Chemical Science

EDGE ARTICLE

View Article Online
View Journal | View Issue

Check for updates

Cite this: *Chem. Sci.*, 2019, **10**, 8438

All publication charges for this article have been paid for by the Royal Society of Chemistry

A Bayesian graph convolutional network for reliable prediction of molecular properties with uncertainty quantification†

Seongok Ryu,[a] Yongchan Kwon [b] and Woo Youn Kim *[ac]

Deep neural networks have been increasingly used in various chemical fields. In the nature of a data-driven approach, their performance strongly depends on data used in training. Therefore, models developed in data-deficient situations can cause highly uncertain predictions, leading to vulnerable decision making. Here, we show that Bayesian inference enables more reliable prediction with quantitative uncertainty analysis. Decomposition of the predictive uncertainty into model- and data-driven uncertainties allows us to elucidate the source of errors for further improvements. For molecular applications, we devised a Bayesian graph convolutional network (GCN) and evaluated its performance for molecular property predictions. Our study on the classification problem of bio-activity and toxicity shows that the confidence of prediction can be quantified in terms of the predictive uncertainty, leading to more accurate virtual screening of drug candidates than standard GCNs. The result of log *P* prediction illustrates that data noise affects the data-driven uncertainty more significantly than the model-driven one. Based on this finding, we could identify artefacts that arose from quantum mechanical calculations in the Harvard Clean Energy Project dataset. Consequently, the Bayesian GCN is critical for molecular applications under data-deficient conditions.

Received 22nd April 2019
Accepted 21st July 2019

DOI: 10.1039/c9sc01992h

rsc.li/chemical-science

Open Access Article. Published on 22 juillet 2019. Downloaded on 20/07/2024 11:15:44.
This article is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported Licence.

1 Introduction

The rise of deep learning has a huge impact on diverse fields, such as computer vision and natural language understanding. Chemistry is not an exception. State-of-the-art deep neural networks (DNNs) have been applied to various problems in chemistry including high-throughput screening for drug discovery,[1–4] *de novo* molecular design[5–12] and planning chemical reactions.[13–15] They show comparable to or sometimes better performance than principle-based approaches in predicting several molecular properties.[16–20] Such a result can be achieved only if a vast amount of well-qualified data is obtained, because the performance of the data-driven approach strongly depends on training data.

Unfortunately, however, many real world applications suffer from a lack of qualified data. For example, Feinberg *et al.* showed that more qualified data should be provided to improve the prediction accuracy on drug–target interactions, which is a key step for drug discovery.[21] The number of ligand–protein complex samples in the PDBbind database[22] is only about 15 000. The number of toxic samples in the Tox21 dataset is less than 10 000.[3] Expensive and time-consuming experiments are inevitable to acquire more qualified data. Like the Harvard Clean Energy Project dataset,[23] synthetic data from computations can be used as an alternative but often include unintentional errors caused by the approximation methods employed. In addition, data-inherent bias and noise hurt the quality of data. Tox21 [3] and DUD-E datasets[24] are such examples. There are far more negative samples than positive ones. Of various toxic types, the lowest percentage of positive samples is 2.9% and the highest is 15.5%. The DUD-E dataset is highly unbalanced in that the number of decoy samples is almost 50 times larger than that of active samples.

In the nature of a data-driven approach, a lack of qualified data can cause severe damage to the reliability of the prediction results of DNNs. This reliability issue should be taken more seriously when models are obtained by point estimation-based methods such as maximum-a-posteriori (MAP) or maximum likelihood (ML) estimation. It is because both estimation methods result in a single deterministic model which can produce unreliable outcomes for new data. In Fig. 1, we exemplify a drawback of using deterministic models for a classification problem with a small dataset. A small amount of data inevitably leads to a number of decision boundaries, which corresponds to a distribution of models, and the MAP (or ML)

[a]*Department of Chemistry, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea. E-mail: wooyoun@kaist.ac.kr*

[b]*Department of Statistics, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Republic of Korea*

[c]*KI for Artificial Intelligence, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea*

† Electronic supplementary information (ESI) available. See DOI: 10.1039/c9sc01992h

This journal is © The Royal Society of Chemistry 2019

Open Access Article. Published on 22 juillet 2019. Downloaded on 20/07/2024 11:15:44.
This article is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported Licence.

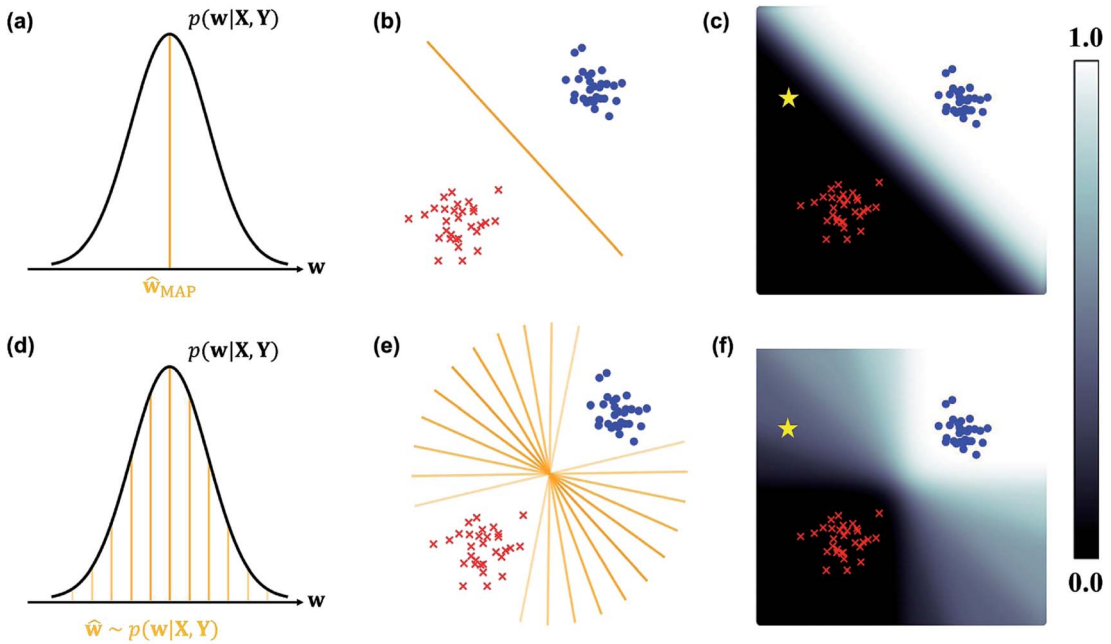

Fig. 1 A simple linearly separable binary classification problem. Positive and negative training data samples are denoted with blue and red markers, respectively. (a) A model estimated by MAP, $\hat{\mathbf{w}}_{MAP}$, corresponds to the $\mathbf{w}$ value of the orange line, and (b) the decision boundary in the two-dimensional space is denoted by the orange line. (c) Output probability values (eqn (3)) are colored in the background. The orange lines with different transparency in (d) are models drawn from the posterior $p(\mathbf{w}|\mathbf{X}, \mathbf{Y})$, and the lines in (e) are the corresponding decision boundaries. (f) Predictive probabilities obtained with Bayesian inference (eqn (4)) are colored in the background. The yellow star in (c) and (f) is a new unlabeled sample.

estimation selects only one from the distribution as shown in Fig. 1(a) and (b). In addition, the magnitude of output values is often erroneously interpreted as the confidence of prediction, and thus higher values are usually believed to be closer to the true value. That being said, relying on predicted outputs to make decisions can produce unreliable results for a new sample located far away from the distribution of training data. We illustrate an example of vulnerable decision making in Fig. 1(c). On one hand, the sample denoted by the yellow star will be predicted to belong to the red sample with nearly zero output probability according to the decision boundary estimated by the MAP. On the other hand, such a decision can be reversed by another possible decision boundary with the same accuracy for the given training data. As such, deterministic models can lead to catastrophic decisions in real-life applications, such as autonomous vehicle and medical fields, that put emphasis on so-called AI-safety problems.[25–27]

Collecting large amounts of data is one definite way to overcome the aforementioned problem but is usually expensive, time-consuming and laborious. Instead, Bayesian inference of model parameters and outputs enables more informative decision making by considering all possible outcomes predicted from the distribution of decision boundaries. In Fig. 1(d)–(f), we describe how to classify the yellow star according to Bayesian inference. Since various model parameters sampled from the posterior distribution will give different answers, the final outcome is obtained by averaging those answers. In addition, uncertainty quantification of prediction results is feasible thanks to the probabilistic nature of Bayesian inference. Kendall and Gal performed quantitative uncertainty analysis on computer vision problems by using DNNs grounded on a Bayesian framework.[28] In particular, they have shown that the uncertainty of predictions can be decomposed into model- and data-driven uncertainties, which helps to identify the sources of prediction errors and further to improve both data and models.[29] It has been known that results from Bayesian inference become identical to those of MAP estimation in the presence of a sufficiently large amount of data.[30] However, as long as the amount of data is not enough like in most real-life applications, Bayesian inference would be more relevant.

In this work, we show that Bayesian inference is more informative in making reliable predictions than the standard ML estimation method. As a practical approach to obtain a distribution of model parameters and the corresponding outputs, we propose to exploit Bayesian neural networks. Since graph representation of molecular structures has been widely used, we chose molecular graphs as inputs for our model and implemented a graph convolutional network (GCN)[31–33] within the Bayesian framework[28,34] for the end-to-end learning of representations and predicting molecular properties.

The resulting Bayesian GCN is applied to the following four examples. In binary classification of bio-activity and toxicity, we show that prediction with a lower uncertainty turned out to be more accurate, which indicates that predictive uncertainty can be regarded as the confidence of prediction. Based on this finding, we carried out a virtual screening of drug candidates and found more known active molecules when using the Bayesian GCN than when using the same GCN model but

Open Access Article. Published on 22 juillet 2019. Downloaded on 20/07/2024 11:15:44.
This article is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported Licence.

estimating by the ML. The third example demonstrates that the uncertainty quantification enables us to separately analyze data-driven and model-driven uncertainties. Finally, we could identify artefacts in the synthetic power conversion efficiency values of molecules in the Harvard Clean Energy Project dataset.[23] We verified that molecules with conspicuously large data-driven uncertainties were incorrectly annotated because of inaccurate approximations. Our results show that more reliable predictions can be achieved using Bayesian neural networks followed by uncertainty analysis.

2 Theoretical background

This section aims to explain the theoretical background of Bayesian neural networks. We first brief about Bayesian inference of model parameters and output to elaborate on our motivation of this research. Then, we briefly discuss variational inference as a practical approximation for implementation of Bayesian neural networks. Lastly, we explain a uncertainty quantification method based on Bayesian inference.

2.1 Bayesian inference of model parameters and output

Training a DNN is a procedure to obtain model parameters that best explain a given dataset. The Bayesian framework underlines that it is impossible to estimate a single deterministic model parameter, and hence one needs to infer the distribution of model parameters. For a given training set $\{\mathbf{X}, \mathbf{Y}\}$, let $p(\mathbf{Y}|\mathbf{X}, \mathbf{w})$ and $p(\mathbf{w})$ be a model likelihood and a prior distribution for a parameter $\mathbf{w} \in \Omega$, respectively. Following Bayes' theorem, a posterior distribution, which corresponds to the conditional distribution of model parameters given the training dataset, is defined as

$$p(\mathbf{w}|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{p(\mathbf{Y}|\mathbf{X})}. \quad (1)$$

By using eqn (1), two different approaches have been derived: (i) MAP-estimation‡ finds the mode of the posterior and (ii) Bayesian inference computes the posterior distribution itself. The MAP estimated model $\hat{\mathbf{w}}_{\mathrm{MAP}}$ is given by

$$\hat{\mathbf{w}}_{\mathrm{MAP}} = \underset{\mathbf{w}\in\Omega}{\operatorname{argmax}} p(\mathbf{w}|\mathbf{X}, \mathbf{Y}), \quad (2)$$

which is illustrated by the orange line in Fig. 1(a). Then, the expectation of output $\mathbf{y}^*$ for a new input $\mathbf{x}^*$ is given by

$$\hat{E}(\mathbf{y}^*|\mathbf{x}^*, \mathbf{X}, \mathbf{Y}) = f^{\hat{\mathbf{w}}_{\mathrm{MAP}}}(\mathbf{x}^*), \quad (3)$$

where is $f^{\hat{\mathbf{w}}_{\mathrm{MAP}}}(\cdot)$ a function parameterized with $\hat{\mathbf{w}}_{\mathrm{MAP}}$. For instance, the orange line in Fig. 1(b) denotes the decision boundary, $f^{\hat{\mathbf{w}}_{\mathrm{MAP}}}(\cdot) = 0.5$, in a simple linearly separable binary classification problem. The background color in Fig. 1(c) represents the output probability that a queried sample has a positive label (blue circle). Note that the right-hand-side term in eqn (3) does not have any conditional dependence on the training set $\{\mathbf{X}, \mathbf{Y}\}$.

In contrast to the MAP estimation, the Bayesian inference of outputs is given by the predictive distribution as follows:

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{X}, \mathbf{Y}) = \int_{\Omega} p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{w})p(\mathbf{w}|\mathbf{X}, \mathbf{Y})d\mathbf{w}. \quad (4)$$

This formula allows more reliable predictions by the following two factors. First, the final outcome is inferred by integrating all possible models and their outputs. Second, it is possible to quantify the uncertainty of the predicted results. Fig. 1(d)–(f) illustrate the posterior distribution, sampled decision boundaries, and the resultant output probabilities, respectively. The new input denoted by the yellow star in Fig. 1(f) can be labeled differently according to the sampled model. Since the input is far away from the given training set, it is inherently difficult to assign a correct label without further information. As a result, the output probability is substantially low, and a large uncertainty of the prediction arises, as indicated by the gray color which is in contrast to the dark black color in Fig. 1(c). This conceptual example demonstrates the importance of the Bayesian framework especially in a limited data environment.

2.2 Variational inference in Bayesian neural networks

Direct incorporation of eqn (4) is intractable for DNN models because of heavy computational costs in the integration over the whole parameter space Ω. Diverse approximation methods have been proposed to mitigate this problem.[35] We adopted a variational inference method which approximates the posterior distribution with a tractable distribution $q_\theta(\mathbf{w})$ parameterized by a variational parameter θ.[36,37] Minimizing the Kullback–Leibler divergence,

$$\mathrm{KL}(q_\theta(\mathbf{w})\|p(\mathbf{w}|\mathbf{X}, \mathbf{Y})) = \int_{\Omega} q_\theta(\mathbf{w})\log\frac{q_\theta(\mathbf{w})}{p(\mathbf{w}|\mathbf{X}, \mathbf{Y})}d\mathbf{w}, \quad (5)$$

makes the two distributions similar to one another in principle. We can replace the intractable posterior distribution in eqn (5) with $p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})$ by following Bayes' theorem in eqn (1). Then, our minimization objective, namely the negative evidence lower-bound, becomes

$$\mathscr{L}_{\mathrm{VI}}(\theta) = -\int_{\Omega} q_\theta(\mathbf{w})\log p(\mathbf{Y}|\mathbf{X}, \mathbf{w})d\mathbf{w} + \mathrm{KL}(q_\theta(\mathbf{w})\|p(\mathbf{w})). \quad (6)$$

For implementation, the variational distribution $q_\theta(\mathbf{w})$ should be chosen carefully. Blundell *et al.* proposed to use a product of Gaussian distributions for the variational distribution $q_\theta(\mathbf{w})$. In addition, a multiplicative normalizing flow[38] can be applied to increase the expressive power of variational distribution. However, these two approaches require a large number of weight parameters. The Monte-Carlo dropout (MC-dropout) approximates the posterior distribution by a product of the Bernoulli distribution,[39] the so-called dropout[40] variational distribution. The MC-dropout is practical in that it does not need extra learnable parameters to model the variational posterior distribution, and the integration over the whole parameter space can be easily approximated with the summation of models sampled using a Monte-Carlo estimator.[25,39]

In practice, optimizing Bayesian neural networks with the MC-dropout, the so-called MC-dropout networks, is technically

 This journal is © The Royal Society of Chemistry 2019

Open Access Article. Published on 22 juillet 2019. Downloaded on 20/07/2024 11:15:44.
This article is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported Licence.

equivalent to that of standard neural networks with dropout as regularization. Hence, the training time for the MC-dropout networks is comparable to that for standard neural networks, which enables us to develop Bayesian neural networks with high scalability. In contrast to standard neural networks that predict outputs by turning-off the dropout at the inference phase, the MC-dropout networks keep turning on the dropout and predict outputs by sampling and averaging them, which theoretically corresponds to integrating the posterior distribution and likelihood.[25] This technical simplicity provides an efficient way of Bayesian inference with neural networks. On the other hand, approximated posteriors implemented by the dropout variational inference often show inaccurate results, and several studies have reported the drawbacks of the MC-dropout networks.[38,41,42] In this work, we focus on the practical advantages of the MC-dropout networks and introduce the Bayesian inference of molecular properties with graph convolutional networks.

2.3 Uncertainty quantification with a Bayesian neural network

A variational inference with an approximated variational distribution $q_\theta(\mathbf{w})$ provides the (variational) predictive distribution of a new output $\mathbf{y}^*$ given a new input $\mathbf{x}^*$ as

$$q_\theta^*(\mathbf{y}^*|\mathbf{x}^*) = \int_\Omega q_\theta(\mathbf{w}) p(\mathbf{y}^*|f^{\mathbf{w}}(\mathbf{x}^*)) d\mathbf{w}, \tag{7}$$

where $f^{\mathbf{w}}(\mathbf{x}^*)$ is a model output with a given $\mathbf{w}$. For regression tasks, a predictive mean of this distribution with T times MC sampling is estimated as

$$\hat{E}[\mathbf{y}^*|\mathbf{x}^*] = \frac{1}{T}\sum_{t=1}^{T} f^{\hat{\mathbf{w}}_t}(\mathbf{x}^*), \tag{8}$$

and a predictive variance is estimated as

$$\widehat{\mathrm{Var}}[\mathbf{y}^*|\mathbf{x}^*] = \sigma^2 I + \frac{1}{T}\sum_{t=1}^{T} f^{\hat{\mathbf{w}}_t}(\mathbf{x}^*)^T f^{\hat{\mathbf{w}}_t}(\mathbf{x}^*) - \hat{E}[\mathbf{y}^*|\mathbf{x}^*]^T \hat{E}[\mathbf{y}^*|\mathbf{x}^*] \tag{9}$$

with $\hat{\mathbf{w}}_t$ drawn from $q_\theta(\mathbf{w})$ at the sampling step t and an assumption $p(\mathbf{y}^*|f^{\mathbf{w}}(\mathbf{x}^*)) = N(\mathbf{y}^*; f^{\mathbf{w}}(\mathbf{x}^*), \sigma^2 I)$. Here, the model assumes homoscedasticity with a known quantity, meaning that every data point gives a distribution with the same variance σ^2. Further, obtaining the distributions with different variances allows deduction of a heteroscedastic uncertainty. Assuming the heteroscedasticity, the output given the t-th sample $\hat{\mathbf{w}}_t$ is

$$\left[\hat{\mathbf{y}}_t^*, \hat{\sigma}_t\right] = f^{\hat{\mathbf{w}}_t}(\mathbf{x}^*). \tag{10}$$

Then, the heteroscedastic predictive uncertainty is given by eqn (11), which can be partitioned into two different uncertainties: aleatoric and epistemic uncertainties.

$$\widehat{\mathrm{Var}}[\mathbf{y}^*|\mathbf{x}^*] = \underbrace{\frac{1}{T}\sum_{t=1}^{T}\left(\hat{\mathbf{y}}_t^*\right)^2 - \left(\frac{1}{T}\sum_{t=1}^{T}\hat{\mathbf{y}}_t^*\right)^2}_{\text{epistemic}} + \underbrace{\frac{1}{T}\sum_{t=1}^{T}\hat{\sigma}_t{}^2}_{\text{aleatoric}}. \tag{11}$$

The aleatoric uncertainty arises from data inherent noise, while the epistemic uncertainty is related to the model incompleteness.[43] Note that the latter can be reduced by increasing the amount of training data, because it comes from an insufficient amount of data as well as the use of an inappropriate model.

In classification problems, Kwon *et al.* proposed a natural way to quantify the aleatoric and epistemic uncertainties as follows.

$$\widehat{\mathrm{Var}}[\mathbf{y}^*|\mathbf{x}^*] = \underbrace{\frac{1}{T}\sum_{t=1}^{T}\left(\hat{\mathbf{y}}_t^* - \bar{\mathbf{y}}\right)\left(\hat{\mathbf{y}}_t^* - \bar{\mathbf{y}}\right)^T}_{\text{epistemic}} + \underbrace{\frac{1}{T}\sum_{t=1}^{T}\left(\mathrm{diag}\left(\hat{\mathbf{y}}_t^*\right) - \left(\hat{\mathbf{y}}_t^*\right)\left(\hat{\mathbf{y}}_t^*\right)^T\right)}_{\text{aleatoric}}, \tag{12}$$

where $\bar{\mathbf{y}} = \sum_{t=1}^{T}\hat{\mathbf{y}}_t^*/T$ and $\hat{\mathbf{y}}_t^* = \mathrm{softmax}(\mathbf{f}^{\hat{\mathbf{w}}_t}(\mathbf{x}^*))$. While Kendall and Gal's method requires extra parameters $\hat{\sigma}_t$ at the last hidden layer and often causes unstable parameter updates in a training phase,[28] the method proposed by Kwon *et al.* has advantages in that models do not need the extra parameters.[34] Eqn (12) also utilizes a functional relationship between the mean and variance of multinomial random variables.

3 Methods

For predicting molecular properties, we adopt molecular graphs as input and the GCN augmented with attention and the gated mechanism suggested by Ryu *et al.*[33] As illustrated in Fig. 2, the Bayesian GCN used in this work consists of the following three parts:

- Three augmented graph convolution layers update node features. The number of self-attention heads is four. The dimension of output from each layer is (75 × 32).
- A readout function produces a graph feature whose dimension is 256.
- A feed-forward MLP, which is composed of two fully connected layers, outputs a molecular property. The hidden dimension of each fully connected layer is 256.

In order to approximate the posterior distribution with a dropout variational distribution, we applied dropouts at every hidden layer. We did not use the standard dropout with a hand-tuned dropout rate but used Concrete dropout[44] to develop as accurate Bayesian models as possible. By using the Concrete dropout, we can obtain the optimal dropout rate for individual hidden layers by gradient descent optimization. We used Gaussian priors $\mathcal{N}(0, l^2)$ with a length scale of $l = 10^{-4}$ for all model parameters. In the training phase, we used the Adam optimizer[45] with an initial learning rate of 10^{-3}, and the learning rate decayed by half every 10 epochs. The number of total training epochs is 100, and the batch size is 100. We randomly split each dataset in the ratio of 0.72 : 0.08 : 0.2 for training, validation and testing. For all experiments, we kept turning on the dropout at the inference phases and sampled outputs with $T = 20$ (in eqn (8), (9) and (12)) and averaged them in order to perform Bayesian inference. We used one GTX-1080

Open Access Article. Published on 22 juillet 2019. Downloaded on 20/07/2024 11:15:44.
This article is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported Licence.

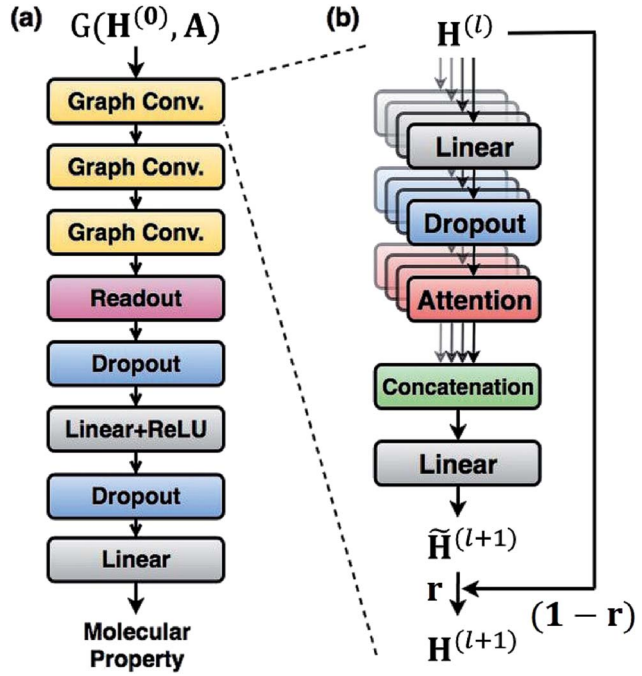

Fig. 2 The architecture of the Bayesian GCN used in this work. (a) The entire model is composed of three augmented graph convolutional layers, readout layers and two linear layers and takes inputs as a molecular graph $G(H^{(0)}, A)$, where H^0 is a node feature and A is an adjacency matrix. (b) Details of each graph convolution layer augmented with attention and gate mechanisms. The l-th graph convolutional layer updates node features and produces $H^{(l+1)}$.

Ti processor for performing all experiments. We provide the number of samples used for training/validation/testing, training time, and accuracy curves for all experiments in the ESI.† The code used for the experiments is available at https://github.com/seongokryu/uq_molecule.

4 Results and discussion

4.1 Relationship between the uncertainty and output probability: bio-activity and toxicity classification

In classification problems, the output probability itself tends to be regarded as the confidence of prediction. For example, in a virtual screening of drug candidates, molecules predicted to be active with high output probability are preferred. However, as Gal and Ghahramani pointed out, such interpretation is erroneous for deterministic models.[39] Fig. 1(c) shows such an example. Indeed, though the MAP-estimated model can give a high output probability to a sample located far away from the distribution of training data, it is difficult to determine its correct label due to lack of information. In contrast, Bayesian inference allows us to obtain predictive uncertainty as well as output probabilities. In the case of the yellow star, the Bayesian inference gives a low output probability with high predictive uncertainty as expected. With two biological classification problems having a limited amount of data, we here show that the higher the output probability of the Bayesian GCN is, the lower the predictive uncertainty and hence predictive uncertainty can be regarded as the confidence of prediction.

We trained the Bayesian GCN with 25 627 molecules which are annotated with EGFR inhibitory activity in the DUD-E dataset. Fig. 3 shows the relationship between predictive uncertainty and output probability for 7118 molecules in the test set. The total uncertainty as well as the aleatoric and epistemic uncertainties are minimum at both highest and lowest output probabilities, while they are maximum at the center. Therefore, one can make a confident decision by taking the highest or lowest output probabilities; however it should be emphasized again that this is not the case for the MAP- or ML-estimated models.

Based on this finding, uncertainty calibrated decision making can lead to high accuracy in classification problems. To verify this, we trained the Bayesian GCNs with bio-activity labels for various target proteins in the DUD-E dataset and toxicity labels in the Tox21 dataset. Then, we sorted the molecules in increasing order of uncertainty and divided them into five groups as follows: molecules in the i-th group have total uncertainties in the range of $[(i-1)\times 0.1, i\times 0.1]$. Fig. 4(a) and (b) show the accuracy of each group for five different bio-activities in the DUD-E dataset and five different toxicities in the Tox21 dataset, respectively. For all cases, the first group having the lowest uncertainty showed the highest accuracy. This result manifests that the uncertainty values can be used as a confidence indicator.

4.2 Virtual screening of EGFR inhibitors in the ChEMBL dataset

We have shown that confident predictions of molecular properties have become feasible thanks to the relationship between the output probability and predictive uncertainty within the Bayesian framework. Here, we examine whether such an uncertainty-calibrated prediction can lead to higher accuracy in real-life applications than a maximum likelihood (ML) and a maximum-a-posteriori (MAP) estimation approach. To this end, we applied the previous Bayesian GCN trained with the DUD-E dataset to the virtual screening of EGFR inhibitors in the ChEMBL dataset.[46] We deliberately used two completely different datasets for training and testing so as to evaluate the generalization ability of the model.

Molecules in the ChEMBL dataset were annotated with an experimental half maximal inhibitory concentration (IC50) value. To utilize this dataset for a classification problem, we assigned molecules with IC50 values above 6.0 as ground truth active, while the others were assigned as ground truth inactive. We compare three GCN models obtained by three different estimation methods: (i) ML, (ii) MAP, and (iii) Bayesian. We turned off the dropout masks and did not use MC-sampling at the inference phase to obtain the MAP-estimated GCN. Also, we obtained the ML-estimated GCN with the same training configurations except the dropout and L2-regularization. Then, we applied the three models to the virtual screening of the ChEMBL dataset.

Table 1 summarizes the screening results of the three models in terms of accuracy, area under receiver operating curve (AUROC), precision, recall and F1-score. The Bayesian GCN outperformed the point-estimated GCNs for all evaluation

Open Access Article. Published on 22 juillet 2019. Downloaded on 20/07/2024 11:15:44.
This article is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported Licence.

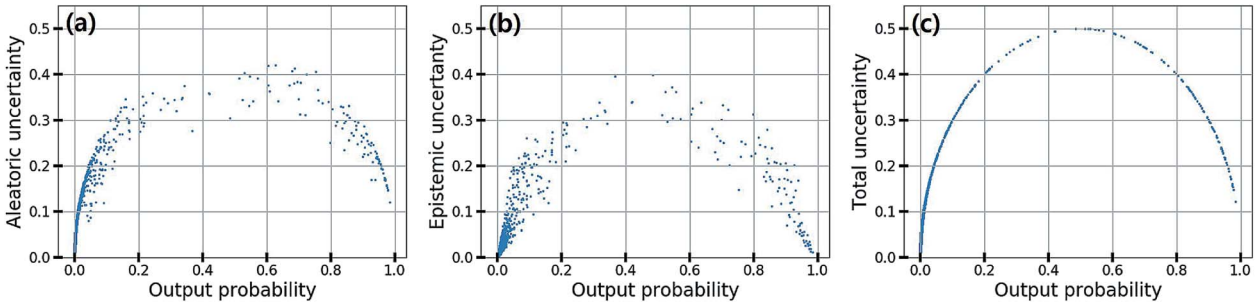

Fig. 3 (a) Aleatoric, (b) epistemic and (c) total uncertainty with respect to the output probability in the classification of EGFR inhibitory activity.

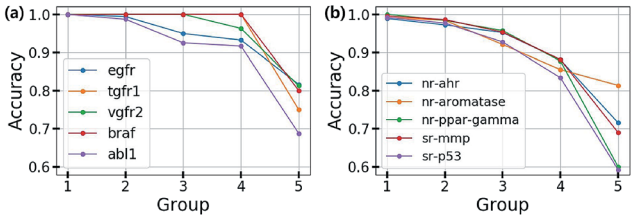

Fig. 4 Test accuracy for the classifications of (a) bio-activities against the five target proteins in the DUD-E dataset and (b) the five toxic effects in the Tox21 dataset.

Table 1 Performance of the GCN models obtained by different estimation methods in predicting the EGFR-inhibitory activity of molecules in the ChEMBL dataset

	ML	MAP	Bayesian
Accuracy	0.728	0.739	**0.752**
AUROC	0.756	0.781	**0.785**
Precision	0.714	0.68	**0.746**
Recall	0.886	**0.939**	0.868
F1-score	0.791	0.789	**0.803**

metrics except the recall. Since Bayesian inference assumes a model prior which corresponds to the regularization term in the training procedure, the Bayesian GCN showed better generalization ability and performance than the ML-estimated GCN as it was applied to the unseen dataset.[36] In contrast to the MAP-estimated GCN, whose model parameter (or decision boundary) is point-estimated, the Bayesian GCN infers predictive probability by MC-sampling of outputs with different dropout masks. This inference procedure allows the model to predict outputs by considering a multiple number of decision boundaries and shows better performance in the virtual screening experiment.

In Fig. 5, we visualize the distribution of output probability by dividing it into true positive, false positive, true negative and false negative groups. The output probability values of the ML-estimated GCN is close to 0.0 or 1.0 for most molecules, which is commonly referred to as over-confident prediction. Because of the regularization effect, the MAP-estimated GCN shows less over-confident results than the ML-estimated GCN. On the other hand, the outputs of the Bayesian GCN are distributed continuously from 0.0 to 1.0. This result is consistent with the previous conclusion that the Bayesian GCN predicts a value between 0.0 and 1.0 according to the extent of the predictive uncertainty for a given sample.

As demonstrated in the previous section, with Bayesian inference, an output probability value closer to one is expected more likely to be a true active label. This allows output probability to be used as a criterion for screening of desirable molecules. Table 2 shows the number of actives existing in each list of the top 100, 200, 300 and 500 molecules in terms of output probability. The Bayesian GCN mined remarkably more active molecules than the ML-estimated GCN did. In particular, it performed better in the top 100 and 200, which is critical for efficient virtual screening purposes with a small amount of qualified data. Also, it performed slightly better than the MAP-estimated GCN for all trials.

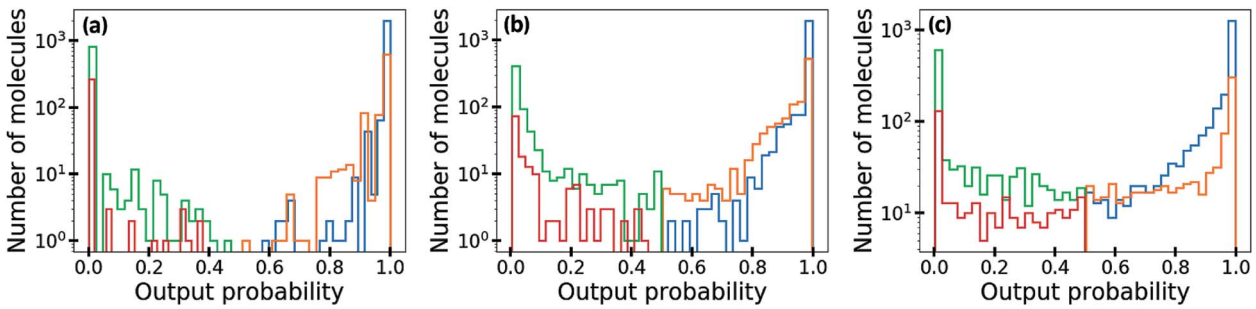

Fig. 5 Distributions of output probability obtained by (a) the ML, (b) the MAP and (c) the Bayesian GCNs. The total distribution is divided into true positive (blue), false positive (orange), true negative (green) and false negative (red) groups. Note that the *y*-axis is represented with a log scale.

Open Access Article. Published on 22 juillet 2019. Downloaded on 20/07/2024 11:15:44.
This article is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported Licence.

Table 2 The number of actives existing in the top N molecules that are sorted in increasing order of output probability

Top N	ML	MAP	Bayesian
100	29	57	**69**
200	67	130	**140**
300	139	202	**214**
500	277	346	**368**

4.3 Implication of data quality on aleatoric and epistemic uncertainties

In this experiment, we investigated how data quality affects predictive uncertainty. In particular, we analyzed the aleatoric and epistemic uncertainties separately in molecular property predictions using the Bayesian GCN. We chose log P prediction as an example because we can obtain a sufficient amount of logP values by using a deterministic formulation in the RDKit.[47] We assumed that these log P values do not include noise (stochasticity) and let them be ground truth labels. In order to control the data quality, we adjusted the extent of noise by adding a random Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Then, we trained the model with 97 287 samples and analyzed uncertainties of each predicted log P for 27 023 samples.

Fig. 6 shows the distribution of the three uncertainties with respect to the amount of additive noise σ^2. As the noise level increases, the aleatoric and total uncertainties increase, but the epistemic uncertainty is slightly changed. This result verifies that the aleatoric uncertainty arises from data inherent noises, while the epistemic uncertainty does not depend on data quality. Theoretically, the epistemic uncertainty should not be increased by the changes in the amount of data noise. Presumably, stochasticity in the numerical optimization of model parameters induced the slight change of the epistemic uncertainty.

4.4 Evaluating the quality of synthetic data based on uncertainty analysis

Based on the analysis of the previous experiment, we attempted to see whether uncertainty quantification can be used to evaluate the quality of existing chemical data.

Synthetic PCE values in the CEP dataset[23] were obtained from the Scharber model with statistical approximations.[48] In this procedure, unintentional errors can be included in the resulting synthetic data. Therefore, this example would be a good exercise problem to evaluate the quality of data through the analysis of aleatoric uncertainty. We used the same dataset of Duvenaud *et al.*§ for training and testing.

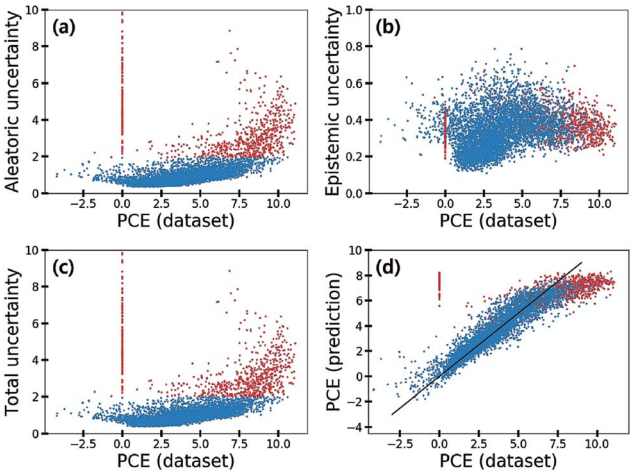

Fig. 7 (a) Aleatoric, (b) epistemic, and (c) total uncertainties and (d) predicted PCE value against the PCE value in the dataset. The samples colored in red show a total uncertainty greater than two.

Fig. 7 shows the scatter plot of three uncertainties in the CEP predictions for 5995 molecules in the test set. Samples with a total uncertainty greater than two are highlighted with red color. Some samples with large PCE values above eight had relatively large total uncertainties. Their PCE values deviated considerably from the black line in Fig. 7(d). Notably most molecules with a zero PCE value had large total uncertainties as well. These large uncertainties came from the aleatoric uncertainty as depicted in Fig. 7(a), indicating that the data quality of these particular samples is relatively poor. Hence, we speculated that data inherent noises might cause large prediction errors.

To elaborate the origin of such errors, we investigated the procedure of obtaining the PCE values. The Harvard Organic Photovoltaic Dataset[49] contains both experimental and synthetic PCE values of 350 organic photovoltaic materials. The synthetic PCE values were computed according to eqn (13), which is the result of the Scharber model.[48]

$$\mathrm{PCE} \propto V_{\mathrm{OC}} \times \mathrm{FF} \times J_{\mathrm{SC}}, \tag{13}$$

where V_{OC} is the open circuit potential, FF is the fill factor, and J_{SC} is the short circuit current density. FF was set to 65%. V_{OC}

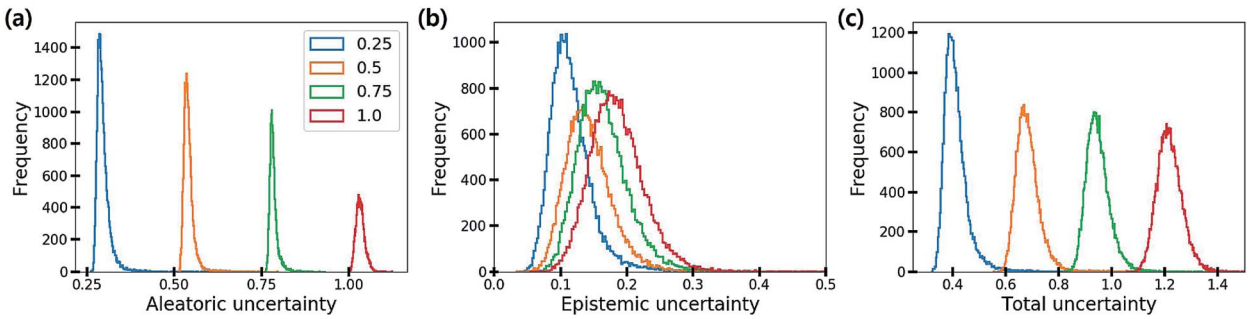

Fig. 6 Histograms of (a) aleatoric, (b) epistemic and (c) total uncertainties as the amount of additive noise σ^2 increases.

This journal is © The Royal Society of Chemistry 2019

Open Access Article. Published on 22 juillet 2019. Downloaded on 20/07/2024 11:15:44.
This article is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported Licence.

and J_{SC} were obtained from electronic structure calculations of molecules.[23] We found that the J_{SC} of some molecules were zero or nearly zero, which might be from the artefact of quantum mechanical calculations. In particular, in contrast to their non-zero experimental PCE values, J_{SC} and PCE values computed by using the M06-2X functional[50] were almost zero consistently. Pyzer-Knapp *et al.* pointed out this problem and proposed a statistical calibration method that can successfully correct the biased results.[51]

To summarize, we suspect that quantum mechanical artefacts caused a significant drop of data quality, resulting in the large aleatoric uncertainties as highlighted in Fig. 7. Consequently, we can identify data inherent noise by analyzing aleatoric uncertainty.

5 Conclusion

Deep neural networks (DNNs) have shown promising success in the prediction of molecular properties as long as a large amount of data is available. However, a lack of qualified data in many chemical problems discourages employing them directly due to the nature of a data-driven approach. In particular, deterministic models, which can be derived from maximum likelihood (ML) or maximum-a-posteriori (MAP) estimation methods, may cause vulnerable decision making in real-life applications where reliable predictions are very important.

Here, we have studied the possibility of reliable predictions and decision making in such cases with the Bayesian GCN. Our results show that output probability from the Bayesian GCN can be regarded as the confidence of prediction in classification problems, which is not the case for the ML- or MAP-estimated models. Moreover, we demonstrated that such a confident prediction can lead to notably higher accuracy for a virtual screening of drug candidates than a standard approach based on the ML-estimation. In addition, we showed that uncertainty analysis enabled by Bayesian inference can be used to evaluate data quality in a quantitative manner and thus helps to find possible sources of errors. As an example, we could identify unexpected errors included in the Harvard Clean Energy Project dataset and their possible origin using the uncertainty analysis. Most chemical applications of deep learning have adopted DNN models estimated by either MAP or ML. Our study clearly shows that Bayesian inference is essential in limited data environments where AI-safety problems are critical.

Beyond reliable prediction of molecular properties along with uncertainty quantification, we expect that DNNs with the Bayesian perspective may be extended to data-efficient algorithms for molecular applications. One of the possible interesting future applications is to use Bayesian GCNs for high-throughput screening of chemical space with Bayesian optimization.[52] For this purpose, Bayesian optimization has been utilized as a promising tool to search for the most desirable candidates based on predictive uncertainty.[6,53–55] In chemistry, Hernández-Lobato *et al.* proposed a computationally efficient Bayesian optimization framework that was built on a Gaussian process with Morgan fingerprints as inputs for the estimation of predictive uncertainty.[55] Thus, we believe that our proposed method has potential for designing efficient high-throughput screening tools for drug or materials discovery.

Another important possible application of Bayesian GCNs is extension for active learning. Since acquiring big data from experiments is expensive and laborious, data-efficient learning algorithms are attracting attention as a viable solution in various real-life applications by enabling neural networks to be trained with a small amount of data.[56] Active learning, is one of such algorithms, employs an acquisition function suggesting new data points that should be added for further improvement of model accuracy. Incorporation of the Bayesian framework in the active learning helps to select new data points by providing fruitful information with predictive uncertainty.[29] In this regard, we believe that the present work offers insights into the development of a deep learning approach in a data-efficient way for various chemical problems, which hopefully promotes synergistic cooperation of deep learning with experiments.

Author contributions

S. R. and Y. K. conceived the idea. S. R. did the implementation and ran the simulation. All the authors analyzed the results and wrote the manuscript together.

Conflicts of interest

The authors declare no competing financial interests.

Acknowledgements

We would like to appreciate the anonymous reviewers for their constructive comments. This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (NRF-2017R1E1A1A01078109).

Notes and references

‡ We would like to note two things in the MAP estimation. First, eqn (2) can be computed by gradient descent optimization, which corresponds to the common training procedure of machine learning systems, minimizing a negative-log-likelihood term (a loss function) and a regularization term. Second, the MAP estimation becomes equivalent to the maximum likelihood estimation which maximizes the likelihood term only when we assume a uniform prior distribution.
§ https://github.com/HIPS/neural-fingerprint

1 J. Gomes, B. Ramsundar, E. N. Feinberg and V. S. Pande, 2017, arXiv preprint arXiv:1703.10603.
2 J. Jiménez, M. Skalic, G. Martínez-Rosell and G. De Fabritiis, *J. Chem. Inf. Model.*, 2018, **58**, 287–296.
3 A. Mayr, G. Klambauer, T. Unterthiner and S. Hochreiter, *Front. environ. sci.*, 2016, **3**, 80.
4 H. Öztürk, A. Özgür and E. Ozkirimli, *Bioinformatics*, 2018, **34**, i821–i829.
5 N. De Cao and T. Kipf, 2018, arXiv preprint arXiv:1805.11973.
6 R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla,

Open Access Article. Published on 22 juillet 2019. Downloaded on 20/07/2024 11:15:44.
This article is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported Licence.

J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams and A. Aspuru-Guzik, *ACS Cent. Sci.*, 2018, **4**, 268–276.
7 G. L. Guimaraes, B. Sanchez-Lengeling, C. Outeiral, P. L. C. Farias and A. Aspuru-Guzik, 2017, arXiv preprint arXiv:1705.10843.
8 W. Jin, R. Barzilay and T. Jaakkola, 2018, arXiv preprint arXiv:1802.04364.
9 M. J. Kusner, B. Paige and J. M. Hernández-Lobato, 2017, arXiv preprint arXiv:1703.01925.
10 Y. Li, O. Vinyals, C. Dyer, R. Pascanu and P. Battaglia, 2018, arXiv preprint arXiv:1803.03324.
11 M. H. Segler, T. Kogej, C. Tyrchan and M. P. Waller, *ACS Cent. Sci.*, 2017, **4**, 120–131.
12 J. You, B. Liu, R. Ying, V. Pande and J. Leskovec, 2018, arXiv preprint arXiv:1806.02473.
13 M. H. Segler, M. Preuss and M. P. Waller, *Nature*, 2018, **555**, 604.
14 J. N. Wei, D. Duvenaud and A. Aspuru-Guzik, *ACS Cent. Sci.*, 2016, **2**, 725–732.
15 Z. Zhou, X. Li and R. N. Zare, *ACS Cent. Sci.*, 2017, **3**, 1337–1344.
16 F. A. Faber, L. Hutchison, B. Huang, J. Gilmer, S. S. Schoenholz, G. E. Dahl, O. Vinyals, S. Kearnes, P. F. Riley and O. A. von Lilienfeld, *J. Chem. Theory Comput.*, 2017, **13**, 5255–5264.
17 J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals and G. E. Dahl, 2017, arXiv preprint arXiv:1704.01212.
18 K. Schütt, P.-J. Kindermans, H. E. S. Felix, S. Chmiela, A. Tkatchenko and K.-R. Müller, *Advances in Neural Information Processing Systems*, 2017, pp. 991–1001.
19 K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller and A. Tkatchenko, *Nat. Commun.*, 2017, **8**, 13890.
20 J. S. Smith, O. Isayev and A. E. Roitberg, *Chem. Sci.*, 2017, **8**, 3192–3203.
21 E. N. Feinberg, D. Sur, B. E. Husic, D. Mai, Y. Li, J. Yang, B. Ramsundar and V. S. Pande, 2018, arXiv preprint arXiv:1803.04465.
22 Z. Liu, M. Su, L. Han, J. Liu, Q. Yang, Y. Li and R. Wang, *Acc. Chem. Res.*, 2017, **50**, 302–309.
23 J. Hachmann, R. Olivares-Amaya, S. Atahan-Evrenk, C. Amador-Bedolla, R. S. Sánchez-Carrera, A. Gold-Parker, L. Vogt, A. M. Brockway and A. Aspuru-Guzik, *J. Phys. Chem. Lett.*, 2011, **2**, 2241–2251.
24 M. M. Mysinger, M. Carchia, J. J. Irwin and B. K. Shoichet, *J. Med. Chem.*, 2012, **55**, 6582–6594.
25 Y. Gal, Uncertainty in Deep Learning, PhD thesis, University of Cambridge, 2016.
26 R. McAllister, Y. Gal, A. Kendall, M. van der Wilk, A. Shah, R. Cipolla and A. V. Weller, Concrete Problems for Autonomous Vehicle Safety, *Advantages of Bayesian Deep Learning, Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence AI and autonomy track*, 2017, pp. 4745–4753.
27 E. Begoli, T. Bhattacharya and D. Kusnezov, *Nat. Mach. Intell*, 2019, **1**, 20.
28 A. Kendall and Y. Gal, *Advances in neural information processing systems*, 2017, pp. 5574–5584.
29 Y. Gal, R. Islam and Z. Ghahramani, *Proceedings of the 34th International Conference on Machine Learning*, 2017, vol. 70, pp. 1183–1192.
30 K. P. Murphy, *Machine Learning: A Probabilistic Perspective, Adaptive Computation and Machine Learning series*, 2018.
31 D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik and R. P. Adams, *Advances in neural information processing systems*, 2015, pp. 2224–2232.
32 T. N. Kipf and M. Welling, 2016, arXiv preprint arXiv:1609.02907.
33 S. Ryu, J. Lim and W. Y. Kim, 2018, arXiv preprint arXiv:1805.10988.
34 Y. Kwon, J.-H. Won, B. J. Kim and M. C. Paik, *international conference on medical imaging with deep learning*, 2018.
35 A. Gelman, H. S. Stern, J. B. Carlin, D. B. Dunson, A. Vehtari and D. B. Rubin, *Bayesian data analysis*, Chapman and Hall/CRC, 2013.
36 C. Blundell, J. Cornebise, K. Kavukcuoglu and D. Wierstra, 2015, arXiv preprint arXiv:1505.05424.
37 A. Graves, *Advances in neural information processing systems*, 2011, pp. 2348–2356.
38 C. Louizos and M. Welling, 2017, arXiv preprint arXiv:1703.01961.
39 Y. Gal and Z. Ghahramani, *international conference on machine learning*, 2016, pp. 1050–1059.
40 N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, *J. mach. learn. res.*, 2014, **15**, 1929–1958.
41 V. Kuleshov, N. Fenner and S. Ermon, 2018, arXiv preprint arXiv:1807.00263.
42 Y. Gal and L. Smith, 2018, arXiv preprint arXiv:1806.00667.
43 A. Der Kiureghian and O. Ditlevsen, *Struct. Saf.*, 2009, **31**, 105–112.
44 Y. Gal, J. Hron and A. Kendall, *Advances in Neural Information Processing Systems*, 2017, pp. 3581–3590.
45 D. P. Kingma and J. Ba, 2014, arXiv preprint arXiv:1412.6980.
46 A. Gaulton, A. Hersey, M. Nowotka, A. P. Bento, J. Chambers, D. Mendez, P. Mutowo, F. Atkinson, L. J. Bellis, E. Cibrián-Uhalte, *et al.*, *Nucleic Acids Res.*, 2016, **45**, D945–D954.
47 G. Landrum, *RDKit: Open-source cheminformatics*, 2006.
48 M. C. Scharber, D. Mühlbacher, M. Koppe, P. Denk, C. Waldauf, A. J. Heeger and C. J. Brabec, *Adv. Mater.*, 2006, **18**, 789–794.
49 S. A. Lopez, E. O. Pyzer-Knapp, G. N. Simm, T. Lutzow, K. Li, L. R. Seress, J. Hachmann and A. Aspuru-Guzik, *Sci. Data*, 2016, **3**, 160086.
50 Y. Zhao and D. G. Truhlar, *Theor. Chem. Acc.*, 2008, **120**, 215–241.
51 E. O. Pyzer-Knapp, G. N. Simm and A. A. Guzik, *Mater. Horiz.*, 2016, **3**, 226–233.
52 D. R. Jones, M. Schonlau and W. J. Welch, *J. Glob. Optim.*, 1998, **13**, 455–492.
53 R.-R. Griffiths and J. M. Hernández-Lobato, 2017, arXiv preprint arXiv:1709.05501.
54 F. HaÌĹse, L. M. Roch, C. Kreisbeck and A. Aspuru-Guzik, *ACS Cent. Sci.*, 2018, **4**, 1134–1145.
55 J. M. Hernández-Lobato, J. Requeima, E. O. Pyzer-Knapp and A. Aspuru-Guzik, *Proceedings of the 34th International Conference on Machine Learning*, 2017, vol. 70, pp. 1470–1479.
56 D. A. Cohn, Z. Ghahramani and M. I. Jordan, *J. Artif. Intell. Res.*, 1996, **4**, 129–145.

This journal is © The Royal Society of Chemistry 2019