initial weights, the ensemble (averaged) prediction of these models is used as the final prediction.

For more detailed information about D-MPNN, we refer a reader to reference.[14]

RDKit-Calculated Features

The overview of GP-MGK and D-MPNN used in this study are sketched in Figure 1. Both of them use a hybrid molecular representation of graphs and descriptors. In Yang et al.'s work, 200 global features that can be rapidly computed using RDKit were concatenated with the learned molecular representation through message passing, which significantly improves the prediction performance.[14] To make a fair comparison, we added the 200 features in GP-MGK using a hybrid kernel:

$$K\big((G, F_{\mathrm{RDKit}}), (G', F'_{\mathrm{RDKit}})\big) = K_G(G, G')K_F(F_{\mathrm{RDKit}}, F'_{\mathrm{RDKit}}), \quad (17)$$

where G, G' are the molecular graphs and F_{RDKit}, F'_{RDKit} are RDKit features. K_G is the MGK described above and K_F is the radial basis function kernel $K_F(F_{\mathrm{RDKit}}, F'_{\mathrm{RDKit}}) = \exp\left(-\frac{\|F_{\mathrm{RDKit}} - F'_{\mathrm{RDKit}}\|^2}{2\sigma^2}\right)$. For the sake of simplicity, MGK and D-MPNN represent variants that incorporate the RDKit features in the following text.

Implementation

All codes for the GP-MGK are available in our GitHub repository.[34] We use the GraphDot[27] python package to compute the marginalized graph kernels and perform GPR. We use the scikit-learn package to carry out GPC, principal component analysis (PCA), and kernel PCA.[35] We use the Descriptatorus

package[36] to calculate the RDKit features and Hyperopt package[37] to optimize hyperparameters.

III. EXPERIMENTS

Data sets

The publicly available data sets used in this study are listed in Table 5. These data sets are commonly used for benchmark studies in molecular property prediction.[2,14]

Hyperparameters Optimization

There are 48 hyperparameters for Gaussian process regression (GPR), 47 hyperparameters for Gaussian process classification (GPC), and four hyperparameters for D-MPNN. We use Tree of Parzen Estimators (TPE) to optimize hyperparameters to obtain optimal performance.[38,39]

For GPR, we use different random seeds to perform Bayesian optimization repeatedly 20 times, with 100 iterations for each optimization. The best hyperparameters with the smallest leave-one-out loss are selected. The optimal hyperparameters are listed in Table S1. For GPC, the data is split 10 times at a ratio of 80:20, and Bayesian optimizations of 100 iterations are performed to determine the best hyperparameters based on the averaged performance on test sets of the 10 data splits. The optimal hyperparameters are listed in Table S2. It is noticed that the hyperparameter F in the last row is fixed in Bayesian

optimization since it does not affect the predicted value but scales the magnitude of the predictive uncertainty. As we will discuss below, it is adjusted by minimizing the miscalibration area.

For D-MPNN, we optimized the hyperparameters following the setting of Yang et al. The data is split 10 times at a ratio of 80:10:10, and Bayesian optimizations of 20 iterations are performed to determine the best hyperparameters based on the averaged performance on validation sets of the 10 data splits.[14] The optimal hyperparameters are listed in Tables S3 and S4.

Data Splits and Performance Evaluation

With the optimized hyperparameters, we evaluate both models on the same data splits. For each data set, we performed both random and scaffold data splits. The scaffold split is more challenging because the molecular scaffolds in the test set are not included in the training set. The data were divided into the training, validation, and test set according to the ratio of 80:10:10. The D-MPNN was trained for 50 epochs, and the model with the best performance on the validation set was used as the final model to make predictions on the test set. For the GP-MGK, we use the training set to build the model and make predictions on the test set. The data of the validation set is not used. The evaluation process was repeated 100 times.

Evaluation Metrics

For regression tasks, mean absolute error (MAE), root mean square error (RMSE), and R^2 are used. For classification tasks, area under the receiver operating characteristic curve (ROC-AUC) is used. For uncertainty quantification (UQ), negative log-likelihood (NLL) and miscalibration area are used.

In statistics, likelihood measures the goodness of fit of the model to a sample of data, and minimize negative log-likelihood (NLL) is commonly used as the loss function for UQ.[32,40]

Miscalibration area is one way to evaluate the UQ quality. An example is shown in Figure S1. We plot the confidence interval versus the percentage of the experimental value of the samples in the test set covered by the confidence interval curve, which is called the calibration curve. The miscalibration area is the area between the calibration curve and the diagonal.

IV. RESULTS AND DISCUSSION

We compared the performances of optimal GP-MGK and D-MPNN models. In this section, “GPR-MGK”, “GPC-MGK” refers to Gaussian process regression and classification with MGK. “D-MPNN Optimized” refers to the D-MPNN with RDKit features and optimized hyperparameters, and “D-MPNN Ensemble” refers to an ensemble of five “D-MPNN Optimized” models. “Ensemble” refers to a model that ensembling GPR-MGK (GPC-MGK for classification) and “D-MPNN Ensemble”.

Benchmark on Same Data Splits

It is important to compare different models on the same data splits, otherwise, contradictory results could be obtained due to random noise. This is illustrated by applying the GPR-MGK model using the ESOL data set. In Figure 2, the RMSE of the test set is plotted as a function of the repeated number of data splits. Each string is the statistical result of 100 individual runs with different random seeds. The difference between the best and worst results is 0.06, 0.02, 0.01, 0.006 for repeating times of 5, 25, 50, 100. Therefore, the same data splits are used to compare GP-MGK and D-MPNN models. Dwivedi et al. also held this viewpoint when benchmarking graph neural networks.[41]

GP-MGK VS D-MPNN: Prediction Performance

We first compare GPR-MGK with D-MPNN Ensemble on the ESOL data set. In Figure 3A,B, comparisons of predictions using GPR-MGK and D-MPNN Ensemble against the reference data are given, and the corresponding RMSE values are listed. The prediction performance of GPR-MGK and D-MPNN Ensemble is in the same level, and ensembling prediction by averaging both predictions is better. In Figure 3C,D, the prediction errors of GPR-MGK and D-MPNN Ensemble are compared, and an obvious correlation between them is observed. In more detail, we draw the difference between the predictions of

GPR-MGK and D-MPNN Ensemble for different molecules in Figure 3E,F. The gray area represents the standard deviation of the same molecule under different data splits. For most molecules, the predictions of GPR-MGK and D-MPNN Ensemble are consistent. The results for QM7, FreeSolv, Lipophilicity and PDBbind data sets are shown in Figures S2-S5. The correlation between the predictions of GPR-MGK and D-MPNN are observed for all data sets and both random and scaffold splits, indicating that the molecular features extracted through D-MPNN are similar to the features extracted using marginalized graph kernels. We think the correlation of GPR-MGK and D-MPNN comes from the fact that both models are based on the Weisfeiler-Lehman graph isomorphism test.

The regression results are numerically summarized in Table 6 and graphically summarized in the left of Figure 4. There are a total of 14 cases (seven data sets $\times$ two data split types). Compared with D-MPNN Optimized, GPR-MGK achieves better results in five cases, similar results in five cases, and poor results in four cases. Compared with D-MPNN Ensemble, GPR-MGK achieves better results in three cases, similar results in four cases, and poor results in seven cases. We emphasize that although the predictive abilities of GPR-MGK and D-MPNN are at the same level, their ensemble predictions are the best in 13 comparisons, except for the GPR-MGK on the QM7 dataset using scaffold splitting.

The classification results are numerically summarized in Table 7 and

graphically summarized on the right of Figure 4. For the BACE, BBBP, and SIDER data sets, the conclusion is the same as the above, that is, the performance of GPC-MGK is similar to D-MPNN Ensemble, and the ensemble prediction of GPC-MGK and D-MPNN Ensemble is the best. For the ClinTox dataset, D-MPNN outperforms GPC-MGK.

GP-MGK VS D-MPNN: Principal Component Analysis

We performed PCA[42,43] on the latent representations of D-MPNN, and kernel PCA[44] on the hybrid kernel. “D-MPNN latent 1” refers to the latent representations between the message passing phase and readout phase, and “D-MPNN latent 2” refers to the latent representations immediately before the output layer.

In Figure 5, the eigenvalue spectra of kernel matrices are plotted. For D-MPNN, the kernel matrices are computed by the element-wise dot product. The eigenvalues of D-MPNN latent 1 decay fastly, which indicates the molecular representation learned after message passing is of low rank. The eigenvalues of D-MPNN latent 2 decay slowly, indicating that the molecular representation is transformed from low-rank to high-rank in the readout phase. We think this is why Hirschfeld et al. use D-MPNN latent 2 as the input of union-based methods for UQ, rather than D-MPNN latent 1.[45] Based on the results in the previous section, the correlation of the predictions of MGK and D-MPNN indicates that

their molecular representations contain the same information. Relatively, D-MPNN latent 2 is more similar to MGK.

In Figure 6, we show the data embedding of the ESOL data set in the first two principal components (PC1 and PC2) of MGK and D-MPNN. In all molecular representations, PC1 and PC2 mainly contain two pieces of information: (1) whether the molecule is cyclic; (2) the number of heavy atoms contained in the molecule. In the data embedding of D-MPNN latent 1, cyclic and non-cyclic molecules are separated, and the distribution of the number of heavy atoms is messy. Compared with D-MPNN latent 1, the data embedding of MGK and D-MPNN latent 2 is very similar. A small part of cyclic and non-cyclic molecules overlap, and the distribution of the number of heavy atoms is ordered. The similarity may provide an indirect clue as to why the predictions of the two models are correlated. The data embedding of other data sets is shown in Figures S6-S15. It is noticed that the data embedding on PC1 and PC2 only contains part of the information, and the real space is high-dimensional.

GP-MGK VS D-MPNN: Uncertainty Quantification

GPR-MGK is a Bayesian inference method, and its prediction is Gaussian distribution. Therefore, the variance of the predicted Gaussian distribution can be used for UQ. This is crucial for the prediction of molecular property because it is currently impossible to have enough data to train an ML model that is

applicable to all molecules. Therefore, we need to understand the range of capabilities of the trained ML model through UQ. Hirschfeld et al. have implemented a series of UQ methods for D-MPNN, among which the top three models are D-MPNN RF (random forest), D-MPNN GP (Gaussian process), and D-MPNN MVE (mean-variance estimation).[45] In this work, we use D-MPNN MVE as a comparison model, because the former two need to retain a large amount of data as a validation set. D-MPNN MVE modifies the output layer of D-MPNN to mean and variance, and the loss function to NLL. In this section, we only use random split.

The predicted uncertainty obtained directly from the ML models is uncalibrated uncertainty, which means that it needs to be scaled by a factor to obtain a meaningful predictive variance. In GPR-MGK, the scale factor is equivalent to the hyperparameter F in eq 9. Figure 7 shows how NLL and miscalibration area vary with the scale factor on the ESOL data set. The NLL is not sensitive to the scale factor, so the optimal scale factors were obtained by minimizing the miscalibration area.

The NLL and the miscalibration area are the metrics for the overall evaluation of the quality of predictive uncertainty. More details of the UQ can be revealed by plotting the relationship between prediction errors and predicted uncertainty. The results of the ESOL data set are shown in Figure 8. In panels A and B, the prediction data are divided into 10 intervals according to predicted uncertainty. For each interval, the errors is plotted in the form of a violin shape,

where the horizontal bars represent the maximum, median, and minimum values, and the width represents the probability distribution. The data percentage, MAE, and R^2 of each interval are displayed below. In panels C and D, we plot the MAE of predictions as a function of predicted uncertainty, and the dashed line is the “ideal” MAE assuming that the truth values to be predicted perfectly obey the Gaussian distribution of predicted mean and variance. For both models, the prediction error increases with the predicted uncertainty, but the slope of GPR-MGK is larger and closer to the "ideal" MAE than D-MPNN MVE, which indicates that the predicted uncertainty of GPR-MGK is more reliable.

The NLL and the miscalibration area are summarized in Table 8, and the relationship between prediction errors and predicted uncertainties of other data sets is shown in Figures S16-S21. Among them, GPR-MGK outperforms in the ESOL, FreeSolv, lipophilicity, and PDBbind-R data sets. D-MPNN MVE outperforms in the PDBbind-C, PDBbind-F, and QM7 data sets. The problem of D-MPNN MVE is that the slope of the true prediction error relative to the predicted uncertainty is smaller, which results in the predicted uncertainty underestimate the error in the low-value range and overestimate the error in the high-value range.

Unlike the case of predictive accuracy evaluation, we think that data quality plays an important role in the comparison of UQ methods. If the noise in the data is too large, it is difficult to judge whether the prediction error is caused by

the model or the noise, which may lead to contradictive results. The prediction R^2 of different data sets is summarized in Table S5. The ESOL and the FreeSolv data sets are the least noisy because the R^2 is larger than 0.9. For the PDBbind data set, R2 is lower than 0.5, indicating that the data noise is too high. For the QM7 dataset, the NLL is too high because we use the two-dimensional graph converted by SMILES as the model input, and the three-dimensional coordinate information is ignored. According to Tang and de Jone's work, using graphs with three-dimensional coordinates as input can improve prediction performance by about one order of magnitude and provide reliable UQ.[29] Therefore, we conclude that the UQ of GPR-MGK is better than D-MPNN MVE.

The advantages of GPR-MGK are its accuracy and its computational efficiency in small data sets. The advantage of MPNN MVE is its computational efficiency for large data sets. For example, Graff et al. used MPNN UQ as a surrogate model to perform high-throughput virtual screening on a data set containing 100M molecules through active learning,[40] which is an impossible mission for GPR-MGK. On the other hand, MPNN UQ needs to retrain the neural network for each step of active learning, which is expensive. Therefore, a batch of samples must be added in each step of active learning, which limits its performance. However, GPR-MGK allows active learning by adding samples one by one.[29]

V. CONCLUSIONS

In this article, we proposed a state-of-the-art hybrid kernel for molecular property prediction. It consists of (1) the marginalized graph kernel with additive node, edge features, and inhomogeneous starting probabilities operating on the molecular graph, and (2) radial basis function kernels operating on RDKit features. Using D-MPNN as a comparison, we benchmarked the GP-MGK on 11 public data sets.

For prediction performance, GP-MGK is at the same level as that of D-MPNN. In addition, by comparing the predictions on a molecule-by-molecule basis, a correlation between the predictions of GPR-MGK and D-MPNN was observed. In addition, the ensemble prediction of GPR-MGK and D-MPNN is more accurate than each of them.

For uncertainty quantification, we demonstrate GPR-MGK outperforms D-MPNN MVE. In practical applications, reliable prediction uncertainty is very important when predicting new compounds with unknown properties.

Although the performances of GP-MGK and D-MPNN are close under the condition of optimal hyperparameters, the computational cost of finding the optimal hyperparameters of GP-MGK is still expensive. Therefore, an efficient algorithm to find the optimal hyperparameters of the graph kernel is needed.

Finally, we guide the application of GP-MGK and D-MPNN. The advantage of D-MPNN lies in computational efficiency, so it is suitable for property

prediction tasks and active learning for large-scale (million) data sets. The advantage of the GP-MGK is its nature of Bayesian inference, which can be applied to small-scale data sets(less than 50k) for uncertainty qualification[30], active learning,[29], and data noisy detection.[46]

Acknowledgments

The computations in this paper were run on the π 2.0 cluster supported by the Center for High-Performance Computing at Shanghai Jiao Tong University. This work was funded by the National Natural Science Foundation of China [Grant No. 21473112], [Grant No. 21403138], [Grant No. 21673138]. We thank Connor Coley for his help in performing uncertainty quantification of D-MPNN.

References

(1) Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph Neural Networks: A Review of Methods and Applications. *AI Open* **2020**, *1*, 57–81. https://doi.org/10.1016/j.aiopen.2021.01.001.

(2) Wu, Z.; Ramsundar, B.; N. Feinberg, E.; Gomes, J.; Geniesse, C.; S. Pappu, A.; Leswing, K.; Pande, V. MoleculeNet: A Benchmark for Molecular Machine Learning. *Chem. Sci.* **2018**, *9* (2), 513–530. https://doi.org/10.1039/C7SC02664A.

(3) Schütt, K. T.; Kindermans, P.-J.; Sauceda, H. E.; Chmiela, S.; Tkatchenko, A.; Müller, K.-R. SchNet: A Continuous-Filter Convolutional Neural Network for Modeling Quantum Interactions. *ArXiv170608566 Phys. Stat* **2017**.

(4) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural Message Passing for Quantum Chemistry. In *Proceedings of the 34th International Conference on Machine Learning*; PMLR, 2017; pp 1263–1272.

(5) Klicpera, J.; Groß, J.; Günnemann, S. Directional Message Passing for Molecular Graphs. *ArXiv200303123 Phys. Stat* **2020**.

(6) Liu, Z.; Lin, L.; Jia, Q.; Cheng, Z.; Jiang, Y.; Guo, Y.; Ma, J. Transferable Multilevel Attention Neural Network for Accurate Prediction of Quantum Chemistry Properties via Multitask Learning. *J. Chem. Inf. Model.* **2021**, *61* (3), 1066–1082. https://doi.org/10.1021/acs.jcim.0c01224.

(7) Faber, F. A.; Hutchison, L.; Huang, B.; Gilmer, J.; Schoenholz, S. S.; Dahl, G. E.; Vinyals, O.; Kearnes, S.; Riley, P. F.; von Lilienfeld, O. A. Prediction Errors of Molecular Machine Learning Models Lower than Hybrid DFT Error. *J. Chem. Theory Comput.* **2017**, *13* (11), 5255–5264. https://doi.org/10.1021/acs.jctc.7b00577.

(8) Anderson, B.; Hy, T.-S.; Kondor, R. Cormorant: Covariant Molecular Neural Networks. *ArXiv190604015 Phys. Stat* **2019**.

(9) Lu, C.; Liu, Q.; Wang, C.; Huang, Z.; Lin, P.; He, L. Molecular Property Prediction: A Multilevel Quantum Interactions Modeling Perspective. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33* (01), 1052–1060. https://doi.org/10.1609/aaai.v33i01.33011052.

(10) 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. https://doi.org/10.1021/acs.jcim.9b00410.

(11) Withnall, M.; Lindelöf, E.; Engkvist, O.; Chen, H. Building Attention and Edge Message Passing Neural Networks for Bioactivity and Physical–Chemical Property Prediction. *J. Cheminformatics* **2020**, *12* (1), 1. https://doi.org/10.1186/s13321-019-0407-y.

(12) Flam-Shepherd, D.; Wu, T. C.; Friederich, P.; Aspuru-Guzik, A. Neural Message Passing on High Order Paths. *Mach. Learn. Sci. Technol.* **2021**.

https://doi.org/10.1088/2632-2153/abf5b8.
(13) Jørgensen, P. B.; Jacobsen, K. W.; Schmidt, M. N. Neural Message Passing with Edge Updates for Predicting Properties of Molecules and Materials. *ArXiv180603146 Cs Stat* **2018**.
(14) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; Palmer, A.; Settels, V.; Jaakkola, T.; Jensen, K.; Barzilay, R. Analyzing Learned Molecular Representations for Property Prediction. *J. Chem. Inf. Model.* **2019**, *59* (8), 3370–3388. https://doi.org/10.1021/acs.jcim.9b00237.
(15) Loukas, A. How Hard Is to Distinguish Graphs with Graph Neural Networks? In *[Advances in Neural Information Processing Systems 33 (NIPS 2020)]*; 2020.
(16) Hu, W.; Fey, M.; Ren, H.; Nakata, M.; Dong, Y.; Leskovec, J. Ogb-Lsc: A Large-Scale Challenge for Machine Learning on Graphs. *ArXiv Prepr. ArXiv210309430* **2021**.
(17) Kriege, N. M.; Johansson, F. D.; Morris, C. A Survey on Graph Kernels. *Appl. Netw. Sci.* **2020**, *5* (1), 1–42. https://doi.org/10.1007/s41109-019-0195-3.
(18) Kashima, H.; Tsuda, K.; Inokuchi, A. Marginalized Kernels Between Labeled Graphs. In *Proceedings of the 20th International Conference on Machine Learning*; ICML '03; Washington D.C, USA, 2003; pp 321–328.
(19) Gärtner, T.; Flach, P.; Wrobel, S. On Graph Kernels: Hardness Results and Efficient Alternatives. In *Learning Theory and Kernel Machines*; Schölkopf, B., Warmuth, M. K., Eds.; Lecture Notes in Computer Science; Springer: Berlin, Heidelberg, 2003; pp 129–143. https://doi.org/10.1007/978-3-540-45167-9_11.
(20) Mahé, P.; Ueda, N.; Akutsu, T.; Perret, J.-L.; Vert, J.-P. Extensions of Marginalized Graph Kernels. In *Proceedings of the 21nd international conference on Machine learning*; ICML '04; ACM Press: Banff, Alberta, Canada, 2004; p 70. https://doi.org/10.1145/1015330.1015446.
(21) Fröhlich, H.; Wegner, J. K.; Sieker, F.; Zell, A. Optimal Assignment Kernels for Attributed Molecular Graphs. In *Proceedings of the 22nd international conference on Machine learning*; ICML '05; ACM Press: New York, NY, USA, 2005; pp 225–232. https://doi.org/10.1145/1102351.1102380.
(22) Kriege, N.; Mutzel, P. Subgraph Matching Kernels for Attributed Graphs. *ArXiv12066483 Cs Stat* **2012**.
(23) Morris, C.; Kriege, N. M.; Kersting, K.; Mutzel, P. Faster Kernels for Graphs with Continuous Attributes via Hashing. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*; 2016; pp 1095–1100. https://doi.org/10.1109/ICDM.2016.0142.
(24) Togninalli, M.; Ghisu, E.; Llinares-López, F.; Rieck, B.; Borgwardt, K. Wasserstein Weisfeiler-Lehman Graph Kernels. *ArXiv190601277 Cs Q-Bio Stat* **2019**.

(25) Schulz, T. H.; Horváth, T.; Welke, P.; Wrobel, S. A Generalized Weisfeiler-Lehman Graph Kernel. *ArXiv210108104 Cs* **2021**.
(26) Wu, L.; Yen, I. E.-H.; Zhang, Z.; Xu, K.; Zhao, L.; Peng, X.; Xia, Y.; Aggarwal, C. Scalable Global Alignment Graph Kernel Using Random Features: From Node Embedding to Graph Embedding. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*; KDD '19; Association for Computing Machinery: New York, NY, USA, 2019; pp 1418–1428. https://doi.org/10.1145/3292500.3330918.
(27) Tang, Y.-H.; Selvitopi, R.; Popovici, D. A. T.; USDOE. *GraphDot v0.1*; Lawrence Berkeley National Lab. (LBNL), Berkeley, CA (United States), 2019. https://doi.org/10.11578/dc.20191015.4.
(28) Tang, Y.-H.; Selvitopi, O.; Popovici, D. T.; Buluç, A. A High-Throughput Solver for Marginalized Graph Kernels on GPU. In *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*; 2020; pp 728–738. https://doi.org/10.1109/IPDPS47924.2020.00080.
(29) Tang, Y.-H.; de Jong, W. A. Prediction of Atomization Energy Using Graph Kernel and Active Learning. *J. Chem. Phys.* **2019**, *150* (4), 044107. https://doi.org/10.1063/1.5078640.
(30) Xiang, Y.; Tang, Y.-H.; Liu, H.; Lin, G.; Sun, H. Predicting Single-Substance Phase Diagrams: A Kernel Approach on Graph Representations of Molecules. *J. Phys. Chem. A* **2021**, *125* (20), 4488–4497. https://doi.org/10.1021/acs.jpca.1c02391.
(31) Haussler, D. *Convolution Kernels on Discrete Structures*; Technical report, Department of Computer Science, University of California at Santa Cruz, 1999.
(32) Rasmussen, C. E.; Williams, C. K. I. *Gaussian Processes for Machine Learning*; The MIT Press, 2006.
(33) Nair, V.; Hinton, G. E. Rectified Linear Units Improve Restricted Boltzmann Machines; 2010.
(34) Graph Kernel Machines for Molecular Property Prediction. https://github.com/Xiangyan93/Chem-Graph-Kernel-Machine (accessed 2021 -05 -30).
(35) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
(36) Descriptor computation(chemistry) and (optional) storage for machine learning. https://github.com/bp-kelley/descriptastorus (accessed 2021 -05 -30).
(37) Distributed Asynchronous Hyperparameter Optimization in Python. https://github.com/hyperopt/hyperopt (accessed 2021 -05 -30).
(38) Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for Hyper-Parameter Optimization; Neural Information Processing Systems

Foundation, 2011; Vol. 24.
(39) Bergstra, J.; Yamins, D.; Cox, D. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In *International Conference on Machine Learning*; PMLR, 2013; pp 115–123.
(40) Graff, D. E.; Shakhnovich, E. I.; Coley, C. W. Accelerating High-Throughput Virtual Screening through Molecular Pool-Based Active Learning. *Chem. Sci.* **2021**, *12* (22), 7866–7881. https://doi.org/10.1039/D0SC06805E.
(41) Dwivedi, V. P.; Joshi, C. K.; Laurent, T.; Bengio, Y.; Bresson, X. Benchmarking Graph Neural Networks. *ArXiv200300982 Cs Stat* **2020**.
(42) Wold, S.; Esbensen, K.; Geladi, P. Principal Component Analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2* (1), 37–52. https://doi.org/10.1016/0169-7439(87)80084-9.
(43) Halko, N.; Martinsson, P. G.; Tropp, J. A. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. *SIAM Rev.* **2011**, *53* (2), 217–288. https://doi.org/10.1137/090771806.
(44) Schölkopf, B.; Smola, A.; Müller, K.-R. Kernel Principal Component Analysis. In *Artificial Neural Networks — ICANN'97*; Gerstner, W., Germond, A., Hasler, M., Nicoud, J.-D., Eds.; Lecture Notes in Computer Science; Springer: Berlin, Heidelberg, 1997; pp 583–588. https://doi.org/10.1007/BFb0020217.
(45) Hirschfeld, L.; Swanson, K.; Yang, K.; Barzilay, R.; Coley, C. W. Uncertainty Quantification Using Neural Networks for Molecular Property Prediction. *J. Chem. Inf. Model.* **2020**, *60* (8), 3770–3780. https://doi.org/10.1021/acs.jcim.0c00502.
(46) Tang, Y.-H.; Zhu, Y.; de Jong, W. A. Detecting Label Noise via Leave-One-Out Cross-Validation. *ArXiv210311352 Cs Math Stat* **2021**.

Table 1. Atom Features for Marginalized Graph Kernel.

feature	description	size
AN	atomic number	1
AN_1_list	atomic number for 1st layer heavy neighbors	variable
AN_2_list	atomic number for 2nd layer heavy neighbors	variable
AN_3_list	atomic number for 3th layer heavy neighbors	variable
AN_4_list	atomic number for 4th layer heavy neighbors	variable
AN_1_count	number of heavy atoms in 1st layer neighbors	1
AN_2_count	number of heavy atoms in 2nd layer neighbors	1
Hcount	number of bonded hydrogens	1
MorganHash	Morgan substructure at radius=3	1
ringSize_list	the ring size of all distinct rings	variable
ring_count	the number of distinct rings	1
chirality	unspecified, tetrahedral CW/CCW, or achiral	1

Table 2. Bond Features for Marginalized Graph Kernel.

feature	description	size
bond type	bond order, single, double, triple, or aromatic	1
stereo	none/E/Z for double bond	1
ring-stereo	none/E/Z for single bond in a ring	1
conjugated	whether the bond is conjugated	1

Table 3. Atom Features for D-MPNN[ab]

feature	description	size
atom type	type of atom (ex. C, N, O), by atomic number	100
# bonds	number of bonds the atom is involved in	6
formal charge	integer electronic charge assigned to atom	5
chirality	unspecified, tetrahedral CW/CCW, or other	4
# Hs	number of bonded hydrogen atoms	5
hybridization	sp, sp2, sp3, sp3d, or sp3d2	5
aromaticity	whether this atom is part of an aromatic system	1
atomic mass	mass of the atom, divided by 100	1

[a]All features are one-hot encodings except for atomic mass, which is a real number scaled to be on the same order of magnitude.

[b]This table is the same as Table 1 in Yang et al.'s paper.[2]

Table 4. Bond Features for D-MPNN[ab]

feature	description	size
bond type	single, double, triple, or aromatic	4
conjugated	whether the bond is conjugated	1
in ring	whether the bond is part of a ring	1
stereo	none, any, E/Z or cis/trans	6

[a]All features are one-hot encodings.

[b]This table is the same as Table 2 in Yang et al.'s paper.[2]

Table 5. Data sets Used in This Paper

data set	task		compounds	metric
ESOL	regression	1	1128	RMSE
FreeSolv	regression	1	642	RMSE
Lipophilicity	regression	1	4200	RMSE
PDBbind-C	regression	1	168	RMSE
PDBbind-R	regression	1	3040	RMSE
PDBbind-F	regression	1	9880	RMSE
QM7	regression	1	6830	MAE
BACE	classification	1	1513	ROC-AUC
BBBP	classification	1	2039	ROC-AUC
SIDER	classification	27	1427	ROC-AUC
ClinTox	classification	2	1478	ROC-AUC

Table 6. Prediction Results of GPR-MGK, D-MPNN, and their Ensembling Model

data set	ESOL	FreeSolv	Lipophilicity	PDBbind-C	PDBbind-R	PDBbind-F	QM7
			random split				
GPR-MGK	0.547±0.050	0.822±0.173	0.595±0.037	1.940±0.289	1.302±0.049	1.284±0.026	53.22±3.12
D-MPNN Optimized	0.570±0.054	0.904±0.184	0.551±0.044	1.849±0.236	1.324±0.052	1.279±0.030	59.71±3.40
D-MPNN Ensemble	0.557±0.051	0.882±0.175	0.539±0.046	1.853±0.232	1.297±0.048	1.261±0.029	57.06±3.34
Ensemble[a]	**0.537±0.049**	**0.817±0.167**	**0.534±0.041**	**1.812±0.239**	**1.273±0.046**	**1.244±0.026**	**50.29±3.13**
			scaffold split				
GPR-MGK	0.789±0.090	1.789±0.605	0.641±0.041	2.005±0.282	1.408±0.067	1.352±0.042	**66.90±9.62**
D-MPNN Optimized	0.822±0.090	1.782±0.591	0.603±0.056	1.901±0.271	1.417±0.074	1.334±0.050	83.98±10.32
D-MPNN Ensemble	0.793±0.079	1.729±0.580	0.589±0.051	1.892±0.281	1.390±0.069	1.315±0.039	79.46±10.11
Ensemble[a]	**0.772±0.081**	**1.703±0.599**	**0.580±0.044**	**1.851±0.252**	**1.371±0.066**	**1.302±0.040**	69.31±9.45

[a]Ensemble prediction of GPR-MGK and D-MPNN Ensemble.

Table 7. Prediction Results of GPC-MGK, D-MPNN, and their Ensembling Model

data set	BACE	BBBP	SIDER	ClinTox
random split				
GPC-MGK	0.883±0.028	0.921±0.023	0.658±0.023	0.774±0.081
D-MPNN Optimized	0.893±0.026	0.924±0.021	0.655±0.026	0.900±0.049
D-MPNN Ensemble	0.899±0.024	0.927±0.021	0.664±0.026	**0.907±0.044**
Ensemble[a]	**0.901±0.024**	**0.931±0.021**	**0.671±0.025**	0.872±0.053
scaffold split				
GPC-MGK	0.858±0.044	0.907±0.030	0.623±0.023	0.814±0.062
D-MPNN Optimized	0.858±0.042	0.911±0.030	0.634±0.030	0.888±0.042
D-MPNN Ensemble	0.864±0.043	0.915±0.026	0.638±0.023	**0.897±0.039**
Ensemble[a]	**0.870±0.042**	**0.920±0.027**	**0.650±0.031**	0.870±0.058

[a]Ensemble prediction of GPR-MGK and D-MPNN Ensemble.

Table 8. Uncertainty Quantification of GPR-MGK, D-MPNN MVE

data set	ESOL	FreeSolv	Lipophilicity	PDBbind-C	PDBbind-R	PDBbind-F	QM7
			GPR-MGK				
NLL	**0.824**	**1.037**	0.900	2.380	**1.699**	1.690	156.4
Miscalibration Area	**0.018**	**0.026**	**0.007**	0.014	**0.003**	0.006	0.049
			D-MPNN MVE				
NLL	0.877	1.286	**0.836**	**2.137**	1.725	**1.661**	**80.27**
Miscalibration Area	0.032	0.028	0.014	**0.010**	0.006	**0.003**	**0.029**

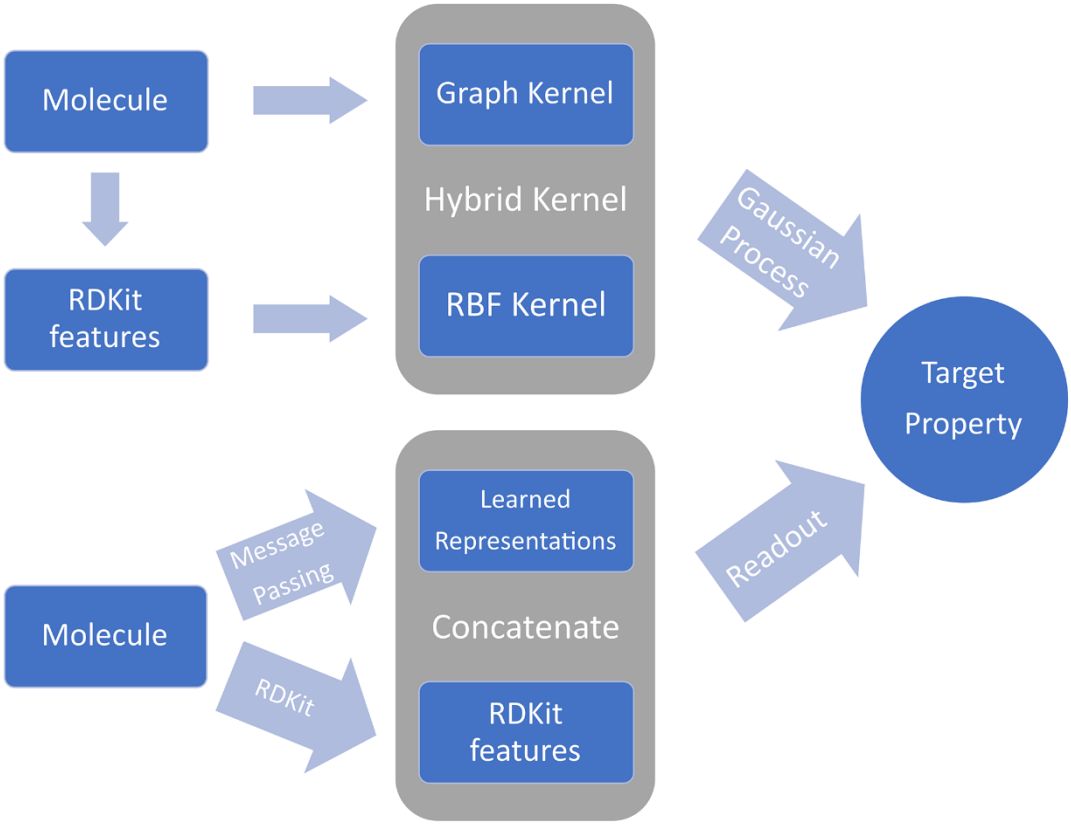


Figure 1. Overviews of machine learning models. Top: In GP-MGK, the marginalized graph kernel with the molecular graph as the input and the RBF kernel with the RDKit features as the input are hybridized, followed by Gaussian process regression or classification. Bottom: In D-MPNN, the learned molecular representations using message passing are concatenated with RDKit features, followed by a feed-forward neural network.

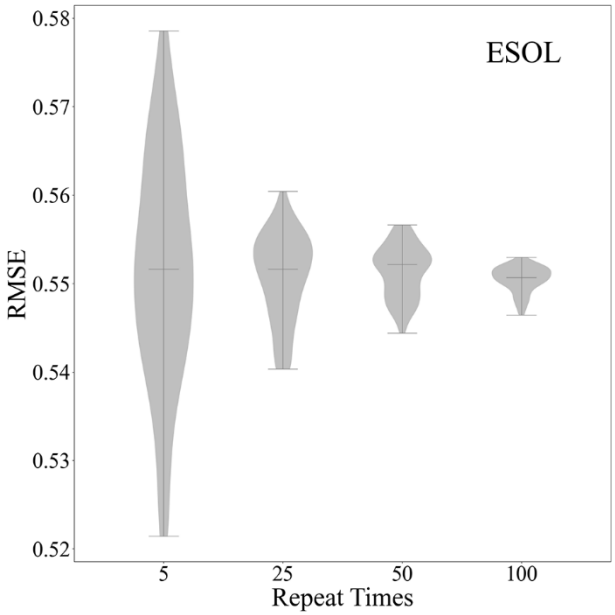


Figure 2. Performance evaluation of GPR-MGK on the ESOL data set with different repetition times. Each column corresponds to the distribution of 100 evaluations. For each evaluation, the data is randomly divided into a training set and a test set at a ratio of 80:20.

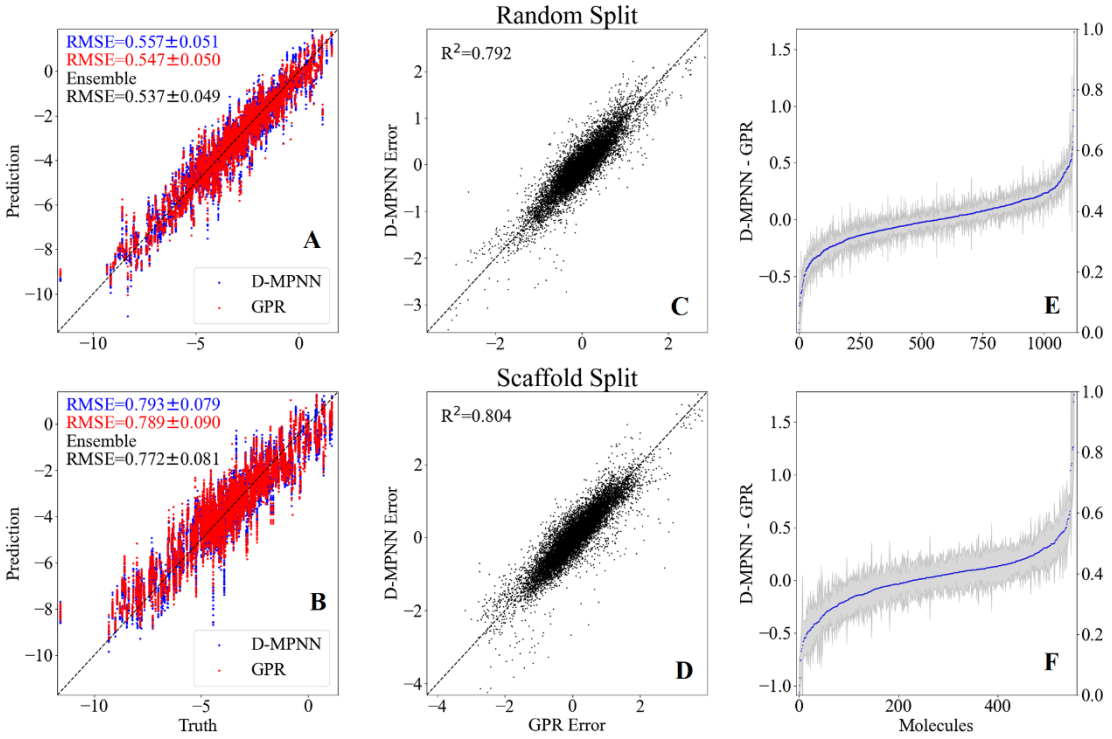


Figure 3. A comparison between GPR-MGK and D-MPNN Ensembles. Top: Random split. Bottom: Scaffold split. (A, B) The prediction on the test set using GPR-MGK (red) and D-MPNN Ensemble (blue) are compared. (C, D) The relationship between GPR-MGK error and D-MPNN Ensemble error. (E, F) The prediction differences between GPR-MGK and D-MPNN Ensemble are sorted by molecule ID. The gray region is the standard deviation obtained by making predictions based on different training sets.

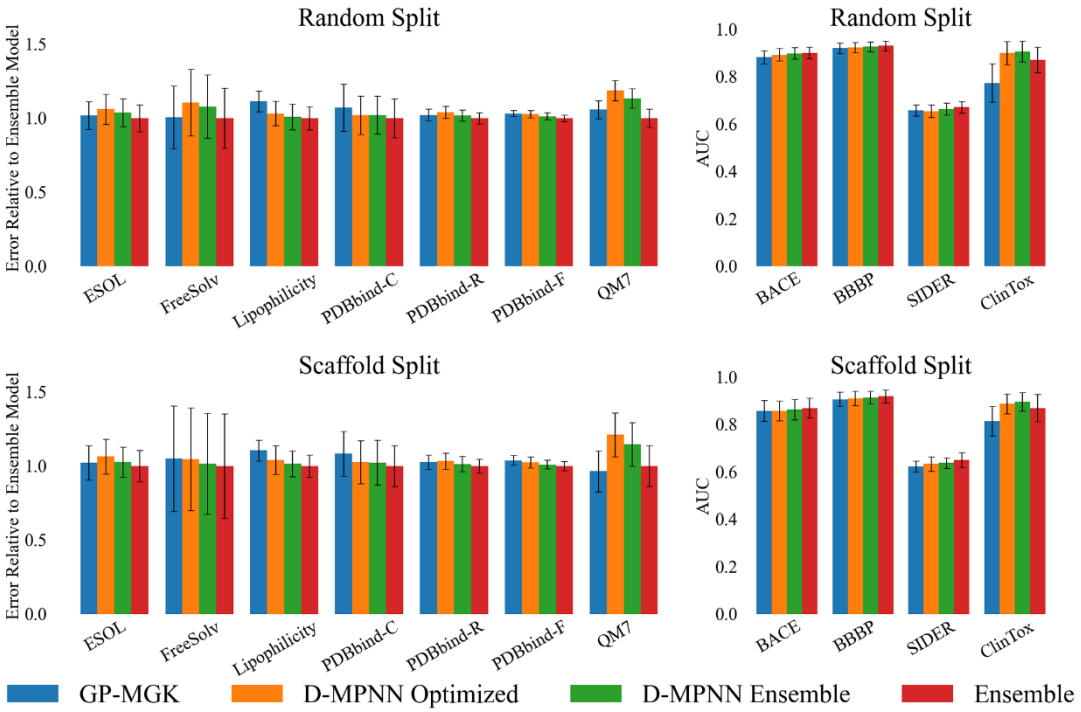


Figure 4. Comparisons of graph kernel models against direct message passing neural networks. Top: Random data split. Bottom: Scaffold data split. Left: Regression data sets. Right: Classification data sets.

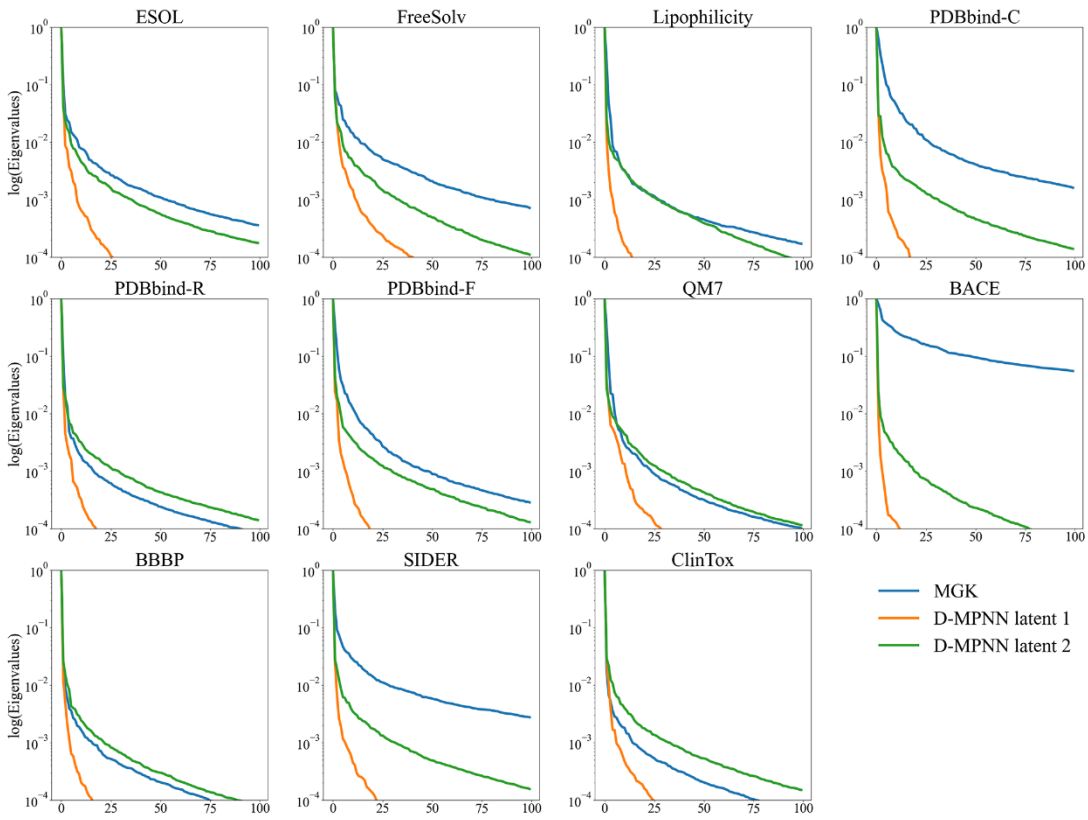


Figure 5. Eigenvalues associated with the first 100 principal components of the latent representations of MGK, D-MPNN latent 1, and D-MPNN latent 2. All eigenvalues are normalized by the leading one. The eigenvalue spectra indicate the effective number of extracted features.

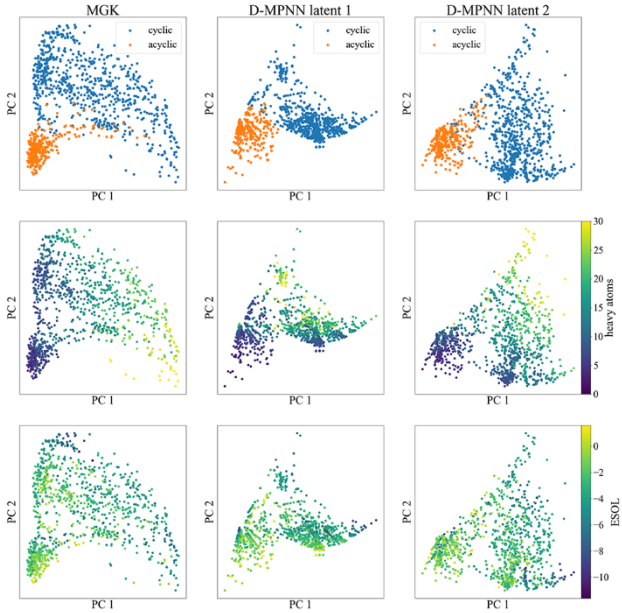


Figure 6. First and second principal components of MGK, D-MPNN latent 1 and D-MPNN latent 2 representations on the ESOL data set. Top: cyclic and acyclic molecules. Middle: the number of heavy atoms. Bottom: log solubility.

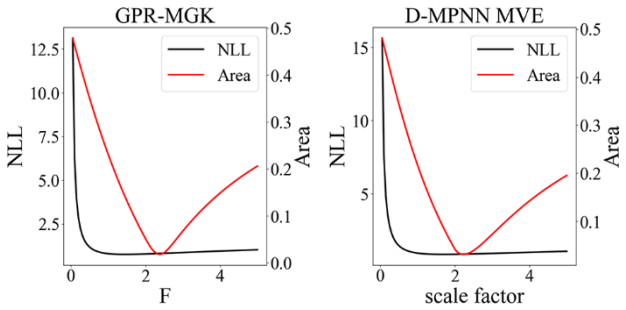


Figure 7. Relationship between the negative log-likelihood (NLL, black), the miscalibration area (red), and the scale factor of predicted uncertainty. ESOL data set.

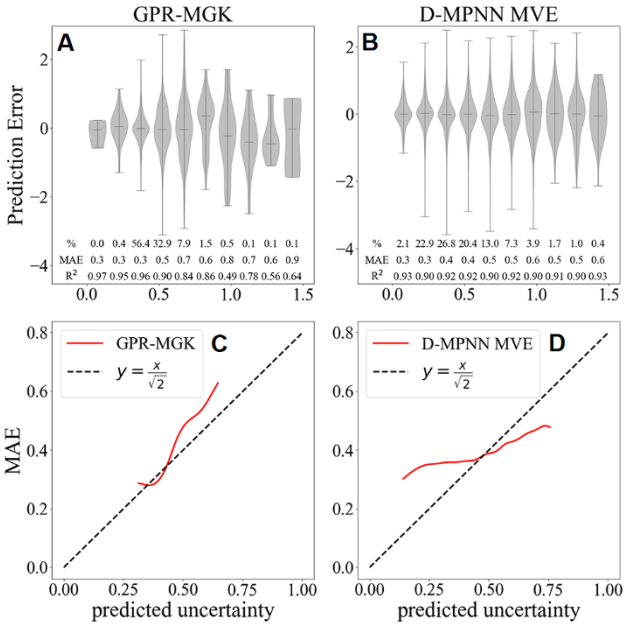


Figure 8. Relationship between the prediction error and the predicted uncertainty using (A) GPR-MGK and (B) D-MPNN MVE. Relationship between the MAE and the predicted uncertainty using (C) GPR-MGK and (D) D-MPNN MVE. ESOL data set.