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]*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

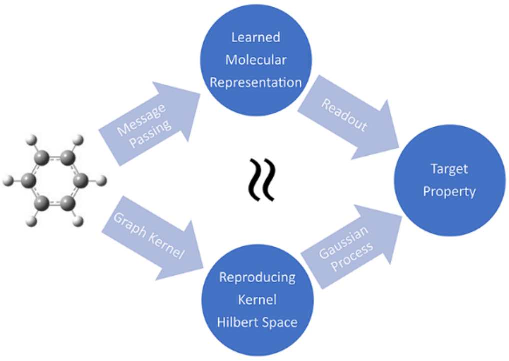
* Corresponding Authors

Tel: +86 136 1180 2895

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

Abstract

This work presents a state-of-the-art hybrid kernel for molecular property predictions. The hybrid kernel consists of a marginalized graph kernel that operates on molecular graphs and radial basis function kernels that operate on global molecular features. Direct message passing neural network (D-MPNN) with global molecular features is used as strong baselines. After using Bayesian optimization to find the optimal hyperparameters, we benchmark the models on 11 publicly available data sets. Our results show that the prediction of the graph kernel is correlated to the prediction of D-MPNN, which indicates that the molecular representation learned from D-MPNN is very close to the reproducing kernel Hilbert space generated by the hybrid kernel. These results may provide clues for research on the interpretability of graph neural networks. In addition, ensembling the graph kernel models with D-MPNN is the best. The advantage of D-MPNN lies in computational efficiency, and the advantage of the graph kernel model lies in the inherent uncertainty qualification of Gaussian process regression.



I. INTRODUCTION

Molecular property prediction is one of the central and classical research topics of cheminformatics, which has attracted widespread attention for decades. Recently, this field is rejuvenated due to the advances in deep learning. Graph neural networks (GNNs) result in state-of-the-art predictions on quantum mechanical properties, physicochemical properties, biological activity and toxicity.[1–11]

To fairly evaluate the quality of different methods, Wu et al. introduced MoleculeNet as a large-scale benchmark for molecular property prediction.[12] It provides multiple public data sets, data splitting, as well as high-quality implementation of popular algorithms of molecular featurization and learning algorithms. They compared six featurization methods, eight conventional models, and six graph-based models in eight regression and nine classification tasks, demonstrating that GNNs outperform molecular fingerprints methods in most cases. Yang et al. proved that a mixed molecular representation that combines GNNs and expert-crafted descriptors is state-of-the-art.[13] They performed an extensive comparison on 19 public and 16 proprietary data sets.

Graph kernel is another research branch of graph-based machine learning methods.[14–23] Compared with GNN, it has received less attention due to the expensive computational cost and programming difficulty. Recently, Tang et al. developed the GraphDot software package,[24] which uses GPUs to efficiently compute marginalized graph kernels (MGK).[25] Using GraphDot, Tang and de Jong introduced an MGK for molecular atomization energy prediction using the QM7 data sets.[26] Xiang et al. developed normalized marginalized graph kernels (nMGK) for molecules and constructed accurate prediction models for various thermodynamic and transport properties of pure substances.[27]

In this paper, we aim to benchmark the marginalized graph kernels using the direct message passing neural network (D-MPNN)[13] as a strong baseline. For a fair comparison, we optimized the hyperparameters of the two methods. For D-MPNN, we follow the setting in the article by Yang et al.[13] For graph kernel methods, we (1) introduce a new kernel architecture that allows features and atoms with different weights. (2) hybrid the graph kernel and the radial basis function (RBF) kernels, where the RBF kernels operate on the global molecular features. (3) optimize the hyperparameters using Bayesian optimization. We compared our graph kernel model and D-MPNN on 11 publicly available data sets.

II. METHODS

Normalized Marginalized Graph Kernel Methods

The overview of graph kernel models is sketched on the top of Figure 1. In MGK, molecules are represented by undirected labeled graphs, where atoms represent vertices, and chemical bonds represent edges. We use MGK to compute molecular similarity, which consists of five parts: atom microkernels, bond microkernels, starting probability, stop probability, and transition probability.

The atom and bond features are listed in Tables 1 and 2. For single-valued feature, the elementary kernel is Kronecker delta

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

For features with variable size, the elementary kernel is the sequence convolution of Kronecker delta

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

where

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

Here, l_1, l_2 are two features vector, and *h* is the hyperparameter.

The microkernel for atom or bond is a linear combination of elementary kernels between individual features

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

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

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

The starting probability of an atom is a linear combination of elementary probability

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

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

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 B, C, N, O, F, Si, P, S, Cl, Br, and I in practice.

The stoping probability is set to be a constant hyperparameter p_q. The transition probability is set to $1/n$ where *n* is the number of neighbors to the current atom.

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

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

Where **h** and **h**' are the random walk paths of length l.

The MGK can be normalized with weight:

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

where *F* and λ are the hyperparameters.

Gaussian processes are used for regression and classification tasks.[28] More details can be found in references.[15,25–27]

Ensemble Direct Message Passing Neural Network

The overview of D-MPNN is sketched on the bottom of Figure 1. The D-MPNN is used as baseline model in this work. Herein, we briefly introduce the model.

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

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

where $\operatorname{cat}(x_v, e_{vw})$ 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.

The message passing update equations are

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

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

where $N(v)$ are the neighbors of *v*. The learned atom hidden states are

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

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

The molecular representation is the sum of atom hidden states

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

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. More details can be found in reference.[13]

RDKit Features

Yang et al concatenated 200 global features that can be rapidly computed using RDKit with the learned molecular representation through message passing, which significantly improves the prediction performance.

To make a fair comparison between the graph kernel and D-MPNN, we also add the 200 RDKit features in graph kernel models using a hybrid kernel

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

where G, G' are the molecular graphs and F_{RDKit}, F'_{RDKit} are RDKit features. K_G is the normalized marginalized graph kernel described above and K_F is the radial basis function kernel $K_F(F_{\mathrm{RDKit}}, F'_{\mathrm{RDKit}}) = \exp\left(-\frac{\|F_{\mathrm{RDKit}} - F'_{\mathrm{RDKit}}\|^2}{2\sigma^2}\right)$.

Hyperparameter Optimization

There are tens of hyperparameters for graph kernel models and four hyperparameters for D-MPNN. In order to maximize the performance of both models, we use Tree of Parzen Estimators (TPE) to optimize hyperparameters.[29,30]

Implementation

All code for the graph kernel models is available in our GitHub repository.[31] We use the GraphDot python package to compute the marginalized graph kernels and perform Gaussian process regression.[24] We use the scikit-learn package to carry out Gaussian process classification.[32] We use the Descriptatorus package[33] to calculate the RDKit features and HyperOpt package[34] to hyperparameters optimization.

III. EXPERIMENTS

Data sets

The publicly available data sets used in this study are listed in Table 5. These data sets are popularly used for benchmark researches in molecular property prediction.[12,13] Mean absolute error (MAE), root mean square error (RMSE), and area under the receiver operating characteristic curve (ROC-AUC) are used as metrics.

Hyperparameters Optimization

For D-MPNN, we follow the setting of Yang et al, "For each data set, we use 20 iterations of Bayesian optimization on 10 randomly seeded 80:10:10 data splits to determine the best hyperparameters, selecting hyperparameters based on validation set performance".[13] The optimal hyperparameters are listed in Tables S1 and S2.

For Gaussian process regression, 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. For Gaussian process classification, we use 100 iterations of Bayesian optimization on 10 randomly seeded 80:20 data splits to determine the best hyperparameters with the best performance on test sets. 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-balanced data splits. The data were divided into the training, validation, and test set according to the ratio of 80:10:10. For D-MPNN, it was trained for 50 epochs, and the model with the highest performance on the validation set was used as the final model to make predictions on the test set. For the graph kernel models, 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.

IV. RESULTS AND DISCUSSION

We only compare the graph kernel models and D-MPNN with optimal performances. In this section, “GPR-MGK”, “GPC-MGK” refers to graph kernel models with RDKit features and optimized hyperparameters. The term "normalization" is no longer used because whether it is used or not is also an adjustable hyperparameter. D-MPNN-OPT refers to D-MPNN with RDKit features and optimized hyperparameters. D-MPNN-OPT-E5 refers to a model that ensembling five D-MPNN-OPT models. Ensemble refers to a model that ensembling GPR-MGK (GPC-MGK for classification) and D-MPNN-OPT-E5.

Benchmark on Same Data Splits

We emphasize that it is very important to compare different models on the same training and test sets, otherwise, you may get contradictory results due to random noise. We performed the GPR-MGK model on the ESOL data set to illustrate this point. In Figure 2, the RMSE of the test set is plotted as a function of the number of data splits. Each string is the statistical result of 100 repeated runs with different random seeds. The difference between the best and worst results could be 0.06, 0.02, and 0.01 for repeating times of 5, 25, and 50. Therefore, we use the same data split in this work to compare the graph kernel model and D-MPNN to get reliable results. All data splits are repeated 100 times. Dwivedi et al. also held this viewpoint when benchmarking graph neural networks.[35]

GPR-MGK VS D-MPNN

We first compare our GPR-MGK model with D-MPNN-OPT-E5 on the

ESOL data set. In Figure 3A, B, comparisons of predictions using GPR-MGK and D-MPNN-OPT-E5 against the reference data are given, and the corresponding RMSE values are provided. The prediction performance of GPR-MGK and D-MPNN-OPT-E5 are the same, and ensembling them prediction that averaging them is better. In Figure 3C, D, the prediction error of GPR-MGK and D-MPNN-OPT-E5 are compared, and a strong correlation between them. In more detail, we draw the difference between the predictions of GPR-MGK and D-MPNN-OPT-E5 for different molecules in Figure 3E, F. The gray area represents the standard deviation of the same molecule under different data splits. The predictions of GPR-MGK and D-MPNN-OPT-E5 for most molecules are similar, except for a few molecules with larger differences. The results for other data sets are shown in Figures S1-4. The correlation of GPR-MGK and D-MPNN comes from the fact that both models treat molecules as graphs, and information flows through bonds.

All results are summarized in Tables 6, 7, and at the left of Figure 4. Compared with D-MPNN-OPT, GPR-MGK achieves better results in 5 comparisons, the same results in 5 comparisons, and poor results in 4 comparisons. Compared with D-MPNN-OPT-E5, GPR-MGK achieves better results in 3 comparisons, the same results in 4 comparisons, and poor results in 7 comparisons. We emphasize that although the predictive abilities of GPR-MGK and D-MPNN are similar, their ensemble predictions are the best in 13 comparisons, except for the GPR-MGK on the QM7 dataset with scaffold splitting. In addition, the difference between D-MPNN-OPT-E5 and D-MPNN-OPT is smaller than the difference between Ensemble and D-MPNN-OPT-E5, indicating that ensembling the D-MPNN with GPR-MGK is more effective than ensembling multiple D-MPNN.

GPC-MGK VS D-MPNN

The results of classification data sets are summarized in Tables 8, 9, and at 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, and the ensemble prediction of GPC-MGK and D-MPNN is the best. For the ClinTox dataset, D-MPNN outperforms GPC-MGK.

Uncertainty Analysis of GPR-MGK

GPR-MGK is a Bayesian inference method, and its prediction is a Gaussian distribution. Therefore, its advantage is that the predicted variance can be used to evaluate the reliability of the prediction. This is very important for the prediction of molecular properties because the existing data occupies only a very small part of the huge chemical compound space (CCS). At this stage, it is impossible to have enough data to train an ML model that can cover the entire CCS. Therefore, we need to know the range of capabilities of the ML model, and GPR-MGK is a way to achieve this.

Figure 5 shows the relationship between predicted error and posterior uncertainty on the ESOL data set. The prediction data are divided into 10 intervals according to posterior uncertainty. For each interval, the error is plotted in the form of a violin, where the horizontal bars represent the maximum, median, and minimum values, and the width represents the probability distribution. The data percentage, RMSE and R^2 are displayed below. Predictions with small posterior uncertainty are more accurate than predictions with large posterior uncertainty. The results of the other data sets are plotted in Figures S5-S10. The PDBbind-C data set contains only 168 data points, so the results are messy. The QM7 data set only contains molecules with no more than 7 heavy atoms, so the similarity between molecules is too high, resulting in too low predicted posterior uncertainty. In other data sets, there is a clear correlation between prediction error and posterior uncertainty.

V. CONCLUSIONS

In this article, we proposed a state-of-the-art hybrid kernel for molecular property prediction. It consists of (1) MGK with additive node, edge features and starting probabilities operating on molecular graph and (2) radial basis function kernel operating on RDKit features. Using D-MPNN as a strong baseline, we have demonstrated the power of the hybrid kernel through extensive comparisons of its performance on various data sets. A strong correlation between the predictions of GPR-MGK and D-MPNN is observed, indicating that the molecular representation learned through message passing is closed to the reproducing kernel Hilbert space generated by the MGK. Furthermore, a better model can be obtained by ensembling GP-MGK with D-MPNN.

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 very expensive. Therefore, an efficient algorithm to find the optimal hyperparameters of the graph kernel is needed. In the current situation, the advantage of GNNs is that the calculation is more efficient, while the advantage of the graph core model is uncertainty qualification[27] and active learning.[26]

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].

References

(1) 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**.

(2) 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.

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

(4) 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.

(5) 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.

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

(7) 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.

(8) 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.

(9) 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.

(10) 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.

(11) 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**.

(12) 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.

(13) 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.

(14) 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.

(15) 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.

(16) 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.

(17) 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.

(18) 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.

(19) Kriege, N.; Mutzel, P. Subgraph Matching Kernels for Attributed Graphs. *ArXiv12066483 Cs Stat* **2012**.

(20) 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.

(21) Togninalli, M.; Ghisu, E.; Llinares-López, F.; Rieck, B.; Borgwardt, K. Wasserstein Weisfeiler-Lehman Graph Kernels. *ArXiv190601277 Cs Q-Bio Stat* **2019**.

(22) Schulz, T. H.; Horváth, T.; Welke, P.; Wrobel, S. A Generalized Weisfeiler-Lehman Graph Kernel. *ArXiv210108104 Cs* **2021**.

(23) 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.

(24) The GraphDot Library. https://gitlab.com/yhtang/graphdot (accessed May 30, 2021).

(25) 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.

(26) 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.

(27) 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.

(28) Rasmussen, C. E.; Williams, C. K. I. *Gaussian Processes for Machine Learning*; The MIT Press, 2006.

(29) Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for Hyper-Parameter Optimization; Neural Information Processing Systems Foundation, 2011; Vol. 24.

(30) 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.

(31) Graph Kernel Machines for Molecular Property Prediction. https://github.com/Xiangyan93/Chem-Graph-Kernel-Machine (accessed May 30, 2021).

(32) 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.

(33) Descriptor computation(chemistry) and (optional) storage for machine learning. https://github.com/bp-kelley/descriptastorus (accessed May 30, 2021).

(34) Distributed Asynchronous Hyperparameter Optimization in Python. https://github.com/hyperopt/hyperopt (accessed May 30, 2021).

(35) Dwivedi, V. P.; Joshi, C. K.; Laurent, T.; Bengio, Y.; Bresson, X. Benchmarking Graph Neural Networks. *ArXiv200300982 Cs Stat* **2020**.

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[a][b]

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
QM7	regression	1	6830	MAE
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
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 Based on Random Split

Data	ESOL	FreeSolv	Lipophilicity	PDBbind-C	PDBbind-R	PDBbind-F	QM7
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-OPT	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-OPT-E5	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**

[a] Ensemble prediction of GPR-MGK and D-MPNN-OPT-E5.

Table 7. Prediction Results of GPR-MGK, D-MPNN, and their Ensembling Model Based on Scaffold Split

Data	ESOL	FreeSolv	Lipophilicity	PDBbind-C	PDBbind-R	PDBbind-F	QM7
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-OPT	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-OPT-E5	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	**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

Table 8. Prediction Results of GPC-MGK, D-MPNN, and their Ensembling Model Based on Random Split

Data	BACE	BBBP	SIDER	ClinTox
GPC-MGK	0.883±0.028	0.921±0.023	0.658±0.023	0.774±0.081
D-MPNN-OPT	0.893±0.026	0.924±0.021	0.655±0.026	0.900±0.049
D-MPNN-OPT-E5	0.899±0.024	0.927±0.021	0.664±0.026	**0.907±0.044**
Ensemble	**0.901±0.024**	**0.931±0.021**	**0.671±0.025**	0.872±0.053

Table 9. Prediction Results of GPC-MGK, D-MPNN, and their Ensembling Model Based on Scaffold Split

Data	BACE	BBBP	SIDER	ClinTox
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	**0.870±0.042**	**0.920±0.027**	**0.650±0.031**	0.870±0.058

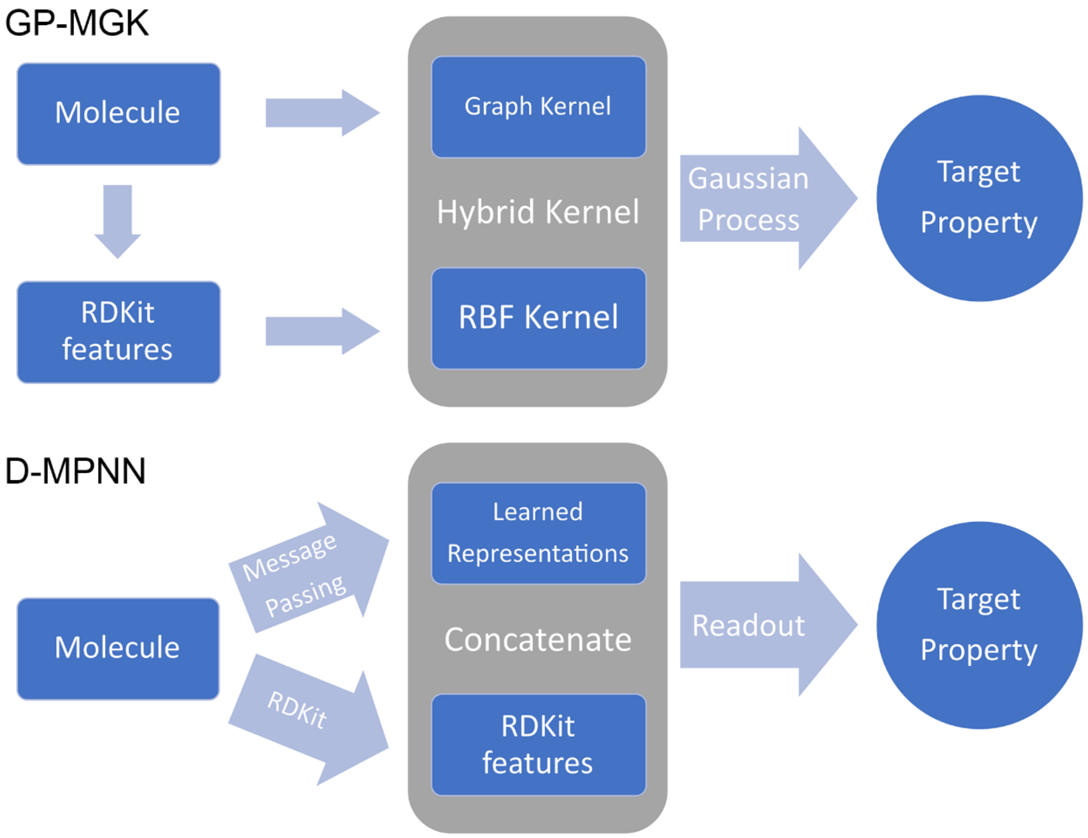


Figure 1. Overviews of machine learning models. Top: In GP-MGK, the marginalized graph kernel with the molecular graph as the input and the RBF kernel with the RDKit features as the input are hybridized, followed by Gaussian process regression or classification. Bottom: In D-MPNN, the learned molecular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

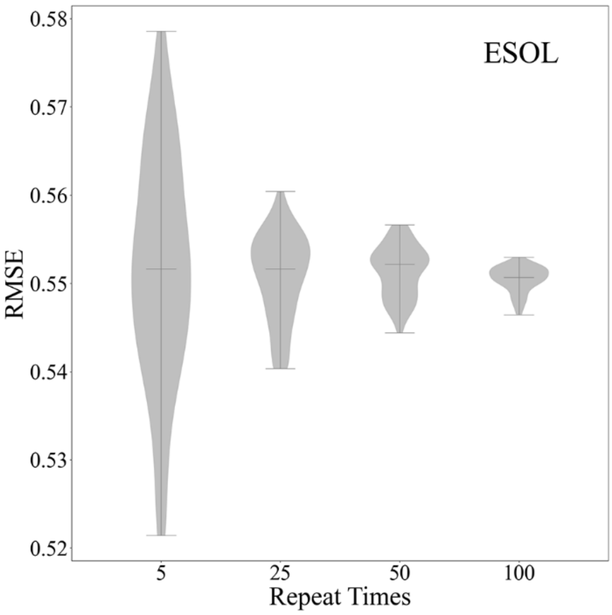


Figure 2. Performance evaluation of GPR-MGK on the ESOL data sets with different repetition times. For each column, the distribution of 100 calculations is counted. For each performance evaluation, the data is randomly divided into the training set and test set at a ratio of 80:20.

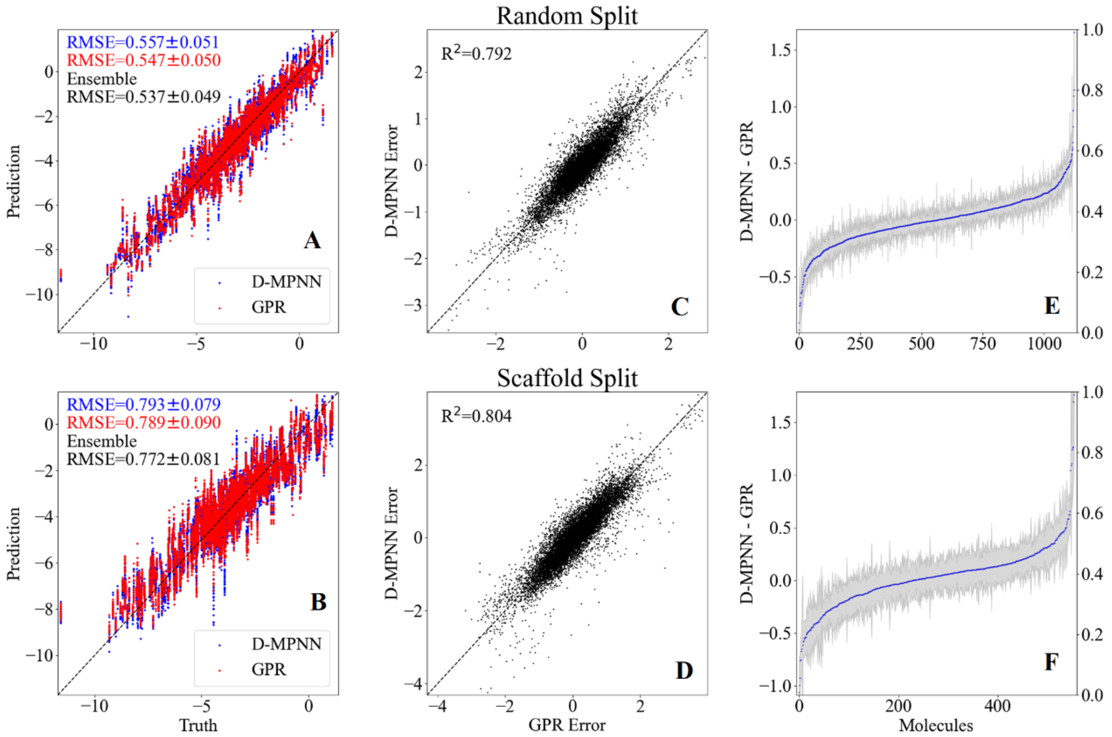


Figure 3. Comparison between GPR-MGK and D-MPNN. Top: Random split. Bottom: Scaffold split. (A, B) The prediction on the test set using GPR-MGK (red) and D-MPNN (blue) are compared. (C, D) The relationship between GPR-MGK error and D-MPNN error. (E, F) The prediction differences between GPR-MGK and D-MPNN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

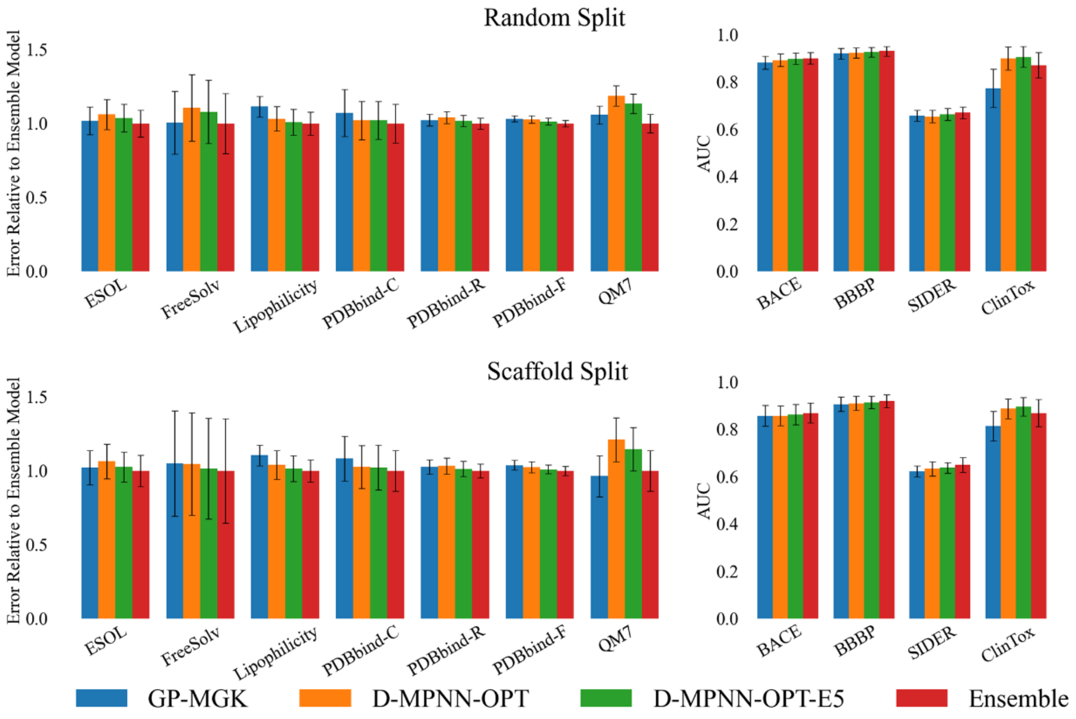


Figure 4. Comparisons of graph kernel models against direct message passing neural networks. Top: Random data split. Bottom: Scaffold data split. Left: Regression data sets. Right: classification data sets.

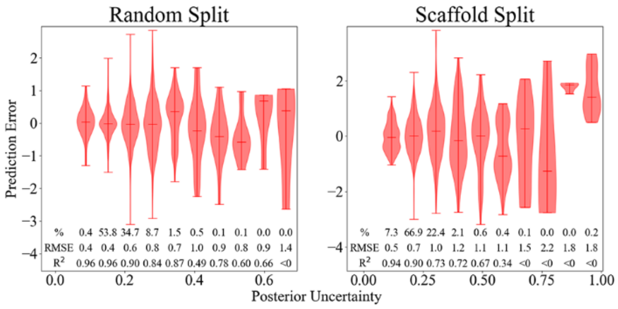


Figure 5. Relationship between predicted error and posterior uncertainty on the ESOL data set.