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

Yan Xiang, Yu-Hang Tang, Guang Lin,* and Huai Sun*

ABSTRACT: This work proposes a state-of-the-art hybrid kernel to calculate molecular similarity. Combined 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 operate on molecular graphs and global molecular features, respectively. Bayesian optimization was used to obtain the optimal hyperparameters for both models. The comparisons are performed on 11 publicly available data sets. Our results show that their performances are similar, their prediction errors are correlated, and the ensemble predictions of the two models perform better than either of them. Through principal component analysis, we found that the molecular embeddings of the hybrid kernel and the D-MPNN are also similar. The advantage of D-MPNN lies in the computational efficiency and scalability of large-scale data, while the advantage of the graph kernel models lies in the 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. 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 schemes, as well as implementations of popular algorithms of molecular featurization and learning algorithms. Their results demonstrate that the graph-based models outperform molecular fingerprint methods in most data sets. GNNs are impressively successful in predicting quantum mechanical properties, physicochemical properties, biological activity, and toxicity.3−13

The prediction quality is correlated with the amount of data. In the results of the 2021 KDD Cup Large-scale Challenge (OGB-LSC), the deep GNNs perform better for the PCQM4M-LSC data set containing about 4 million molecules.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 in which the amount of data is limited, the optimal performance is usually reached within a few message-passing steps. Yang et al. showed that a hybrid molecular representation that combines the 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.16 These findings indicate that the molecular representations learned through message passing are fundamentally localized, the ceiling of prediction quality is determined by the amount of data, and it is beneficial to introduce features that describe the molecular global features when the amount of data is insufficient.

Like GNNs, the graph kernel is a branch of graph-based machine-learning methods.17−26 Marginalized graph kernel (MGK) is a random walk graph kernel, whose fundamental principle was first proposed by Kashima et al. in 2003.18 However, it has not been thoroughly studied for a long time due to the computational cost and programming difficulty. Recently, Tang et al. developed the GraphDot package,27 which uses GPUs to accelerate the computation of MGK by several orders of magnitude and supports labeling features on the vertices and edges of the graph.28 Using GraphDot, Tang and de Jong presented an MGK for quantum mechanical property prediction using 3D molecular graph as inputs.29 More recently, Xiang et al. developed normalized marginalized graph kernels (nMGK) to predict the thermodynamic properties of pure organic liquids using 2D molecular graphs as inputs.30

Received: September 12, 2021
Published: November 1, 2021
Despite the success of GNNs and graph kernels, these two different types of graph-based machine-learning models have not been compared in detail to clarify their respective advantages and disadvantages. This article aims to address this issue. The MGK is selected because other graph kernels reported in the literature are designed primarily for graph classification tasks, and their performance in regression tasks is less well understood. D-MPNN is selected because it has been benchmarked extensively using various models on 19 public and 16 proprietary industrial data sets.

In this work, we develop a new MGK to achieve better performance based on our previous work. Gaussian process regression and classification (GP-MGK) were used as kernel machines. 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 seven regression and four classification data sets. We also provide suggestions for practical applications of GP-MGK and D-MPNN.

II. METHODS

Marginalized Graph Kernel. 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.

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:

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

(1)

Table 1. Atom Features for Marginalized Graph Kernel

<table>
<thead>
<tr>
<th>feature</th>
<th>description</th>
<th>size</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN</td>
<td>atomic number</td>
<td>1</td>
</tr>
<tr>
<td>AN_1_list</td>
<td>atomic number for first layer heavy neighbors</td>
<td>variable</td>
</tr>
<tr>
<td>AN_2_list</td>
<td>atomic number for second layer heavy neighbors</td>
<td>variable</td>
</tr>
<tr>
<td>AN_3_list</td>
<td>atomic number for third layer heavy neighbors</td>
<td>variable</td>
</tr>
<tr>
<td>AN_4_list</td>
<td>atomic number for fourth layer heavy neighbors</td>
<td>variable</td>
</tr>
<tr>
<td>AN_1_count</td>
<td>number of heavy atoms in first layer neighbors</td>
<td>1</td>
</tr>
<tr>
<td>AN_2_count</td>
<td>number of heavy atoms in second layer neighbors</td>
<td>1</td>
</tr>
<tr>
<td>Hcount</td>
<td>number of bonded hydrogens</td>
<td>1</td>
</tr>
<tr>
<td>MorganHash</td>
<td>Morgan substructure at radius = 3</td>
<td>1</td>
</tr>
<tr>
<td>RingSize_list</td>
<td>ring size of all distinct rings</td>
<td>variable</td>
</tr>
<tr>
<td>Ring_count</td>
<td>number of distinct rings</td>
<td>1</td>
</tr>
<tr>
<td>chirality</td>
<td>unspecified, tetrahedral CW/CCW, or achiral</td>
<td>1</td>
</tr>
</tbody>
</table>

For features that are a list of discrete variables, the associated elementary kernel is a sequence convolution of Kroneker deltas:

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

(2)

Table 2. Bond Features for Marginalized Graph Kernel

<table>
<thead>
<tr>
<th>feature</th>
<th>description</th>
<th>size</th>
</tr>
</thead>
<tbody>
<tr>
<td>bond type</td>
<td>bond order, single, double, triple, or aromatic</td>
<td>1</td>
</tr>
<tr>
<td>stereo</td>
<td>none/E/Z for double bond</td>
<td>1</td>
</tr>
<tr>
<td>ring-stereo</td>
<td>none/E/Z for single bond in a ring</td>
<td>1</td>
</tr>
<tr>
<td>conjugated</td>
<td>whether the bond is conjugated</td>
<td>1</td>
</tr>
</tbody>
</table>

where

$$
f(l_1, l_2) = \sum_{\phi_i \in l_1} \sum_{\phi_j \in l_2} \delta(\phi_i, \phi_j)
$$

(3)

Here, $l_1$ and $l_2$ 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:

$$
\kappa(v, v') = \sum c_{j} \mu_{j}(\phi(v), \phi(v'))
$$

(4)

$$
\kappa(e, e') = \sum c_{j} \mu_{j}(\phi(e), \phi(e'))
$$

(5)

where $\mu_j$ is the elementary kernel for the $j$th feature $\phi_j$ 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:

$$
p_\kappa(v) = 1.0 + \sum_{k \in \mathcal{K}} p_k(v)
$$

(6)

$$
p_k(v) = \begin{cases} 
p, & v \text{ in group } k \\
0, & \text{otherwise}
\end{cases}
$$

(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 $\mathcal{K} = \{B, C, N, O, F, Si, P, S, Cl, Br, I\}$ in practice.

The stopping probability is set to be a hyperparameter $p_0$ 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 $G'$.
The learned atom hidden states are
\[ \mathbf{x}_v^t = \text{ReLU}(\mathbf{W}_m \mathbf{h}_v^t) \]
where \( \mathbf{W}_m \) is the learned matrix, and the bond features \( \mathbf{e}_{vw} \) are defined to describe the chemical environment of atoms and bonds. The initial edge hidden states are
\[ \mathbf{h}_{vw}^0 = \tau(W_c \text{cat}(\mathbf{x}_v, \mathbf{e}_{vw})) \]
where \( W_c \) is a learned matrix, and \( \tau \) is the ReLU activation function. The message-passing update equations are
\[ m_{vw}^{t+1} = \sum_{k \in N(v) \setminus w} h_{kv}^t \]
\[ h_{vw}^{t+1} = \tau(h_{vw}^0 + W_m m_{vw}^{t+1}) \]
where \( N(v) \) are the neighbors of \( v \) and \( W_m \) is a learned matrix. The learned atom hidden states are
\[ m_v = \sum_{w \in N(v)} h_{vw}^T \]
\[ h_v = \tau(W_c \text{cat}(\mathbf{x}_v, m_v)) \]
The molecular representation is the mean of atom hidden states
\[ h = \frac{1}{|G|} \sum_{i \in G} h_i \]  \hspace{1cm} (17)

Readout phase: The final property is obtained through a feed-forward neural network \( \hat{f}(\cdot) \)
\[ \hat{y} = f(h) \]  \hspace{1cm} (18)

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 information about D-MPNN can be found in ref 16.

**RDKit-Calculated Features.** The GP-MGK and D-MPNN models are sketched in Figure 1. Both use hybrid molecular representations of graphs and descriptors. A total of 200 global features computed using RDKit were concatenated with the learned molecular representation through message passing. To make a fair comparison, we added the 200 features in GP-MGK using a hybrid kernel:
\[ K(G, G') = K_G(G, G') K_F(F_{RDKit}, F_{RDKit}') \]  \hspace{1cm} (19)

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
\[ K_F(F_{RDKit}, F_{RDKit}') = \exp \left( -\frac{\| F_{RDKit} - F_{RDKit}' \|_2^2}{\sigma^2} \right) \]

**Implementation.** We used the GraphDot27 python package to compute the marginalized graph kernels and perform GPR, the scikit-learn package to carry out GPC, principal component analysis (PCA), and kernel PCA,26 the Descriptatorus package28 to calculate the RDKit features, and the Hyperopt package29 to optimize hyperparameters. All codes for the GP-MGK are written in python, available in our GitHub repository.30

**III. EXPERIMENTS**

**Data Sets.** The publicly available data sets used in this study are listed in Table 3. These data sets are commonly used for benchmark studies in molecular property prediction.6,16

**Hyperparameters Optimization.** There are 48 hyperparameters for GPR, 47 hyperparameters for GPC, and 4 hyperparameters for D-MPNN. We used the Tree of Parzen Estimators (TPE) to optimize hyperparameters to obtain optimal performance.35,40

For GPR, we used 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 were selected. The optimal hyperparameters are listed in Table S1. For GPC, the data was split 10 times at a ratio of 80:20, and Bayesian optimizations of 100 iterations were performed to determine the best hyperparameters based on the averaged performance on the 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 discuss below, it is adjusted by minimizing the miscalibration area.

For D-MPNN, we optimized the hyperparameters following the setting of Yang et al.16 The data was split 10 times at a ratio of 80:10:10, and Bayesian optimizations of 20 iterations were performed to determine the best hyperparameters based on the averaged performance on the validation sets of the 10 data splits.16 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 used the training set to build the model and make predictions on the test set. The data of the validation set was 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, the 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 evidence of the model to a sample of data, and minimize NLL is commonly used as the loss function for UQ.33,41

The 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 one way between the calibration curve and the diagonal.

**IV. RESULTS AND DISCUSSION**

We compare 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 ensemble GPR-MGK (GPC-MGK for classification) and “D-MPNN Ensemble”, which simply averages their predicted values.

---

**Table 3. Data Sets Used in This Paper**

<table>
<thead>
<tr>
<th>data set</th>
<th>task type</th>
<th>no. of tasks</th>
<th>no. of compounds</th>
<th>metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESOL</td>
<td>regression</td>
<td>1</td>
<td>1128</td>
<td>RMSE</td>
</tr>
<tr>
<td>FreeSolv</td>
<td>regression</td>
<td>1</td>
<td>642</td>
<td>RMSE</td>
</tr>
<tr>
<td>Lipophilicity</td>
<td>regression</td>
<td>1</td>
<td>4200</td>
<td>RMSE</td>
</tr>
<tr>
<td>PDBbind-C</td>
<td>regression</td>
<td>1</td>
<td>168</td>
<td>RMSE</td>
</tr>
<tr>
<td>PDBbind-R</td>
<td>regression</td>
<td>1</td>
<td>3040</td>
<td>RMSE</td>
</tr>
<tr>
<td>PDBbind-F</td>
<td>regression</td>
<td>1</td>
<td>9880</td>
<td>RMSE</td>
</tr>
<tr>
<td>QM7</td>
<td>regression</td>
<td>1</td>
<td>6830</td>
<td>MAE</td>
</tr>
<tr>
<td>BACE</td>
<td>classification</td>
<td>1</td>
<td>1513</td>
<td>ROC-AUC</td>
</tr>
<tr>
<td>BBP</td>
<td>classification</td>
<td>1</td>
<td>2039</td>
<td>ROC-AUC</td>
</tr>
<tr>
<td>SIDER</td>
<td>classification</td>
<td>27</td>
<td>1427</td>
<td>ROC-AUC</td>
</tr>
<tr>
<td>ClinTox</td>
<td>classification</td>
<td>2</td>
<td>1478</td>
<td>ROC-AUC</td>
</tr>
</tbody>
</table>
Benchmark on the 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, and 0.006 for repeating times of 5, 25, 50, and 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.42

GP-MGK vs D-MPNN: Prediction Performance. We first compare GPR-MGK with D-MPNN Ensemble on the ESOL data set. In Figures 3A and 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 at the same level, and ensembling prediction by averaging both predictions is better. In Figures 3C and 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 Figures 3E and 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 correlations between the prediction errors 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 stems from the fact that both models are closely related to the Weisfeiler–Lehman graph isomorphism test.

The regression results are numerically summarized in Table 4 and graphically summarized in the left of Figure 4. There are a total of 14 cases (seven data sets × 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

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. Comparison between GPR-MGK and D-MPNN Ensembles. Top: Random split. Bottom: Scaffold split. (A, B) The predictions 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.
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 data set using scaffold splitting.

The classification results are numerically summarized in Table 5 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 that of D-MPNN Ensemble, and the ensemble prediction of GPR-MGK and D-MPNN Ensemble is the best. For the ClinTox data set, D-MPNN outperforms GPC-MGK.

**GP-MGK vs D-MPNN: Principal Component Analysis (PCA).** Based on the results of the previous section, the correlation of the prediction errors of GPR-MGK and D-MPNN is visualized in Figure 4. The scatter plots and the corresponding histograms show the distribution of prediction errors for each data set and each model.

**Table 4. Prediction Results of GPR-MGK, D-MPNN, and Their Ensembling Model**

<table>
<thead>
<tr>
<th>data set</th>
<th>ESOL</th>
<th>FreeSolv</th>
<th>Lipophilicity</th>
<th>PDBbind-C</th>
<th>PDBbind-R</th>
<th>PDBbind-F</th>
<th>QM7</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPR-MGK</td>
<td>0.547 ± 0.050</td>
<td>0.822 ± 0.173</td>
<td>0.595 ± 0.037</td>
<td>1.940 ± 0.289</td>
<td>1.302 ± 0.049</td>
<td>1.284 ± 0.026</td>
<td>53.22 ± 3.12</td>
</tr>
<tr>
<td>D-MPNN Optimized</td>
<td>0.570 ± 0.054</td>
<td>0.904 ± 0.184</td>
<td>0.551 ± 0.044</td>
<td>1.849 ± 0.236</td>
<td>1.324 ± 0.052</td>
<td>1.279 ± 0.030</td>
<td>59.71 ± 3.40</td>
</tr>
<tr>
<td>D-MPNN Ensemble</td>
<td>0.557 ± 0.051</td>
<td>0.882 ± 0.175</td>
<td>0.539 ± 0.046</td>
<td>1.853 ± 0.232</td>
<td>1.297 ± 0.048</td>
<td>1.261 ± 0.029</td>
<td>57.06 ± 3.34</td>
</tr>
<tr>
<td>Ensemble a</td>
<td>0.537 ± 0.049</td>
<td>0.817 ± 0.167</td>
<td>0.534 ± 0.041</td>
<td>1.812 ± 0.239</td>
<td>1.273 ± 0.046</td>
<td>1.244 ± 0.026</td>
<td>50.29 ± 3.13</td>
</tr>
</tbody>
</table>

**Table 5. Prediction Results of GPC-MGK, D-MPNN, and Their Ensembling Model**

<table>
<thead>
<tr>
<th>data set</th>
<th>BACE</th>
<th>BBBP</th>
<th>SIDER</th>
<th>ClinTox</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPC-MGK</td>
<td>0.883 ± 0.028</td>
<td>0.921 ± 0.023</td>
<td>0.658 ± 0.023</td>
<td>0.774 ± 0.081</td>
</tr>
<tr>
<td>D-MPNN Optimized</td>
<td>0.893 ± 0.026</td>
<td>0.924 ± 0.021</td>
<td>0.655 ± 0.026</td>
<td>0.900 ± 0.049</td>
</tr>
<tr>
<td>D-MPNN Ensemble</td>
<td>0.899 ± 0.024</td>
<td>0.927 ± 0.021</td>
<td>0.664 ± 0.026</td>
<td>0.907 ± 0.044</td>
</tr>
<tr>
<td>Ensemble a</td>
<td>0.901 ± 0.024</td>
<td>0.931 ± 0.021</td>
<td>0.671 ± 0.025</td>
<td>0.872 ± 0.053</td>
</tr>
</tbody>
</table>

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

**“Ensemble prediction of GPR-MGK and D-MPNN Ensemble.”**

**“Ensemble prediction of GPR-MGK and D-MPNN Ensemble.”**
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: Number of heavy atoms. Bottom: Log solubility.
MPNN indicates that their molecular representations are similar. Furthermore, in this section, we performed PCA on the latent representations of D-MPNN and kernel PCA on the hybrid kernel to find more evidence of their similarity. “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. It is noticed that the molecular representations are indeed high-dimensional and complicated. Especially for D-MPNN, it is always difficult for humans to understand and interpret the GNNs learned by the backpropagation algorithm. Therefore, based on the results in this section, we can understand more about the molecular representations of MGK and D-MPNN, but it is insufficient to draw definite conclusions.

In Figure 5, the eigenvalue spectra of kernel matrices are plotted. For D-MPNN, the kernel matrices are computed by the elementwise dot product. The eigenvalues of D-MPNN latent 1 decay quickly, which indicates that 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 explains why Hirschfeld et al. used D-MPNN latent 2 as the input of union-based methods for UQ rather than D-MPNN latent 1. Relatively, D-MPNN latent 2 is closer 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. The three columns are the data embedding of MGK, D-MPNN latent 1, D-MPNN latent 2, respectively. In each column, the data embedding is the same, but the three rows represent (1) whether the molecule is cyclic, (2) the number of heavy atoms of the molecules, and (3) the value of the target property, respectively. In the data embedding of D-MPNN latent 1, cyclic and noncyclic molecules are separated, and the distribution of the number of heavy atoms is messy. Compared with D-MPNN latent 1, the data embeddings of MGK and D-MPNN latent 2 are similar. A small part of cyclic and noncyclic molecules overlaps, and the distribution of the number of heavy atoms is ordered. The similarity provides an indirect clue as to why the prediction errors of GPR-MGK and D-MPNN are correlated. The data embedding of other data sets is shown in Figures S6–S15.

**GP-MGK vs D-MPNN: Uncertainty Quantification.**

GPR-MGK is a Bayesian inference method, and its prediction is a Gaussian distribution. Therefore, the variance of the predicted Gaussian distribution can be used for UQ. This is crucial for the prediction of molecular properties because insufficient data hinders the training of an ML model applicable for all molecules. 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). In this work, we use D-MPNN MVE for comparison 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 analysis, we only used random data 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 varies 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 are 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 functions 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 the 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 6, 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 contradictory 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, \( R^2 \) is lower than 0.5, indicating that the data noise is too high. For the QM7 data set, 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 1 order of magnitude and provide reliable UQ. Therefore, we conclude that the UQ of GPR-MGK is better than that of 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 100 M molecules through active learning, 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.

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 D-MPNN, which indicates that the expressive power of a well-defined graph kernel can be as strong as GNNs. In addition, by comparing the predictions on a molecule-by-molecule basis, a correlation between the prediction errors of GPR-MGK and D-MPNN was observed. In addition, the ensemble prediction of GPR-MGK and D-MPNN is more accurate than either of them.

For UQ, we demonstrate that 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, the application guidance of GP-MGK and D-MPNN can be drawn. The advantage of D-MPNN lies in computational efficiency and scalability of large-scale data sets, so it is suitable for property prediction tasks and active learning for large-scale (million) data sets. The advantage of the GP-MGK is accurate uncertainty quantification, which can be applied to small-scale data sets (less than 50k) for active learning and data noise detection.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.1c01118.

Optimal hyperparameters of GP-MGK and D-MPNN on all data sets, model performance comparisons based on \( R^2 \) values, calibration curve of UQ, further comparisons of GP-MGK and D-MPNN predictions on both random and scaffold-based splits of all data sets, PCA on all data sets, and comparisons of GP-MGK and D-MPNN UQ on all data sets (PDF).

AUTHOR INFORMATION

Corresponding Authors

Huai Sun — School of Chemistry and Chemical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China; orcid.org/0000-0002-0783-8194; Email: huaisun@sjtu.edu.cn

Guang Lin — Department of Mathematics & School of Mechanical Engineering, Purdue University, West Lafayette,

Table 6. Uncertainty Quantification of GPR-MGK and D-MPNN MVE

<table>
<thead>
<tr>
<th>data set</th>
<th>ESOL</th>
<th>FreeSolv</th>
<th>Lipophilicity</th>
<th>PDBbind-C</th>
<th>PDBbind-R</th>
<th>PDBbind-F</th>
<th>QM7</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPR-MGK NLL</td>
<td>0.824</td>
<td>1.037</td>
<td>0.900</td>
<td>2.380</td>
<td>1.699</td>
<td>1.690</td>
<td>156.4</td>
</tr>
<tr>
<td>NLL</td>
<td>0.018</td>
<td>0.026</td>
<td>0.007</td>
<td>0.014</td>
<td>0.003</td>
<td>0.006</td>
<td>0.049</td>
</tr>
<tr>
<td>D-MPNN NLL</td>
<td>0.877</td>
<td>1.286</td>
<td>0.836</td>
<td>2.137</td>
<td>1.725</td>
<td>1.661</td>
<td>80.27</td>
</tr>
<tr>
<td>NLL</td>
<td>0.032</td>
<td>0.028</td>
<td>0.014</td>
<td>0.010</td>
<td>0.006</td>
<td>0.003</td>
<td>0.029</td>
</tr>
</tbody>
</table>
We thank Connor Coley for his help in performing uncertainty quantification.

REFERENCES

(28) Tang, Y.-H.; Selvitori, R.; 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. DOI: 10.1109/IPDPS47924.2020.00080.


(32) Haussler, D. *Convolution Kernels on Discrete Structures*; Technical report; Department of Computer Science, University of California at Santa Cruz: Santa Cruz, CA, 1999.


