JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

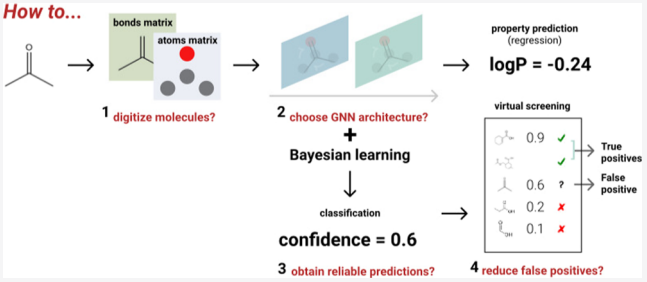
Comprehensive Study on Molecular Supervised Learning with Graph Neural Networks

Doyeong Hwang, Soojung Yang, Yongchan Kwon, Kyung Hoon Lee, Grace Lee, Hanseok Jo, Seyeol Yoon, and Seongok Ryu*

Cite This: J. Chem. Inf. Model. 2020, 60, 5936−5945 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: This work considers strategies to develop accurate and reliable graph neural networks (GNNs) for molecular property predictions. Prediction performance of GNNs is highly sensitive to the change in various parameters due to the inherent challenges in molecular machine learning, such as a deficient amount of data samples and bias in data distribution. Comparative studies with well-designed experiments are thus important to clearly understand which GNNs are powerful for molecular supervised learning. Our work presents a number of ablation studies along with a guideline to train and utilize GNNs for both molecular regression and classification tasks. First, we validate that using both atomic and bond meta-information improves the prediction performance in the regression task. Second, we find that the graph isomorphism hypothesis proposed by [Xu, K.; *et al* How powerful are graph neural networks? 2018, arXiv:1810.00826. arXiv.org e-Print archive. https://arxiv.org/abs/1810.00826] is valid for the regression task. Surprisingly, however, the findings above do not hold for the classification tasks. Beyond the study on model architectures, we test various regularization methods and Bayesian learning algorithms to find the best strategy to achieve a reliable classification system. We demonstrate that regularization methods penalizing predictive entropy might not give well-calibrated probability estimation, even though they work well in other domains, and Bayesian learning methods are capable of developing reliable prediction systems. Furthermore, we argue the importance of Bayesian learning in virtual screening by showing that well-calibrated probability estimation may lead to a higher success rate.



INTRODUCTION

The improvement in quantitative structure−activity relationship (QSAR) modeling has long been with the history of computational chemistry.[2] QSAR models aim to predict molecular properties using structural descriptors such as Morgan fingerprints,[3] simplified molecular-input line-entry systems (SMILES), and molecular graphs. Among these expressions, the most natural way to digitize a molecular structure is a molecular graph. Recent advancements in graph neural networks (GNNs) have extended the use of deep neural networks in molecular applications such as property prediction,[4−7] molecular design,[8−11] and chemical reaction planning.[12,13]

Even though various GNNs have been proposed, we would like to claim that it is difficult to clearly understand which GNNs are the most suitable for molecular applications. Of course, there are previous research studies devoted to the benchmark of molecular supervised learning via GNNs.[7,14,15] However, while certain pitfalls regarding data-inherent problems in chemistry should have been dealt with to guarantee the fair comparison of GNNs, previous works have not straightforwardly addressed such problems. First, training and evaluating models with small-sized data sets, where most of the molecular data sets are in this situation, usually lead to a large deviation in prediction performance with respect to various configurations in model developments, e.g., model architectures, hyperparameters, and random seeds. Moreover, in most cases, the data set is biased in the sense that it contains structurally similar molecules. Not only that, the ground truth labels are often imbalanced. Such a bias occurs frequently in classification data sets in particular. To alleviate such problems, it has been recommended to carefully follow some guidelines such as scaffold splitting of data sets[7] and averaging results from a sufficient number of random initialization seeds. Nevertheless, the prediction results are highly sensitive to the changes in the model and training schemes, which emphasize the importance of ablation studies to understand how GNNs work.

Received: April 22, 2020
Published: November 9, 2020

There are other challenges in developing both accurate and reliable models, arising from the nature of statistical learning, even if we designed well-performing GNN architectures. Since modern neural networks are prone to make overconfident predictions, the predictive output is usually measured higher than the expected true (accuracy).[16] For example, for binary classification tasks, one tends to interpret the final output as a probability of belonging to a target class. If an output of a perfect model is 0.8, then one can interpret the result that the predictive label is positive with an 80% probability of correctness. Such a probabilistic interpretation enables one to rely on the final model output when they aim to select compounds that are more likely expected to belong to a specific target class. However, actual accuracy from an overconfident model may be lower than 80% for predictions with an output probability of 0.8, and such a discrepancy may eventually discourage robust decision making.

Reliable prediction systems would be more crucial for virtual screening, which is one of the key goals of computational chemists.[17] For example, in virtual screening to find COVID-19 therapeutics, a compound with 0.9 predictive probability will be considered having a 90% probability of being active and thus will be taken into account for experimental validation. However, overconfident predictions will entail an unexpectedly large number of false positive predictions. In that sense, evaluating and improving prediction reliability would be essential for successful virtual screening with ML models.

In this work, we comprehensively study well-performing GNNs for molecular regression and classification tasks with various perspectives. Our study aims to answer the following questions, which might be naturally raised by those who hope to develop accurate and reliable GNNs rather than to seek state-of-the-art performance.

- "How to digitize molecular graphs?" Since there is no unique expression method of molecular graphs, in contrast to the existence of standard ways for digitizing image and natural language data, we try to understand how atomic and bond meta-information affect prediction performance.
- "How to design powerful GNN architecture for molecular prediction tasks?" We hypothesize that the GNN that satisfies the graph isomorphism (GI) test will show good prediction performance, which was first proposed by Xu et al.[1] Our experimental results show that it is valid for the regression task but invalid for classification tasks. These surprising results make us rethink how GNNs solve given labels.
- "How can we improve the reliability of GNNs for molecular classification tasks?" We try to overcome the overconfident prediction issue by adopting regularization and Bayesian learning methods, which are widely used in other research domains, such as computer vision and natural language processing. We find that inappropriate regularization methods can deteriorate correct probability estimation, even if they were effective in other benchmark studies,[18−20] and provide theoretical analysis on such consequence. On the other hand, Bayesian learning methods consistently improve prediction reliability for all model architectures and classification data sets.
- "How can we develop effective GNNs for virtual screening?" We suggest Bayesian GNNs for virtual screening, as the method gives the ability to correctly estimate the predictive probability. In a virtual screening scheme, most queried compounds in a library would be out-of-distributions (OODs) to the training data set. Thus, it is reasonable to conjecture that the outcomes will not be predicted with high confidence. Accordingly, when the model queries an external data set, the fraction of the predicted probabilities located near 0 or 1 should be small. Bayesian GNNs show the proper behavior, while the baseline model provides a large number of overconfident predictions.

We have released our implementation based on PyTorch and Deep Graph Library[21] at https://github.com/AITRICS/mol_reliable_gnn.

RESULTS AND DISCUSSION ON REGRESSION TASKS

We first investigate the powerfulness of different GNNs on p log P regression tasks, as similarly performed by Dwivedi et al.[15] The expression of p log P for the molecule m is defined as

$$\text{p log } P(m) = \log P(m) - \text{SAS}(m) - \text{cycle}(m) \quad (1)$$

where $\log P(\cdot)$ is the octanol−water partition coefficient, SAS$(\cdot)$ is the synthetic accessibility score, and cycle$(\cdot)$ is the number of rings that have more than six atoms. We used 50 000/5000/5000 samples for train/validation/test with 8-fold scaffold splitting,[7] while 10 000/1 000/1000 with 4-fold splitting was used by Dwivedi et al.[15] The reason for choosing the p log P regression task as our benchmark test is described below.

- Since p log P is the combination of three different molecular properties, as shown in eq 1, the task might be more complicated than a single property prediction and would effectively reduce the computational costs to perform experiments on each molecular property prediction.
- Since it is able to calculate p log P of an arbitrary molecule using RDKit,[22] we can obtain an enormously large number of labeled data samples for the benchmark study, where the calculated p log P value is considered the ground truth label.
- A lot of previous works[15,23,24] adopted p log P labels for the assessment of their models' predictive or conditional generative performance.

Using an Appropriate Input Expression Is Necessary. Our first concern is the input expression of molecular graphs that can correctly feature molecules and distinguish two different molecules. For this purpose, atom attributes are expressed with atomic types as well as additional meta-information, e.g., number of attached hydrogens, bond degree, implicit valence, and aromaticity, as proposed by Duvenaud et al.[4] This meta-information is especially helpful for the case of representing molecules only with connectivity. Also, using bond types as bond attributes can also enhance the expressive power of molecular graphs. In Figure 1, we exemplify the featurization results for propanone and 2-propanol. Without both atomic meta-information and bond types, two molecules are featured on the same molecular graph. On the other hand, using additional atom and bond attributes enables us to distinguish the two given molecules, which emphasize the importance of using correct featurization.

https://dx.doi.org/10.1021/acs.jcim.0c00416
J. Chem. Inf. Model. 2020, 60, 5936−5945

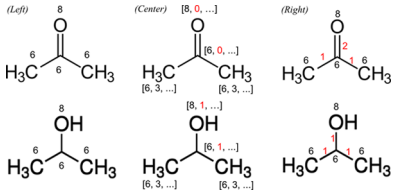


Figure 1. Two different molecules, propanone (top) and 2-propanol (bottom), featured by three approaches. (Left) Using atomic types for node features and connectivity for edge features. This input digitization cannot distinguish the molecules. (Middle) Using atomic types and the number of attached hydrogens (and additional meta-information; colored in red) for node features and connectivity for edge features. (Right) Using atomic types for node features and bond types (colored in red) for edge features. The last two digitization methods can distinguish the molecules.

Table 1 shows the p log *P* prediction results of using molecular graphs expressed with and without atomic meta-information. We can confirm that using atomic meta-information is obviously helpful for prediction performance. This might be because a more detailed description of atoms enables us to distinguish different (hydrogen-included) substructures more easily and results in a richer representation of molecular graphs. As the results from GatedGCN and GGNN demonstrated, GNN architectures that incorporate bond features in node-updating layers show superior performance than the others.

Table 1. Mean Absolute Error of the p log *P* Prediction Results of the Models Using Input Expression Utilizing the Atomic Meta-Information or Not[a]

	using meta-info	no meta-info
GCN	0.3431 (0.0136)	1.0387 (0.0118)
GraphSAGE	0.2786 (0.0010)	0.9937 (0.0062)
GIN	0.3134 (0.0093)	1.0301 (0.0118)
GAT	0.3268 (0.0057)	1.0398 (0.0163)
GatedGCN	0.2549 (0.0081)	0.5747 (0.0179)
GGNN	0.2532 (0.0053)	0.5087 (0.0059)

[a]We report the mean and standard deviation of the results from eight different experiments with scaffold splitting of the data sets. We note that GatedGCN and GGNN utilize bond features and the other GNNs do not.

GNNs That Pass the Graph Isomorphism Test Show Better p log *P* Prediction Performance. We next attempt to investigate the expressive power of GNN architectures on p log *P* prediction performance according to the graph isomorphism test.[1] Though we use a correct molecular graph expression that can distinguish two different molecules, as investigated in the previous experiment, incorrect GNN architecture can lead to indistinguishable hidden representations. In the message passing framework of GNNs, node-updating layers aggregate messages from adjacent nodes and a readout layer aggregates node representations to produce a graph representation.[25] Based on this framework, an incorrect choice of an aggregation function can lead to the same graph representation for two different molecules, which can degrade the prediction performance of GNNs.

We show two example graphs, each of which consists of identical nodes but with different numbers of nodes in Figure 2. For brevity, let us denote the node representation of each node as h. If we use sum aggregation for the node-updating layer or readout layer, the aggregation results for the two graphs will be $(h + h + h) = 3h$ and $(h + h + h + h) = 4h$, respectively. On the other hand, if we use mean aggregation, the aggregation results will be $(1/3)(h + h + h) = h$ and $(1/4)(h + h + h + h) = h$ for the two graphs, which cannot distinguish the two graphs and therefore fail the graph isomorphism (GI) test.

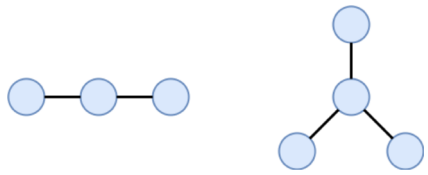


Figure 2. Two graphs, each of which consists of identical nodes, with three and four nodes, respectively.

Thus, we hypothesize that GNNs that can pass the GI test will show better prediction performance. Table 2 shows the prediction results of the GNNs whose node-updating layers aggregate message states (information from neighbor nodes) with mean and sum aggregations. Note that we only consider isotropic node-updating methods—GCN, GraphSAGE, and GIN—for testing the GI test, while GAT, GatedGCN, and GGNN are anisotropic node-updating methods. As shown in the results, we confirm that the sum aggregation shows a better prediction performance than the mean aggregation, which highlights that GNNs with higher expressive power is better for our regression study.

Table 2. Mean Absolute Error of p log *P* Predictions of the GNNs Whose Node-Updating Layers Aggregate Messages with Sum and Mean Aggregations[a]

	sum aggregation	mean aggregation
GCN	0.3431 (0.0136)	0.3524 (0.0121)
GraphSAGE	0.2786 (0.0010)	0.2943 (0.0055)
GIN	0.3134 (0.0093)	0.3194 (0.0167)

[a]We report the mean and standard deviation of the results from eight different experiments with scaffold splitting of the data sets.

Also, we examine readout methods including sum, mean, and max aggregations. We again note that only the sum readout can pass the GI test and the others cannot. Table 3 shows the prediction results of the GNNs with the three different readouts. For all six node-updating methods, the sum readout outperforms the other readouts. These results again emphasize the importance of choosing readout functions according to the GI test.

Table 3. Mean Absolute Error of the p log *P* Prediction Results of the Models Using the Sum, Mean, and Max Readouts[a]

	sum	mean	max
GCN	0.3431 (0.0136)	0.3696 (0.0110)	0.4209 (0.0125)
GraphSAGE	0.2786 (0.0100)	0.2818 (0.0088)	0.3565 (0.0061)
GIN	0.3134 (0.0093)	0.3407 (0.0101)	0.3791 (0.0077)
GAT	0.3268 (0.0057)	0.3672 (0.0093)	0.3863 (0.0064)
GatedGCN	0.2549 (0.0081)	0.2883 (0.0089)	0.2791 (0.0086)
GGNN	0.2532 (0.0053)	0.2920 (0.0076)	0.2833 (0.0036)

[a]We report the mean and standard deviation of the results from eight different experiments with scaffold splitting of the data sets.

https://dx.doi.org/10.1021/acs.jcim.0c00416
J. Chem. Inf. Model. 2020, 60, 5936−5945

Prediction Performance as the Number of Training Sample Changes. In this experiment, we attempt to answer the following question "Are GNNs that consist of a large number of weight parameters still powerful for a low-data training scenario?" Since many molecular supervised learning tasks have a small number of training data samples, we perform the experiment to investigate models' performance in a low-data regime.

Figure 3 shows the p log *P* prediction results of the GNNs trained with 1000, 5000, 10 000, 30 000, and 50 000 training examples. Note that we used the sum aggregation for the isotropic node-updating methods (i.e., GCN, GIN, and GraphSAGE) and for the readout of all GNNs. For the smallest number of training examples, that is, using 1000 labeled samples, all of the GNNs except GGNN show similar prediction performances. However, as the number of training examples increases, GatedGCN and GGNN that utilize bond features for node updating significantly outperform the other GNNs.

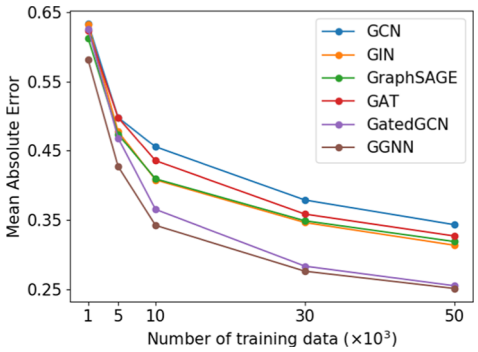


Figure 3. Mean absolute error of the p log *P* prediction results of the models trained with different numbers of training examples.

EXPERIMENTAL RESULTS ON CLASSIFICATION TASKS

In this section, we show our experimental results on the classification tasks—BACE, BBBP, HIV, and Tox21 prediction tasks. All prediction tasks are the binary classification tasks, which are widely used as benchmarks in molecular supervised learning. In Table 4, we show a detailed description of the four classification data sets. In our experiments, we split data sets by a ratio of 80:10:10% each for train/validation/test with an 8-fold scaffold splitting.[7] We performed all of the classification experiments, except for the results shown in Figure 4, with the GIN architecture. That is because (i) the performances of GNNs in the classification tasks were not significantly different, as shown in Figure 4, (ii) GIN can be a universal function approximator where graph input is used,[1] and (iii) GAT, GGNN, and GatedGCN required heavier computational costs than GIN, thus choosing GIN allowed us to perform a large number of ablation experiments with a limited computational budget.

Table 4. Specifications of the Four Data sets Used For Classification Tasks[a]

	binary classification			
task type	BACE	BBBP	HIV	Tox21
number of tasks	1	1	1	12
number of samples	1513	2050	41 127	
positives/negatives	822:691	483:1567	39 684:1443	

[a]Note that the Tox21 data set consists of 12 different binary classification tasks, each of which has different numbers of positive and negative samples. We refer to Mayr et al.[26] for more details on the Tox21 data set.

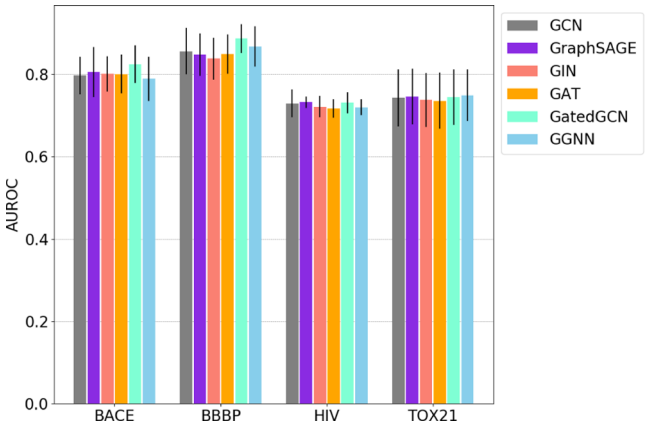


Figure 4. Prediction performance (AUROC) of models on the four classification tasks: BACE, BBBP, HIV, and Tox21 predictions. We report the mean and standard deviation of the results from eight different experiments with scaffold splitting of the data sets.

Strategies to Build Powerful GNNs in the Regression Task Do Not Hold for Classification Tasks. First, we investigate the effect of GNN architectures on the classification tasks. Figure 4 shows the performance of models in terms of area under receiver operating curve (AUROC). Note that we adopted the sum readout for this experiment. We can confirm that GNN architectures do not significantly affect the prediction performance of the classification tasks. While GatedGCN shows better performance than the other GNNs in three out of four tasks, it is not marginally better than the second-best model.

Next, we investigate the effect of readout methods on the classification tasks. Figure 5 shows the performance of models with sum, mean, and max aggregations, where GIN was adopted for node updating. Surprisingly, the max readout

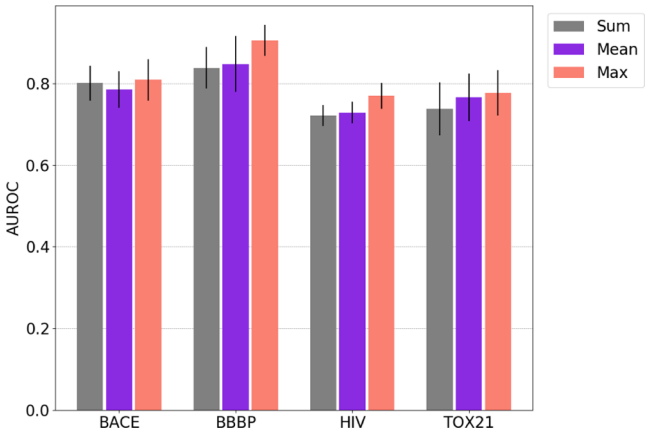


Figure 5. Prediction performance (AUROC) of GIN models with sum, mean, and max readouts on the four classification tasks: BACE, BBBP, HIV, and Tox21 predictions. We report the mean and standard deviation of the results from eight different experiments with scaffold splitting of the data sets.

https://dx.doi.org/10.1021/acs.jcim.0c00416
J. Chem. Inf. Model. 2020, 60, 5936−5945

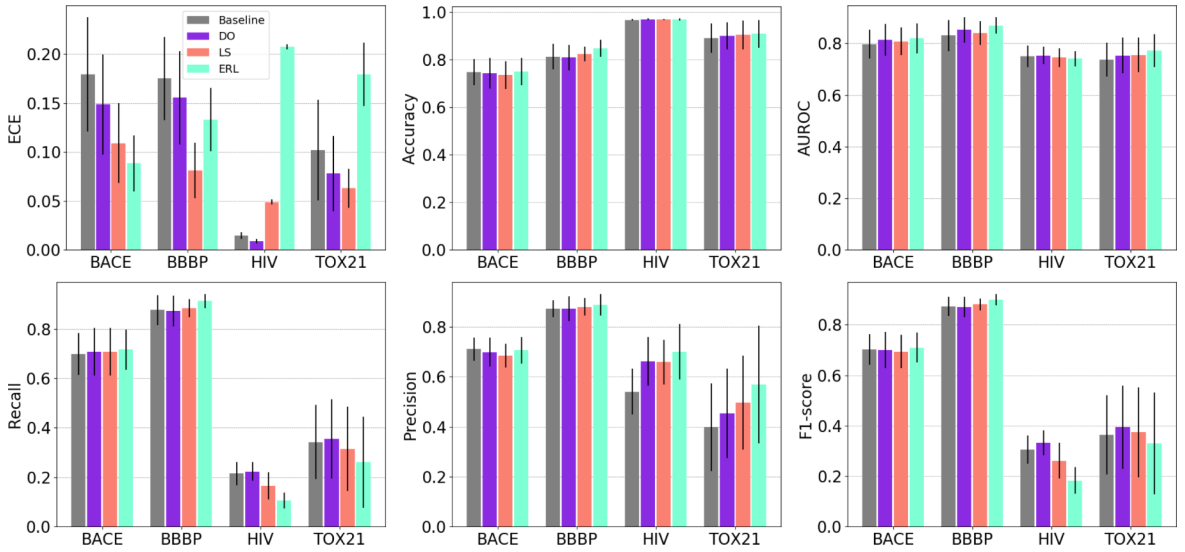


Figure 6. Prediction reliability (ECE; ↓) and performance (accuracy, AUROC, precision, recall, and F1-score; ↑) of the GIN model trained with different regularization methods for the BACE, BBBP, HIV, and Tox21 prediction tasks. We report the mean and standard deviation of the results from eight different experiments with scaffold splitting of the data sets.

outperforms the other readout methods, whereas the sum readout showed the best performance in the p log P regression task. Interestingly, our findings demonstrate that the more expressive GNN does not guarantee a better generalization ability, in contrast to the original work of the GI test, which suggests a positive correlation between the model expressive power and generalization (refer to page 10 of Xu et al[1]). Thus, our naturally raised question is "why does the max readout, which gives less expressive power than the sum readout, shows the best prediction performance in classification tasks?" We draw the following conjecture: "finding common (frequently observed) molecular substructures in positive (or negative) training samples is sufficient for molecular classification tasks". For example, it is widely accepted that a certain molecular substructure can induce toxicity in molecules. Inspired by such a chemists' rationale, rule-based medicinal chemistry filters[27] have been utilized to screen out the potentially toxic compounds in virtual screening processes,[10] relying on the (sub)structure−activity relationship. Along the same line of reasoning, as the max readout is effective for detecting key substructures that determine the true label of the given molecules, it may show the best performance among the three readouts in our demonstrations.

Evaluating Models with Prediction Reliability. Though our findings are not fully understood and remain as a conjecture, in the following experimental sections, we would like to claim other difficulties in molecular classification tasks—prediction results from the neural classification models are occasionally too high in value to be interpreted as predictive probabilities. Such a phenomenon is called over-confident behavior.[16]

Our goal is to develop classification systems whose final output can be regarded as a predictive probability and thus is eligible for more reliable decision making. To evaluate the prediction reliability of binary classification systems, we utilize calibration curves and compute the expected calibration error (ECE). If we divide the predictive results into a total of M number of bins (intervals), the fraction of positives (FOP) and the mean predicted value (MPV) of predictions in the mth bin B_m is given by

$$\mathrm{FOP}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{I}(\hat{y}_i = 1) \tag{2}$$

and

$$\mathrm{MPV}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i \tag{3}$$

where $|B_m|$ is the number of the samples in B_m, and $\mathbb{I}$ is an indicator function. Here, $\hat{p}_i$ is the probability of being a positive sample, and $\hat{y}_i$ is the corresponding predictive label. The calibration curve visualizes $\mathrm{FOP}(B_m)$ and $\mathrm{MPV}(B_m)$ for all bins $m \in \{0, ..., M\}$, as shown in Figure 7. The calibration error of each bin can be computed as a gap between the perfect calibration curve and the accuracy−confidence curve. Then, their average, ECE suggested by Naeini et al.,[28] summarizes the calibration errors over the entire data points, whose estimator is given by

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} |\mathrm{FOP}(B_m) - \mathrm{MPV}(B_m)| \tag{4}$$

where n is the number of data points. We used $M = 10$ for computing ECE in the experiments. Also, we obtained calibration curves[29] using the implementation provided by scikit-learn: https://scikit-learn.org/stable/modules/generated/sklearn.calibration.calibration_curve.html.

The remaining experimental sections investigate the efficacy of the regularization and Bayesian learning methods that have been widely used in other domains on the molecular classification tasks.

Appropriate Regularization Is Required for Reliable Predictions. We investigate the effect of the well-known regularization methods—dropout (DO),[30] label smoothing (LS),[18] and entropy regularization (ERL)[31]—on the prediction reliability of our baseline model, that is, GIN with the sum readout. We set the hyperparameters in the regularization methods as $p_{\mathrm{do}} = 0.2$, $\alpha_{\mathrm{LS}} = 0.1$, and $\beta_{\mathrm{ERL}} = 1.0$. (details on regularization methods are described in the Supporting Information).

https://dx.doi.org/10.1021/acs.jcim.0c00416
J. Chem. Inf. Model. 2020, 60, 5936−5945

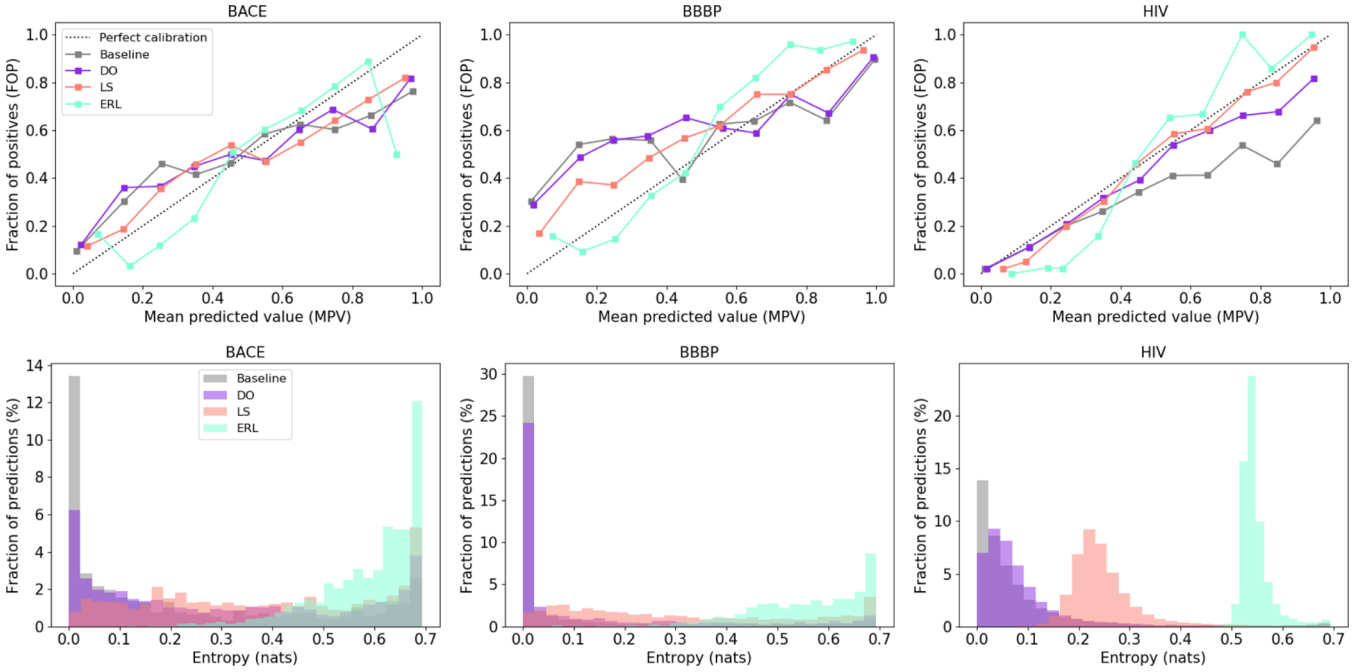


Figure 7. (Top) Calibration curves and (bottom) predictive entropy histograms of the GIN model trained with different regularization methods for BACE, BBBP, and HIV.

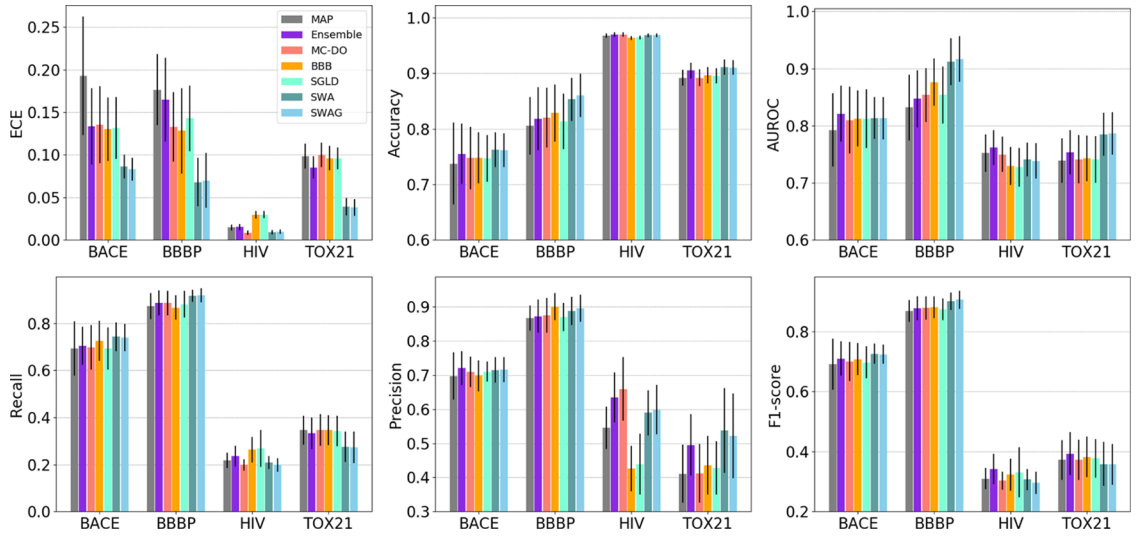


Figure 8. Prediction reliability (ECE; ↓) and performance (accuracy, AUROC, precision, recall, and F1-score; ↑) of the GIN model for the BACE, BBBP, HIV, and Tox21 prediction tasks. We report the mean and standard deviation of the results from eight different experiments with scaffold splitting of the data sets.

Figure 6 summarizes the prediction performance (accuracy, AUROC, recall, precision, and F1-score) and reliability (ECE) of the models implementing the above methods. All of the models resulted in a similar prediction performance. On the other hand, prediction reliability widely varied depending on regularization methods. Applying DO was effective for all the four classification tasks, while LS and ERL deteriorated the prediction reliability for some tasks. For example, LS and ERL showed significantly higher ECE results for the HIV prediction task. Also, ERL was not effective for the Tox21 prediction task.

Though our experimental investigation was limited to the four prediction tasks, we attempted to analyze such phenomena by theoretical justification, as shown in the Supporting Information. According to our theoretical analysis, a well-calibrated probability may not be granted for LS and ERL, which simply penalize predictive entropy. Figure 7 shows the calibration curves and entropy histograms of the regularization methods in the three prediction tasks. (We exclude the result from the Tox21 data set because it consists of 12 different binary classification tasks.) For the baseline model, predictive entropy values are largely frequent at 0.0, implying that the predictive outputs are mostly 0 or 1. On the contrary, for LS and ERL models, most of the predictive entropy values are larger than 0.0. It seems that ERL is showing an excessive regularization effect. In fact, minimizing the forward KL divergence between the predictive distribution and a uniform distribution sometimes gives such a result. The inadequate reliability of LS for HIV predictions and ERL for

https://dx.doi.org/10.1021/acs.jcim.0c00416
J. Chem. Inf. Model. 2020, 60, 5936−5945

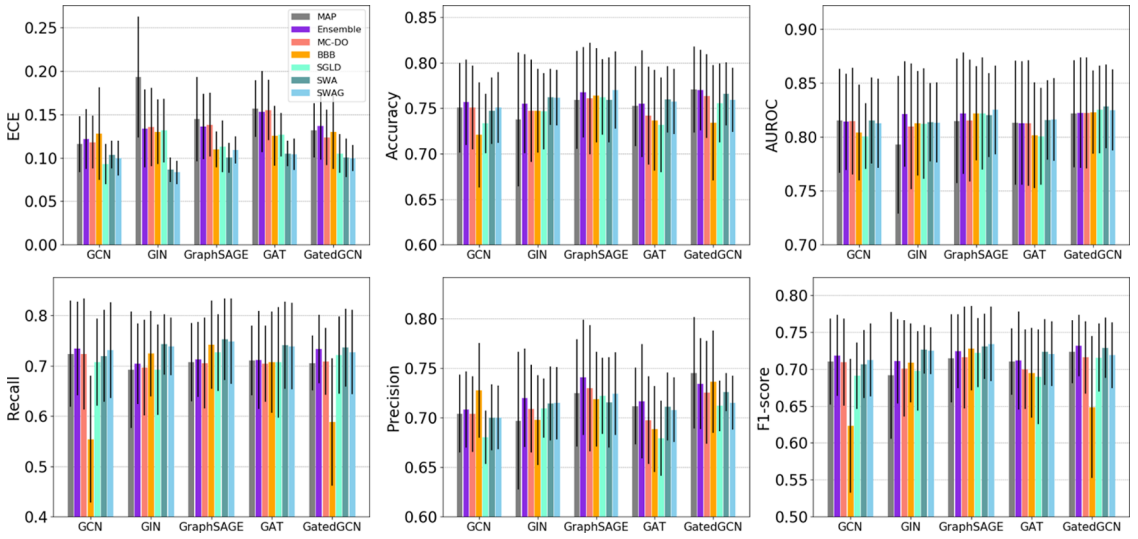


Figure 9. Prediction reliability (ECE; ↓) and performance (accuracy, AUROC, precision, recall, and F1-score; ↑) of the five different GNN models on the BACE prediction task. We report the mean and standard deviation of the results from eight different experiments with scaffold splitting.

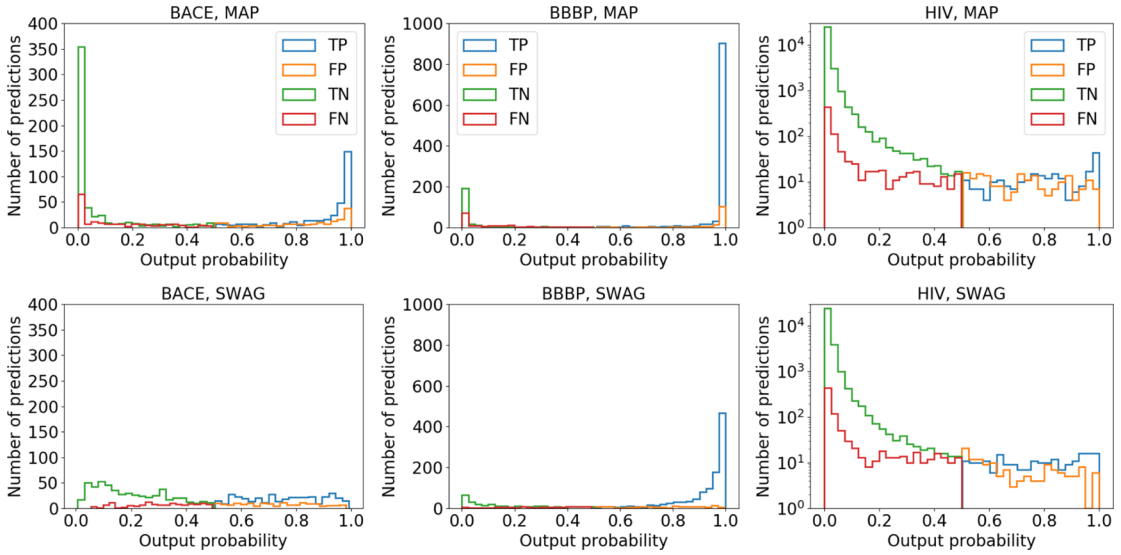


Figure 10. Histograms of true positive (TP), false positive (FP), true negative (TN), and false negative (FN) predictions from the GIN trained with MAP (top) and SWAG (bottom).

overall data sets can be explained. Also, as the lowered ECE values point out, applying DO and LS has diminished the deviation between the perfect calibration curve (the black dotted line) and the experimental calibration curves.

Bayesian Learning Enjoys Reliable Predictions. We have found that inappropriate regularizations—in our study, LS and ERL, which simply penalize the entropy of the predictive distribution—can lead to poorly calibrated predictive probability; in other words, poor prediction reliability. On the other hand, using dropout gives a well-calibrated probability thanks to its ability to approximate Bayesian model averaging.[32,33] Furthermore, recent machine learning research studies have shed light on Bayesian learning methods, as they enable us to obtain a well-calibrated prediction probability.[33−35] Thus, we further investigate the prediction behavior of molecular classification models when they are implemented with the recently proposed approximate Bayesian inference approaches.

We consider maximum a posteriori (MAP) estimation as the non-Bayesian baseline model, and deep ensemble,[36] Monte Carlo dropout (MC-DO),[32] stochastic gradient Langevin dynamics (SGLD),[37] stochastic weight averaging (SWA),[38] and stochastic weight averaging Gaussian (SWAG)[34] as our methods to demonstrate the effectiveness of Bayesian learning in reliable molecular property predictions.

In Figure 8, we show the prediction results for the four different prediction tasks—BACE, BBBP, HIV, and Tox21 predictions—where GIN was chosen as the baseline model architecture and various Bayesian learning methods (ensemble, MC-DO, BBB, SGLD, SWA, and SWAG) were implemented. We observe that most of the Bayesian methods are helpful for improving both prediction reliability (lower ECE) and performance (higher AUROC). Except for the HIV prediction task, all Bayesian methods led to a lower ECE value than the MAP model. In particular, SWA and SWAG show a lower ECE value compared to the other Bayesian methods (i.e., ensemble, MC-DO, BBB, and SGLD) as well as the MAP model.

https://dx.doi.org/10.1021/acs.jcim.0c00416
J. Chem. Inf. Model. 2020, 60, 5936−5945

Furthermore, SWA and SWAG show superior prediction performances (in terms of accuracy and AUROC) when compared to other Bayesian approaches in most cases.

Also, we attempted to check whether the Bayesian learning methods are effective for the GNN architectures other than GIN. In Figure 9, we show the prediction performance and reliability of the five GNN models (i.e., GCN, GIN, GraphSAGE, GAT, and GatedGCN) trained with different Bayesian learning methods on the BACE prediction task. We observe that SWA and SWAG consistently show better performance and reliability results compared to MAP. On the other hand, other methods show worse results for some GNN models compared to MAP. For example, ensemble, MC-DO, and BBB show higher ECE values than MAP. Specifically, BBB gives poor results for GCN and GatedGCN, showing significantly deteriorated accuracy, recall, and F1-score, despite the fact that we adopted a scaling factor to the Kullback–Leibler divergence term in the learning objective of BBB. (see the Supporting Information) Consequently, we can learn the lessons from Figures 8 and 9 that adopting SWA/SWAG is obviously beneficial for improving prediction reliability and gaining additional prediction performance.

Wilson and Izmailov[33] proposed that the ensemble of SWAG (MultiSWAG), which uses an ensemble of variational posterior to model a multimodal posterior, can improve the single SWAG. Motivated by the MultiSWAG, we compare the ECE and AUROC results of using single Bayesian models and the ensemble of Bayesian models on the four prediction tasks, as shown in Tables S1 and S2 (in the Supporting Information). Using ensemble additionally improved both prediction reliability and performance for all Bayesian approaches in most cases, while the amount of improvement gain is relatively smaller than that of SWA/SWAG. Thus, we conclude that using the ensemble of SWA/SWAG would be the best choice to accomplish both high prediction performance and reliability as long as enough computing resource is secured.

Bayesian GNNs are Able to Reduce the Amount of False Positives in Virtual Screening. Overconfident prediction behavior is frequently observed in neural networks, especially in MAP-estimated models. As shown in Figure 10, we can confirm that most prediction results from the MAP-estimated models are positioned near 0 or 1. On the other hand, SWAG models effectively mitigate overconfident predictions, which were quantitatively evaluated using ECE, as shown in Figures 9 and 8. A much smaller number of predictions were observed near 0 or 1, and the ratio between true positive (negative) and false positive (negative) was higher.

We show the results obtained with other Bayesian learning methods in Figures S1, S2, and S3 (see the Supporting Information). Interestingly, we can confirm that SWA and SWAG models are the most effective for reducing the number of predictions near 0 or 1 and mitigating overconfident predictions.

In addition, we demonstrate a virtual screening experiment—using the model trained with the BACE data set to screen BACE—active compounds from the lead-like subset of the ZINC database.[39] Though we do not have true labels of the compounds in the zinc data set, most of them might be out-of-distribution (OOD) against the samples in the BACE data set. Figure 11 shows the histogram of predictive probability for the zinc compounds inferred by the models trained with the BACE data set. We observe that most of the

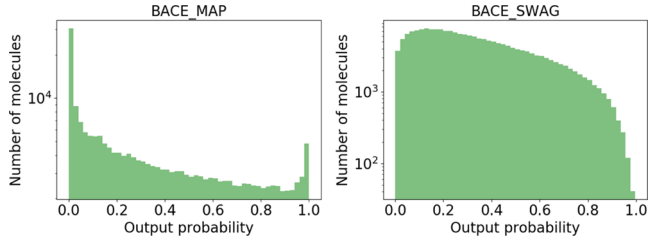


Figure 11. Histograms of the predictive probability of the BACE activity from the GIN models trained with MAP (left) and SWAG (right) for the compounds in the ZINC data set.

prediction results from the MAP model are positioned near 0 or 1, which might be unnatural for predictions on the OOD samples. On the other hand, the SWAG model shows a small number of predictions in which the probability is less than 0.05 or greater than 0.95 over entire negative or positive predictions, respectively. In particular, only 238 compounds show predictive probability values greater than 0.95 out of the total 200 000 samples. Considering the situation of selecting the experimental candidates using the output probability values, for example, selecting compounds with output probability higher than 0.95, our demonstrations shown in Figures 10 and 11 imply that Bayesian approaches would be able to give a higher success rate compared to non-Bayesian approaches in virtual screening.

Related Works: Uncertainty Quantification. The scope of our study does not cover the prediction reliability for regression models. A number of previous works[40−44] studied uncertainty quantification of regression tasks. Considering that property values obtained from experiments or simulations are usually noisy, quantifying both data-driven (aleatoric) and model-driven (epistemic) uncertainties would be helpful for molecular applications such as detecting erroneous results from experiments[40] and active learning.[41] The Bayesian learning methods incorporated in this study can also be utilized for regression tasks. Moreover, these methods might work better than the methods used in previous works, e.g., Monte Carlo dropout (MC-DO), for uncertainty quantification.

In addition, we note that the uncertainty of classification outputs can be quantified using the final predictive probability $\overline{p}$.[40,45] The variance of the predictive distribution is given by $\overline{p}(1 - \overline{p})$, since the predictive distribution of binary classification assumes the Bernoulli distribution. Albeit we did not explicitly use the term uncertainty quantification in this study, the following statement provides a simple insight, connecting our reliability study to uncertainty quantification: "The closer to 0.5 the predictive probability is, the more uncertain our prediction is". For further information regarding quantification of aleatoric and epistemic uncertainty in classification results, we refer to Kwon et al.[45] and Ryu et al.[40]

REMARKS AND CONCLUSIONS

We have studied molecular regression and classification tasks with GNNs from various perspectives—input expression, model architecture, regularization, and learning algorithms. To this end, we have gained the following remarkable lessons.

Lesson 1: Our ablation studies have revealed that suitable model architectures can be different for regression and classification tasks. In regression tasks, GNN models that pass the graph isomorphism test showed better prediction performance than models that fail the graph isomorphism test,

https://dx.doi.org/10.1021/acs.jcim.0c00416
J. Chem. Inf. Model. 2020, 60, 5936−5945

especially for designing readout layers. These results are in line with our and Xu et al.[1] hypothesis: "GNNs with higher expressive power will show better prediction performance". On the other hand, such a hypothesis was invalid for molecular classification tasks that were covered in this study. We conjecture that classifying molecules may not require exact summary statistics of atoms and/or substructures but instead seek important substructures to determine the true label. The fact that the best performance was given by the max readout model in our demonstration (Figure 5) supports our conjecture.

Lesson 2: To enhance the prediction reliability of classification models, appropriate learning algorithms and regularization are necessary. Previous research studies validated that models with a large number of weight parameters are vulnerable to overconfident prediction problems and thus lead to poor probability estimation and prediction reliability. Motivated by this, we focused on the effect of regularization and learning algorithms instead of the model architecture. Our study revealed that regularization methods based on penalizing predictive entropy (i.e., label smoothing and entropy regularization) can deteriorate prediction reliability, and theoretical analysis on these phenomena was omitted. On the other hand, Bayesian learning methods were effective for improving prediction performance, especially SWA and SWAG combined with deep ensemble were the most powerful approaches for calibrating prediction performance.

Lesson 3: Lastly, we demonstrate that Bayesian models are more suitable for achieving a high success rate in virtual screening, as shown in the prediction behavior of models (Figures 10 and 11). While the baseline model (MAP) shows highly overconfident results for the external data set, the SWAG model effectively mitigates the overconfident prediction problems and enables better probability estimation.

Comprehensively, we would like to claim that our studies give fruitful insights into how an accurate and reliable molecular property prediction system can be achieved with graph neural networks.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.0c00416.

> Backgrounds and implementations of graph neural networks; backgrounds of quantification of prediction reliability in classification; backgrounds of regularizations to alleviate overconfident predictions; interpretation of label smoothing and entropy regularization; backgrounds and implementations of Bayesian learning; and details on model training (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Seongok Ryu − *AITRICS, Seoul, Republic of Korea;* orcid.org/0000-0001-5752-6335; Email: seongokryu@aitrics.com

Authors

Doyeong Hwang − *AITRICS, Seoul, Republic of Korea*

Soojung Yang − *AITRICS, Seoul, Republic of Korea; Department of Chemistry, KAIST, Daejeon 34141, Republic of Korea*

Yongchan Kwon − *Department of Biomedical Data Science, Stanford University, Stanford, California 94305, United States*

Kyung Hoon Lee − *Department of Chemistry, KAIST, Daejeon 34141, Republic of Korea*

Grace Lee − *AITRICS, Seoul, Republic of Korea*

Hanseok Jo − *AITRICS, Seoul, Republic of Korea*

Seyeol Yoon − *AITRICS, Seoul, Republic of Korea*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.0c00416

Author Contributions

D.H., S.Y., and S.R. conceived the ideas, implemented them, and performed the experiments. Y.K. developed theoretical interpretation. All the authors analyzed the results and wrote the manuscript together.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the project NRF-2019M3E5D4065965.

■ REFERENCES

(1) Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How powerful are graph neural networks?. 2018, arXiv:1810.00826. arXiv.org e-Print archive. https://arxiv.org/abs/1810.00826.

(2) Cherkasov, A.; Muratov, E. N.; Fourches, D.; Varnek, A.; Baskin, I. I.; Cronin, M.; Dearden, J.; Gramatica, P.; Martin, Y. C.; Todeschini, R.; et al. QSAR modeling: where have you been? Where are you going to? *J. Med. Chem.* **2014**, *57*, 4977−5010.

(3) Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.

(4) Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. In *Convolutional Networks on Graphs For Learning Molecular Fingerprints*, Advances in Neural Information Processing Systems, 2015; pp 2224−2232.

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

(6) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. In *Neural Message Passing for Quantum Chemistry*, Proceedings of the 34th International Conference on Machine Learning-Volume 70, 2017; pp 1263−1272.

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

(8) De Cao, N.; Kipf, T. MolGAN: An Implicit Generative Model for Small Molecular Graphs. 2018, arXiv:1805.11973. arXiv.org e-Print archive. https://arxiv.org/abs/1805.11973.

(9) Sanchez-Lengeling, B.; Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science* **2018**, *361*, 360−365.

(10) Zhavoronkov, A.; Ivanenkov, Y. A.; Aliper, A.; Veselov, M. S.; Aladinskiy, V. A.; Aladinskaya, A. V.; Terentiev, V. A.; Polykovskiy, D. A.; Kuznetsov, M. D.; Asadulaev, A.; et al. Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nat. Biotechnol.* **2019**, *37*, 1038−1040.

(11) Hong, S. H.; Ryu, S.; Lim, J.; Kim, W. Y. Molecular Generative Model Based On Adversarially Regularized Autoencoder. *J. Chem. Inf. Model.* **2020**, *60*, 29−36.

(12) Coley, C. W.; Jin, W.; Rogers, L.; Jamison, T. F.; Jaakkola, T. S.; Green, W. H.; Barzilay, R.; Jensen, K. F. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chem. Sci.* **2019**, *10*, 370−377.

(13) Dai, H.; Li, C.; Coley, C.; Dai, B.; Song, L. In *Advances in Neural Information Processing Systems*, Advances in Neural Information Processing Systems, 2019; pp 8870−8880.
(14) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; et al. Analyzing learned molecular representations for property prediction. *J. Chem. Inf. Model.* **2019**, *59*, 3370−3388.
(15) Dwivedi, V. P.; Joshi, C. K.; Laurent, T.; Bengio, Y.; Bresson, X. Benchmarking Graph Neural Networks 2020, arXiv:2003.00982. arXiv.org e-Print archive. https://arxiv.org/abs/2003.00982.
(16) Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K. Q. In *On Calibration of Modern Neural Networks*, Proceedings of the 34th International Conference on Machine Learning-Volume70, 2017; pp1321−1330.
(17) Stokes, J. M.; Yang, K.; Swanson, K.; Jin, W.; Cubillos-Ruiz, A.; Donghia, N. M.; MacNair, C. R.; French, S.; Carfrae, L. A.; Bloom-Ackerman, Z.; et al. A Deep Learning Approach to Antibiotic Discovery. *Cell* **2020**, *180*, 688−702.
(18) Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. A. In *Inception-v4, Inception-resnet Andthe Impact of Residual Connections on Learning*, Thirty-first AAAI conference on Articial Intelligence, 2017.
(19) Zhang, H.; Cisse, M.; Dauphin, Y. N.; Lopez-Paz, D. mixup: Beyond Empirical Risk Minimization 2017. arXiv:1710.09412. arXiv.org e-Print archive. https://arxiv.org/abs/1710.09412.
(20) Müller, R.; Kornblith, S.; Hinton, G. E. In *When does label smoothing help?*, Advances in Neural Information Processing Systems, 2019; pp 4696−4705.
(21) Wang, M.; Yu, L.; Zheng, D.; Gan, Q.; Gai, Y.; Ye, Z.; Li, M.; Zhou, J.; Huang, Q.; Ma, C. et al. Deep graph library: Towards efficient and scalable deep learning on graphs 2019. arXiv:1909.01315v1. arXiv.org e-Print archive. https://arxiv.org/abs/1909.01315v1.
(22) Landrum, G.et al.*RDKit: Open-Source Cheminformatics Software*, 2006.
(23) Jin, W.; Barzilay, R.; Jaakkola, T. Junction tree variational autoencoder for molecular graph generation 2018. arXiv:1802.04364. arXiv.org e-Print archive. https://arxiv.org/abs/1802.04364.
(24) You, J.; Liu, B.; Ying, Z.; Pande, V.; Leskovec, J. In *Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation*, Advances in Neural Information Processing Systems, 2018; pp 6410−6421.
(25) Battaglia, P. W.; Hamrick, J. B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R. et al. Relational inductive biases, deep learning, and graph networks 2018. arXiv:1806.01261. arXiv.org e-Print archive. https://arxiv.org/abs/1806.01261.
(26) Mayr, A.; Klambauer, G.; Unterthiner, T.; Hochreiter, S. DeepTox: toxicity prediction using deep learning. *Front. Environ. Sci.* **2016**, *3*, No. 80.
(27) Dahlin, J. L.; Nissink, J. W. M.; Strasser, J. M.; Francis, S.; Higgins, L.; Zhou, H.; Zhang, Z.; Walters, M. A. PAINS in the assay: chemical mechanisms of assay interference and promiscuous enzymatic inhibition observed during a sulfhydryl-scavenging HTS. *J. Med. Chem.* **2015**, *58*, 2091−2113.
(28) Naeini, M. P.; Cooper, G. F.; Hauskrecht, M. In *Obtaining well Calibrated Probabilities Using Bayesian Binning*, Proceedings of the... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence 2015; pp 2901.
(29) Niculescu-Mizil, A.; Caruana, R. In *Predicting Good Probabilities with Supervised Learning*, Proceedings of the 22nd International Conference on Machine Learning, 2005; pp 625−632.
(30) Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929−1958.
(31) Pereyra, G.; Tucker, G.; Chorowski, J.; Kaiser, Ł.; Hinton, G. Regularizing neural networks by penalizing confident output distributions 2017. arXiv:1701.06548. arXiv.org e-Print archive. https://arxiv.org/abs/1701.06548.
(32) Gal, Y.; Ghahramani, Z. In *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*, International Conference on Machine Learning, 2016; pp 1050−1059.
(33) Wilson, A. G.; Izmailov, P. Bayesian Deep Learning and a Probabilistic Perspective of Generalization 2020. arXiv:2002.08791. arXiv.org e-Print archive. https://arxiv.org/abs/2002.08791.
(34) Maddox, W. J.; Izmailov, P.; Garipov, T.; Vetrov, D. P.; Wilson, A. G. In *A Simple Baseline for Bayesian Uncertainty in Deep Learning*, Advances in Neural Information Processing Systems, 2019; pp 13132−13143.
(35) Osawa, K.; Swaroop, S.; Khan, M. E. E.; Jain, A.; Eschenhagen, R.; Turner, R. E.; Yokota, R. In *Practical Deep Learning with Bayesian Principles*, Advances in Neural Information Processing Systems, 2019; pp 4289−4301.
(36) Lakshminarayanan, B.; Pritzel, A.; Blundell, C. In *Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles*, Advances in Neural Information Processing Systems, 2017; pp 6402−6413.
(37) Welling, M.; Teh, Y. W. In *Bayesian Learning via sTochastic Gradient Langevin Dynamics*, Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011; pp 681−688.
(38) Izmailov, P.; Podoprikhin, D.; Garipov, T.; Vetrov, D.; Wilson, A. G. Averaging weights leads to wider optima and better generalization 2018. arXiv:1803.05407. arXiv.org e-Print archive. https://arxiv.org/abs/1803.05407.
(39) Irwin, J. J.; Shoichet, B. K. ZINC- a free database of commercially available compounds for virtual screening. *J. Chem. Inf. Model.* **2005**, *45*, 177−182.
(40) Ryu, S.; Kwon, Y.; Kim, W. Y. A Bayesian graph convolutional network for reliable prediction of molecular properties with uncertainty quantification. *Chem. Sci.* **2019**, *10*, 8438−8446.
(41) Zhang, Y.; et al. Bayesian semi-supervised learning for uncertainty-calibrated prediction of molecular properties and active learning. *Chem. Sci.* **2019**, *10*, 8154−8163.
(42) Noh, J.; Gu, G. H.; Kim, S.; Jung, Y. Uncertainty-Quantified Hybrid Machine Learning/Density Functional Theory High Throughput Screening Method for Crystals. *J. Chem. Inf. Model.* **2020**, *60*, 1996−2003.
(43) Scalia, G.; Grambow, C. A.; Pernici, B.; Li, Y.-P.; Green, W. H. Evaluating Scalable Uncertainty Estimation Methods for Deep Learning-Based Molecular Property Prediction. *J. Chem. Inf. Model.* **2020**, *60*, 2697−2717.
(44) Hirschfeld, L.; Swanson, K.; Yang, K.; Barzilay, R.; Coley, C. W. Uncertainty Quantification Using Neural Networks for Molecular Property Prediction 2020, arXiv:2005.10036. arXiv.org e-Print archive. https://arxiv.org/abs/2005.10036.
(45) Kwon, Y.; Won, J.-H.; Kim, B. J.; Paik, M. C. Uncertainty quantification using Bayesian neural networks in classification: Application to biomedical image segmentation. *Comput. Stat. Data Anal.* **2020**, *142*, No. 106816.

https://dx.doi.org/10.1021/acs.jcim.0c00416
J. Chem. Inf. Model. 2020, 60, 5936−5945