Deep Learning Model for Predicting Solvation Free Energies in Generic Organic Solvents

Hyuntae Lim* and YounJoon Jung*

Department of Chemistry, Seoul National University, Seoul 08826, Korea

E-mail: ht0620@snu.ac.kr; yjjung@snu.ac.kr

Abstract

Prediction of aqueous solubilities or hydration free energies is an extensively studied area in machine learning applications on chemistry since water is the sole solvent in the living system. However, for non-aqueous solutions, few machine learning studies have been undertaken so far despite the fact that the solvation mechanism plays an important role in various chemical reactions. Here, we introduce a novel, machine-learning based quantitative structure-property prediction method which predicts solvation free energies for various organic solute and solvent systems. A novelty of our method involves two separate solvent and solute encoder networks that can quantify structural features of given compounds via word embedding and recurrent layers, with the attention mechanism which extracts important substructures from outputs of recurrent neural networks. As a result, the predictor network calculates solvation free energy of a given mixture using features from encoders. With results obtained from extensive calculations on 2495 solute-solvent mixtures, we demonstrate that our methodology outperforms both *ab initio* and MD solvation model in terms of estimation error for solvation energy.

1 Introduction

The most common strategies to predict biological or physicochemical properties of chemical compounds are *ab initio* quantum mechanical approaches[1–5] like Hartree-Fock (HF) or density functional theory (DFT), and molecular dynamics (MD) simulation method based on classical Newtonian and statistical mechanics.[6–9] These methods with precisely defined theoretical backgrounds have been successfully used in calculating various features of chemical compounds. However, such methods have limitations in computational resources and time costs since they require an enormous amount of numerical calculations. As an alternative, recent successes in machine learning (ML) technique and its implementation in to cheminformatics are promoting broad applications of ML for chemical studies. Quantitative structure-activity relationship (QSAR) or quantitative structure-property relationship (QSPR) analysis is one of such techniques to predict various properties of a given compound from its empirical or structural features.[10,11] The underlying architecture of QSAR/QSPR consists of two elementary mathematical functions.[11] One is the *encoding function*, which encodes the chemical structure of the given compound into a *molecular descriptor*. The *mapping function*, the other, predicts the target property (or activity) that we intend to know using the encoded descriptor.

There have been various molecular descriptors proposed to represent structural features of compounds efficiently. For example, we can feature a given molecule with simple enumerations of empirical properties like molecular weights, rotatable bonds, the number of hydrogen bond donors and acceptors, or some pre-experimented or pre-calculated properties.[12] On the other hands, molecular fingerprints, which is another option, are commonly used in cheminformatics to estimate chemical 'difference' between more than two compounds.[13] They usually have a fixed size of binary sequence and are easily obtainable from SMILESs with pre-defined criteria. Graph representations of molecules based on graph theory are another major encoding methods in QSAR/QSPR which have receive a great attention in recent days.[14,15] They have exhibited their outstanding prediction performances in diverse

chemical or biophysical properties.[16]

The mapping function extracts properties which we want to know from encoded molecular features of the given compound via classification or regression method. We can use any suitable machine learning methods in mapping functions[11,16] such as random forest(RF), support vector machine (SVM), neural network(NN), and so on. Among these diverse technical options, NN seems to be the method which shows the most rapid advances in recent years,[12,17{21] on the strength of the theoretical advances[22] and evolution of computational power. Many studies have already been performed to show that various chemical or biophysical properties of compounds are obtainable from the QSAR/QSPR combined with machine learning technique.[12,16{21,23,24]

Solvation is one of the most fundamental processes occurring in physical chemistry, and many theoretical and computational studies have been executed to calculate solubilities or solvation free energies using a variety of methodologies.[25,26] For example, we can roughly guess solubilities using solvation parameters, but solvation parameters only provide relative order, not the quantitative value.[27] The general solubility equation (GSE) enables us to calculate solubilities from some empirical parameters, but it only provides solubilities for aqueous solutions.[28] *Ab initio*[1{5] or MD simulations[6{9] provide us with more concrete, accurate results and more in-depth knowledge about solvation mechanism, but they have practical limitations due to high usage of computational resources as mentioned before.

Recent studies demonstrated that QSPR with ML successfully predicts aqueous solubilities or hydration free energies of diverse solutes.[12,16,17,21,29,30] They also proved that ML guarantees faster calculations than computer simulations and more precise estimations than GSE estimation; a decent number of models showed accuracies comparable to *ab initio* solvation models.[16] However, the majority of QSPR prediction for solubilities have been limited to aqueous solutions cases. For non-aqueous solutions, few studies have been undertaken to predict the solubility despite the fact that predicting solubilities play an important role in the development of varied fields of chemistry, e.g., organic synthesis,[31] electrochemical

reactions in batteries,[32] and so on.

In the present work, we introduce a QSPR combined with recurrent neural network (RNN) model which is specialized in predicting solvation free energies of organic compounds in various solvents. The model has three primary sub-neural networks: the solvent and solute encoder networks and the predictor network. For basic featurization of a given molecule, we use the word embedding technique.[30,33] We calculate solvation energies of 2,495 mixtures for 418 solutes and 91 solvents,[34] demonstrate that our model performs more precise predictions than both classical[7] and quantum mechanical[2,5] solvation models.

The rest of the present paper is outlined as follows: Section 2 describes the embedding method for molecular structure and overall architecture of the neural network. In Section 3, we mainly compare the performance of our methodology with both MD and *ab initio* simulation strategies, and visualize important substructures via attention mechanism. We also discuss about database sensitivity using cluster cross-validation method. In the last section, we conclude our work.

2 Methods

2.1 Word Embedding

Natural language processing (NLP) is one of most cutting-edge technologies in varied applications of machine learning and neural networks.[33,35–38] To process human languages using computers, we need to encode words and sentences and extract their linguistic properties. The process is commonly implemented via *word embedding* method.[33,35] To perform the task, unsupervised learning schemes such as skip-gram and continuous bag of words (CBOW) algorithms generate a vector representation of the given word in an arbitrary vector space.[33,35] If the necessary vector space is well-defined, one can conjecture the semantic or syntactic features of the given word from the position of the embedded vector, and the inner product of two vectors corresponding to two different words provides information about their semantic

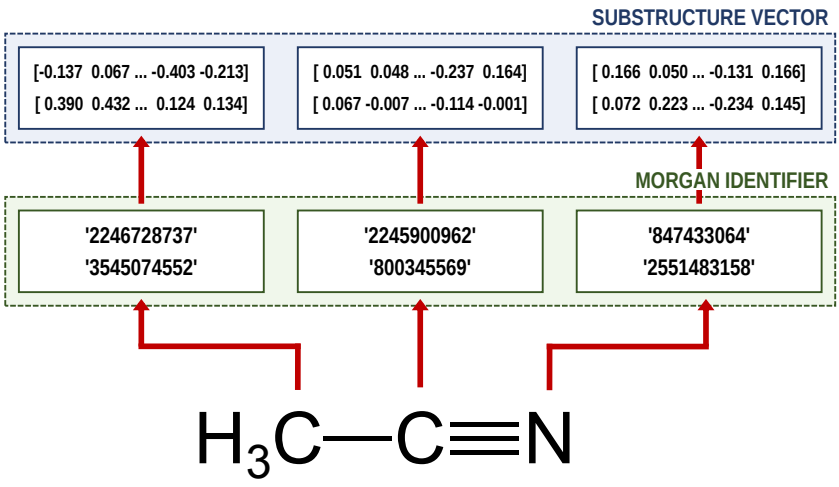

Figure 1: Schematic illustration of the molecular embedding process for acetonitrile (SMILES: CC#N) and $r_{\text{max}} = 1$. The Morgan algorithm discriminates identifiers between two substructures: one is for itself ($r = 0$) and the other considers its nearest neighbor atoms ($r = 1$). Then the embedding layer calculates the vector representation from the given identifier.

similarity.

It is worthwhile to note that we can employ the embedding technique for chemical or biophysical processes if we consider an atom or a substructure as a word and a compound as a sentence.[29,30,39] In that case, positions of molecular substructures in the embedded vector space represent their chemical and physical properties, instead of linguistic information. There are already bio-vector models[39] that have been developed to which encode sequences of proteins or DNAs, and atomic-vector embedding models have been introduced recently to encode structural features of chemical compounds.[29,30] Mol2Vec is one of such embedding techniques, and it generates vector representations of a given molecule from the *molecular sentence*.[30] To make molecular sentences, Mol2Vec uses the Morgan algorithm[40] that assorts identical atoms in the molecule. The algorithm is commonly used to generate ECFP fingerprints,[41] which are the *de facto* standard in cheminformatics,[13] and it makes identifiers of the given atom from the chemical environment where the atom is positioned. An atom may have multiple identifiers depending on the pre-set maximum value of *radius* r_{max}, which denotes the maximum topological distance between the given atom and its neighbor-

ing atoms. The atom itself is identified by $r = 0$, and additional substructure identifiers for adjacent atoms are denoted by $r = 1$ (nearest neighbor), $r = 2$ (next nearest neighbor), and so on. Since Mol2Vec has demonstrated promising performances in several applications of QSAR/QSPR,[30] we use Mol2Vec as the primary encoding means in the present study. We schematically illustrated embedding procedure for acetonitrile in Fig. 1.

2.2 Encoder-Predictor Network

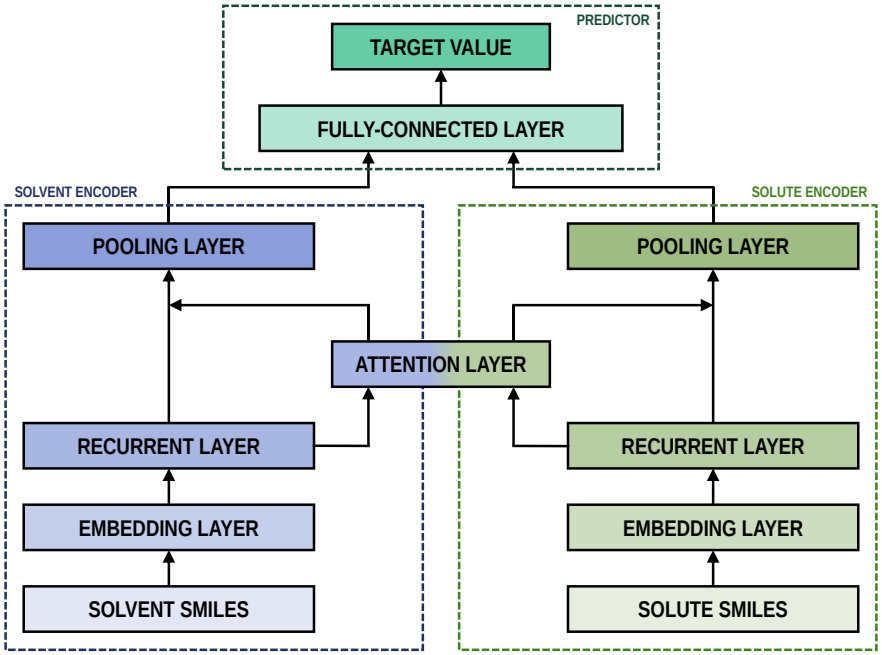

Figure 2: The fundamental architecture of our prediction model. Each encoder network has one embedding and one recurrent layer, while the predictor has a fully-connected MLP layer. Two encoders share an attention layer, which weights outputs from recurrent layers. Black arrows indicate flow of input data.

As shown in Fig. 2, the proposed machine learning model has three sub-networks: the solvent and the solute encoders extract dominant structural features of the given compound from SMILES strings, while the predictor calculates the solvation energy of the given solvent/solute mixture from their encoded features.

The primary architecture of the encoder is based on two bidirectional recurrent neural networks (BiRNNs).[42] The network is designed for handling sequential data and we consider

the molecular sentence $[\mathbf{x}_1; \cdots ; \mathbf{x}_N]$ as a sequence of embedded substructures, $\mathbf{x}_i$. RNNs may have a failure when input sequences are lengthy; gradients of the loss function can be diluted or amplified because of accumulated precision error from the backpropagation process.[43] The excessive or restrained gradient may cause a decline in learning performance, and we call these two problems as vanishing or exploding gradient. To overcome these limits which stem from lengthy input sequences, one may consider using both forward-directional RNN ($\overrightarrow{\text{RNN}}$) and backward-directional RNN ($\overleftarrow{\text{RNN}}$) within a single layer:

$$\begin{aligned}
\overrightarrow{\text{RNN}}([\mathbf{x}_1; \cdots ; \mathbf{x}_N]) &= [\overrightarrow{\mathbf{h}_1}; \cdots ; \overrightarrow{\mathbf{h}_N}] \\
\overleftarrow{\text{RNN}}([\mathbf{x}_1; \cdots ; \mathbf{x}_N]) &= [\overleftarrow{\mathbf{h}_1}; \cdots ; \overleftarrow{\mathbf{h}_N}] \\
\overleftrightarrow{\text{RNN}}([\mathbf{x}_1; \cdots ; \mathbf{x}_N]) &= [\mathbf{h}_1; \cdots ; \mathbf{x}_N]
\end{aligned} \tag{1}$$

In Eqn. 1, $\mathbf{x}_i$ is the embedded atomic vector of a given molecule, $\overrightarrow{\mathbf{h}_i}$ and $\overleftarrow{\mathbf{h}_i}$ are hidden state outputs of each recurrent unit, and $\mathbf{h}_i = \overrightarrow{\mathbf{h}_i}; \overleftarrow{\mathbf{h}_i}$ means concatenation of two hidden states, respectively. The long-short term memory[44] (LSTM) and gated recurrent unit[45] (GRU) networks, which are modifications of RNN, are invented to handle lengthy input sequences. They introduce *gates* in each RNN cell state to memorize important information of the previous cell state and minimize vanishing and exploding gradient problem.

After RNN layers, the molecular sentences of both the solvent $\mathbf{X} = [\mathbf{x}_1; \cdots ; \mathbf{x}_N]$ and the solute $\mathbf{Y} = [\mathbf{y}_1; \cdots ; \mathbf{y}_M]$ are converted to hidden states, $\mathbf{H} = [\mathbf{h}_1; \cdots \mathbf{h}_N]$ and $\mathbf{G} = [\mathbf{g}_1; \cdots ; \mathbf{g}_M]$, respectively. Each hidden state is then inputted to the shared *attention* layer and weighted. The attention mechanism, which was originally proposed to enhance performances of machine translator,[36] is an essential technique in diverse NLP applications nowadays.[37,38] Principles of the attention start from the definition of the score function of

hidden states and its normalization with the softmax function:

$$
\begin{aligned}
{}_{ij} &= \frac{\exp(\mathrm{score}(\mathbf{h}_i, \mathbf{g}_j))}{\sum_k \exp(\mathrm{score}(\mathbf{h}_i, \mathbf{g}_k))} \\
\mathbf{p}_i &= \sum_j^M {}_{ij}\mathbf{g}_j
\end{aligned}
\tag{2}
$$

There are various score functions that have been introduced to achieve efficient predictions,[36–38] and among them we use the dot product of two hidden states, $\mathbf{h}_i \cdot \mathbf{g}_j$ as a score function. The solvent context, $\mathbf{P} =$ $\mathbf{G}$ denotes an *emphasized* hidden state $\mathbf{H}$ with the attention alignment, . We also get the solute context $\mathbf{Q}$ using the same procedure. The context weighted from the attention layer is an $L \times 2D$ matrix, where L is the sequence length and D is the dimension of two RNN hidden layers since we use bidirectional RNN (BiRNN). Two max-pooling layers, which is the last part of each encoder reduces contexts $\mathbf{H}$, $\mathbf{G}$, $\mathbf{P}$, and $\mathbf{Q}$ to $2D$-dimensional feature vectors $\mathbf{u}$ and $\mathbf{v}$:[38]

$$
\begin{aligned}
\mathbf{u} &= \mathrm{MaxPooling}([\mathbf{h}_1; \mathbf{p}_1; \cdots; \mathbf{h}_N; \mathbf{p}_N]) \\
\mathbf{v} &= \mathrm{MaxPooling}([\mathbf{g}_1; \mathbf{q}_1; \cdots; \mathbf{g}_M; \mathbf{q}_M])
\end{aligned}
\tag{3}
$$

The predictor has a single fully-connected perceptron layer with rectifier unit (ReLU) and an output layer. It uses the concatenated feature of the solvent and solute $[\mathbf{u}; \mathbf{v}]$ as an input. The overall architecture of our model is shown in Figure 2. We also consider encoders without RNN and attention layers in order to quantify the impact of these layers on prediction performances of the network; each encoding network contains only the embedding layer and directly connected to the MLP layer. The solvent and solute features are simple summations of atomic vectors, $\mathbf{u} = \sum_i^N \mathbf{x}_i$ and $\mathbf{v} = \sum_i^M \mathbf{y}_i$, respectively. This model was initially used for gradient boosting (GBM) regression analysis for aqueous solubilities and toxicities.[30]

3 Results and Discussions

3.1 Computational Setup and Results

We use the Minnesota solvation database[34] (MNSOL) as the dataset over which we train and test, and it provides 3,037 experimental measures of free energies of solvation and transfer energies for 790 unique solutes in 92 solvents. Because the MNSOL only contains common names of compounds, we perform an automated searching process using PubChemPy script and receive SMILES strings of compounds from PubChem database. There are 363 results for charged solutes and 144 results for transfer free energies in the MNSOL which are excluded from machine learning dataset, and 35 results of solvent-solute combinations are not valid in PubChem. We finally prepare SMILES specifications of 2,495 solutions for 418 solutes and 91 solvents for the machine learning input.

For an implementation of the neural networks, we use Keras 2.2.4 framework with TensorFlow 1.12 backend. At the very first of stage, Morgan algorithm for $r = 0$ and $r = 1$ generates molecular sentences of the solvent and solute from their SMILES strings. Then the given molecular sentence is embedded to a sequence of 300-dimensional substructure vectors by pre-trained Word2Vec model available at https://github.com/samoturk/mol2vec, which contains information of $\sim 20;000;000$ compounds and $\sim 20;000$ substructures from ZINC and ChEMBL databases.[30] We consider BiLSTM and BiGRU layers in both solvent and solute encoders to compare their performances. Since our model is a regression problem, we use mean squared error (MSE) as the loss function.

We employ 10-fold cross-validation (CV) for secure representativeness of the test data because the dataset we use has a limited number of experimental measures; the total dataset is uniformly and randomly split into 10 subsets, and we iteratively choose one of the subsets as a test set and the training run uses the remainder 9 subsets. Consequentially, a 10-fold CV task performs 10 independent training and test runs, and relative sizes of the training and test sets are 9 to 1. We use Scikit-Learn library to implement the CV task and perform

an extensive grid search for tuning hyperparameters: learning algorithms, learning rates, and dimensions of hidden layers. We select the stochastic gradient descent (SGD) algorithm with Nesterov momentum, whose learning rate is 0.0002 and momentum is 0.9. Optimized hidden dimensions are 150 for recurrent layers and 2000 for the fully connected layer. To minimize the variance of the test run, we take averages for all results over 9 independent random CV, split from different random states.

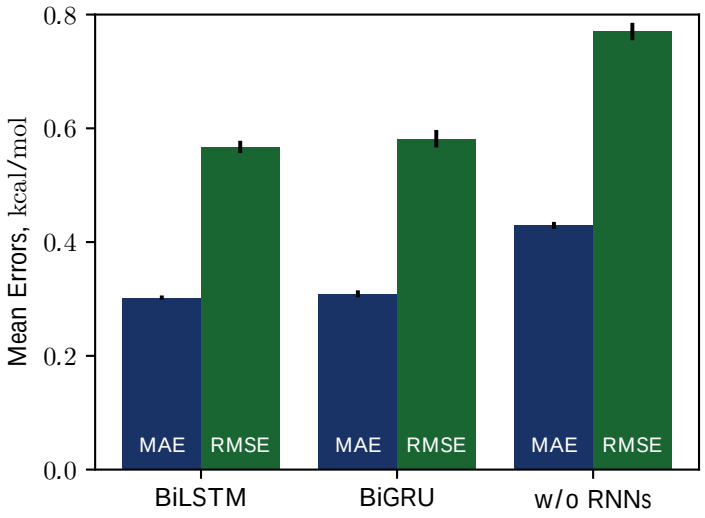

Figure 3: Benchmark chart for three kinds of encoder networks, for two metrics (MAE and MSE). The BiLSTM and the BiGRU models show no significant differences, while it makes relatively inaccurate predictions without recurrent networks. All results are averaged over 9 independent test runs and black lines on tops of boxes denote variances.

Solvation free energies that we calculated from the MNSOL using attentive BiRNN encoders are exhibited in Fig. 3 and 4. Prediction errors for the BiLSTM model are ± 0.57 kcal=mol in RMSE, ± 0.30 kcal=mol in MAE, and the Pearson correlation coefficient is $R^2 = 0.96$ while results from the BiGRU model indicate there is no meaningful difference between the two recurrent models. The encoder without BiRNN and attention layers produces much more inaccurate results, whose error metrics are ± 0.77 kcal=mol in RMSE, ± 0.43 kcal=mol in MAE, and 0.92 in R^2 value, respectively. We cannot directly compare our results with other ML models because this is the first ML-based study using the MNSOL database. Nonetheless, several studies on aqueous system have previously calculated

solubilities or hydration free energies using various ML techniques and molecular descriptors.[12,16,17,21,29,30] Those results showed correlation coefficients for hydration energies were below $R^2 = 0.92$ while our predictions from the BiLSTM encoder are $R^2 = 0.94$ for 374 aqueous solutions, which implies our neural network model guarantees considerably precise predictions.

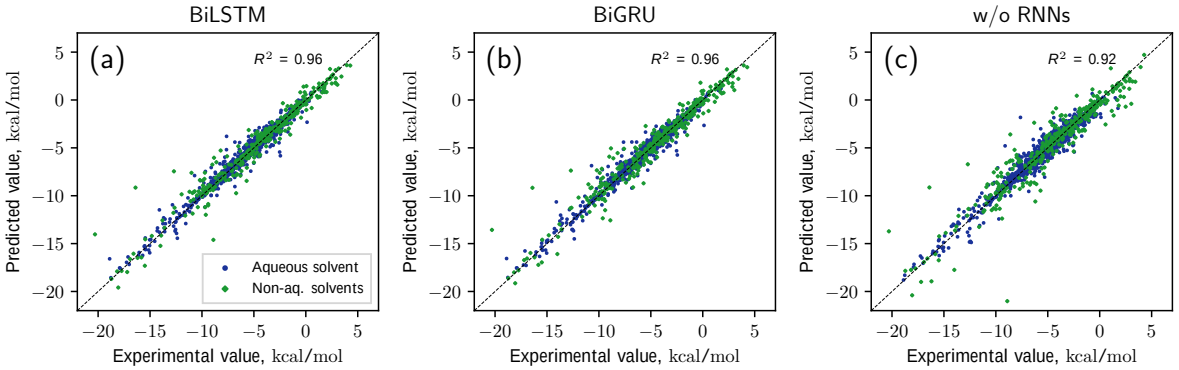

Figure 4: Scatter plot for true (x-axis) and ML predicted (y-axis) values of solvation energies in three different models: (a) BiLSTM, (b) BiGRU, and (c) without recurrent layers. All results are averaged over 9 independent 10-fold CV runs.

Meanwhile, for studies which are not ML-based, there are several results from both classical and quantum-mechanical simulation studies that use the MNSOL as the reference data.[2{5,7,9] A DFT study which introduced SM12 implicit solvation model calculated most of the solvation energies in the MNSOL.[2] The researchers used several hybrid functionals and basis sets, calculated free energies of solvation of uncharged solutes for 374 aqueous solutions and 2129 non-aqueous solutions. They obtained the best result at SM12CM5 solvation model, B3LYP hybrid functional, and MG3S basis set. Computational errors for those conditions were ±0.77 kcal=mol for aqueous solvents while non-aqueous solvents indicated ±0.54 kcal=mol in MAE. The BiLSTM model we built shows slightly more precise predictions in water solvents, which MAE is ±0.64 kcal=mol. Our model makes much better predictions in organic solvents; machine learning for 2121 non-aqueous solutions result in ±0.24 kcal=mol, which is lesser than half of SM12CM5 solvation model.

Table 1: Comparisons between encoder-predictor networks and various quantum-mechanical solvation models for aqueous and non-aqueous solutions. The error metric is MAE and kcal=mol. Data in bold texts are our results, while all QM results are taken from the work of Marenich et al.[2].

Solvent	Method	N_{data}	MAE	Ref
Aqueous	SM12CM5/B3LYP/MG3S	374	0.77	Marenich et al.[2]
	SM12ESP/ChElPG/M06-2X	374	0.84	Marenich et al.[2]
	SM8/M06-2X/6-31G(d)	366	0.89	Marenich et al.[2]
	SMD/M05-2X/6-31G(d)	366	0.88	Marenich et al.[2]
	BiLSTM	**374**	**0.64**	
	BiGRU	**374**	**0.68**	
	w/o RNNs	**374**	**0.90**	
Non-aqueous	SM12CM5/B3LYP/MG3S	2129	0.54	Marenich et al.[2]
	SM12ESP/ChElPG/M06-2X	2129	0.56	Marenich et al.[2]
	SM8/M06-2X/6-31G(d)	2129	0.61	Marenich et al.[2]
	SMD/M05-2X/6-31G(d)	2129	0.67	Marenich et al.[2]
	BiLSTM	**2121**	**0.24**	
	BiGRU	**2121**	**0.24**	
	w/o RNNs	**2121**	**0.36**	

3.2 Visualization of Attention Mechanism

A useful aspect of attention mechanism is that the model provides not only the prediction value of solvation energy of a given mixture but also a clue to why the neural network makes such a prediction based on the correlations between recurrent hidden states.[21,29,37] In this section, we visualize how the attention layer operates, and verify how such correlations correspond well to chemical intuitions for inter-molecular interactions. The matrix of attention alignments, from Eqn. 2 indicates which substructures in the given solvent and solute mixture strongly correlated with each other so they play dominant roles in determining their solvation energy. In Figure 5, we demonstrate attention alignments of nitromethane (CH_3NO_2) solute in four different solvents: 1-octanol ($C_8H_{17}OH$, 3:51 kcal=mol), 1-butanol (C_4H_9OH, 3:93 kcal=mol), ethanol (C_2H_5OH, 4:34 kcal=mol), and acetonitrile (CH_3CN, 5:62 kcal=mol). The scheme for visualizing attention alignments is as follows: (i) first, we calculate the average alignment $\langle \ \rangle_j$ of each substructure j of the solute over the entire solvent structure $\{i\}$, $\langle \ \rangle_j = \sum_i^N \ _{ij}=N$. (ii) Then, we get relative amounts of averaged alignments $[\ \tilde{}_1; \cdots ; \ \tilde{}_M]$

from dividing by the maximum value, $\tilde{}_j = \langle\ \rangle_j / \max(\langle\ \rangle_1; \cdots; \langle\ \rangle_M)$. (iii) Also, since the embedding algorithm we use generates two substructure vectors per an atom, we individually visualize two alignments maps, $[\tilde{}_1; \tilde{}_3; \cdots; \tilde{}_{M-1}]$ (for $r = 0$) and $[\tilde{}_2; \tilde{}_4; \cdots; \tilde{}_M]$ (for $r = 1$) for more simple and intuitive illustration. (iv) Finally, the color representation of each atom in Fig. 5 denotes the amount of $\tilde{}_j$; the neural network judges that red-colored substructures (higher $\tilde{}_j$) in the solute are more "similar" to the solvent and the model puts more weights on them during the prediction task. In contrast, green-colored substructures have lower $\tilde{}_j$, which means they do not have similarity with the solvent molecule so much as red-colored one.

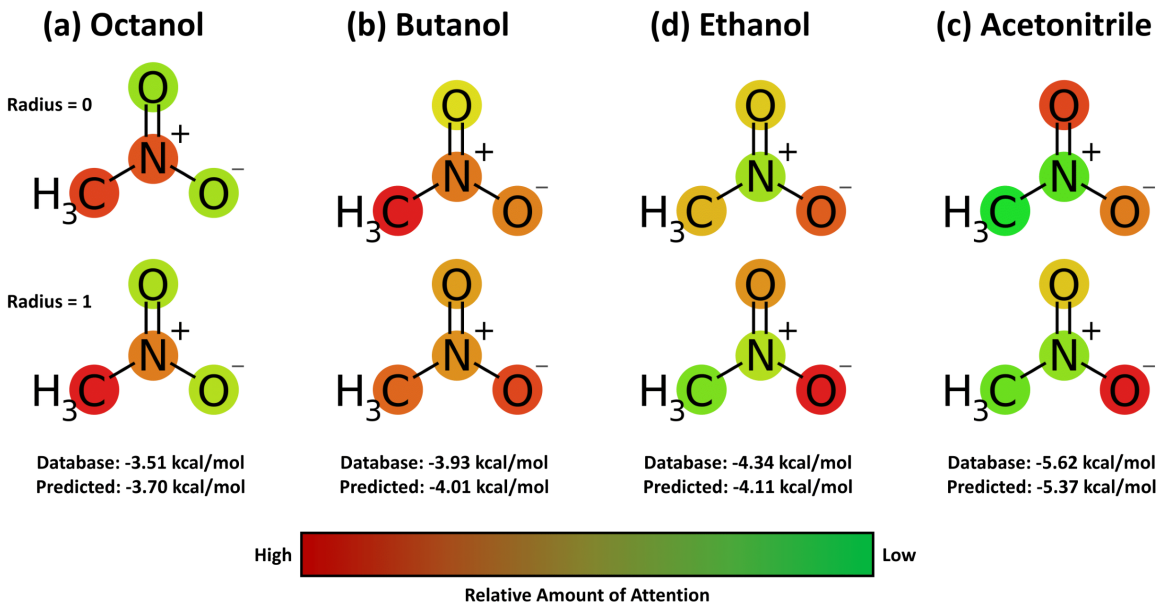

Figure 5: Relative and mean attention alignments map for nitromethane in four different solvents: (a) octanol, (b) butanol, (c) ethanol, (d) and acetonitrile, respectively. Color representations denote that the neural network invests more weights on red, while green substructures have relatively low contributions for the solvation energy.

Overall results in Fig. 5 imply that the *chemical similarity* taken from the attention layer has a significant connection to fundamental knowledge of chemistry like polarity or hydrophilicity. Each alcoholic solvent has one hydrophilic $-$OH group, and it results in increasing contributions of the nitro group in the solute as hydrocarbon chains of alcohols

shorten. For the acetonitrile-nitromethane solution, the attention mechanism reflects the highest contributions of $-NO_2$ groups due to strong polarity and aprotic nature of the solvent. Although the attention mechanism seems to reproduce molecular interactions in a faithful way however, we find there is a defective prediction which does not agree with chemical knowledge. Two oxygen atoms =O and $-O^{\text{ɬ}}$ in the nitro group are indistinguishable due to the resonance structure, thus they must have equivalent contributions in any solvents, but we find they show different attention scores in our model. We believe those problems happen because the SMILES string of nitromethane (C[N+](=O)[O-]) does not encode the resonance effect in the nitro group. Indeed, the Morgan algorithm generates different identifiers for two oxygen atoms in the nitro group, [864942730, 2378779377] for =O and [864942795, 2378775366] for $-O^{\text{ɬ}}$. The absence of resonance might be a problem worthwhile considering when one intends to use word embedding models with SMILES strings,[29,30,46] although estimated solvation energies for nitromethane mixtures from the BiLSTM model are within a moderate error range as shown in Fig. 5.

3.3 Transferability of the Model for New Compounds

Since our study uses techniques of machine learning with empirical data from experimental measures, there is a likelihood that the model would not guarantee prediction accuracy for "entirely new" solvents or solutes which are not present in the dataset, although the MNSOL contains a considerable number commonly-used solvents and solutes.[34] In order to investigate this potential issue, we perform another train and test runs with the cluster cross-validation,[46,47] instead of using the random-split CV. As a start, we individually obtain 10 clusters for solvents and solutes using the K-mean clustering algorithm and the molecular vector. The molecular vector is a simple summation of substructure vectors as we used for the simple MLP model without RNN encoders:[30] $\mathbf{u} = \sum_i^N \mathbf{x}_i$ for solvents and $\mathbf{v} = \sum_i^M \mathbf{y}_i$ for solutes, respectively. Then, we iteratively perform cross-validation process over each cluster. The size of each cluster is [422, 482, 186, 231, 443, 243, 143, 251, 15, 79] for solvents and

[401, 672, 514, 75, 64, 6, 512, 54, 42, 155] for solutes, respectively.

Table 2: Prediction accuracy of the random-split CV, the solvent and solute cluster CVs using K-mean algorithm, and several MD/*ab initio* solvation models for four different organic solvents: toluene ($C_6H_5CH_3$), chloroform ($CHCl_3$), acetonitrile (CH_3CN), and dimethyl sulfoxide ($(CH_3)_2SO$), respectively. Units of MAE and RMSE are kcal=mol.

Solvent	Method	N_{data}	MAE	RMSE	Ref
All	**Random CV**	**2495**	**0.30**	**0.57**	
	Solvent Clustering	**2495**	**0.82**	**1.45**	
	Solute Clustering	**2495**	**0.99**	**1.61**	
Toluene	MD/GAFF	21	0.48	0.63	Mohamed et al.[7]
	MD/AMOEBA	21	0.92	1.18	Mohamed et al.[7]
	Random CV	**21**	**0.16**	**0.37**	
	Solvent Clustering	**21**	**0.66**	**1.10**	
	Solute Clustering	**21**	**0.93**	**1.46**	
Chloroform	MD/GAFF	21	0.92	1.11	Mohamed et al.[7]
	MD/AMOEBA	21	1.68	1.97	Mohamed et al.[7]
	Random CV	**21**	**0.35**	**0.56**	
	Solvent Clustering	**21**	**0.78**	**0.87**	
	Solute Clustering	**21**	**1.14**	**1.62**	
Acetonitrile	MD/GAFF	6	0.43	0.52	Mohamed et al.[7]
	MD/AMOEBA	6	0.73	0.77	Mohamed et al.[7]
	PM6/SMD	6	-	1.2	Kromann et al.[5]
	PM6/COSMO	6	-	2.3	Kromann et al.[5]
	Random CV	**6**	**0.29**	**0.39**	
	Solvent Clustering	**6**	**0.74**	**0.82**	
	Solute Clustering	**6**	**0.80**	**0.94**	
DMSO	MD/GAFF	6	0.61	0.75	Mohamed et al.[7]
	MD/AMOEBA	6	1.12	1.21	Mohamed et al.[7]
	PM6/SMD	6	-	1.7	Kromann et al.[5]
	PM6/COSMO	6	-	3.1	Kromann et al.[5]
	Random CV	**6**	**0.41**	**0.44**	
	Solvent Clustering	**6**	**0.93**	**1.19**	
	Solute Clustering	**6**	**0.91**	**1.11**	

Results from the solvent and the solute cluster CV tasks in Table 2 indicate generalized expectation error ranges for new solvents or solutes which are not in the dataset. As shown in the table, we find that the split method based on the clustering brings an apparent decline in prediction performance. For the BiLSTM encoder model, increments of MAE are 0.52 kcal=mol for the solvent clustering and 0.69 kcal=mol for the solute clustering. The

reason why the random K-fold CV exhibits superior performances is obvious; if we have a mixture ($\mathcal{A}$, $\mathcal{B}$) of solvent $\mathcal{A}$ and solute $\mathcal{B}$ in the test set and the training set have ($\mathcal{A};\mathcal{C}$) and ($\mathcal{D};\mathcal{B}$) mixtures, then both ($\mathcal{A};\mathcal{C}$) and ($\mathcal{D};\mathcal{B}$) could enhance prediction accuracy of ($\mathcal{A}$, $\mathcal{B}$). However, the clustering limits the location of a specific compound, and mixtures of specific solvent or solute should be either in the test set or the train set.

For comparison, Table 2 also contains calculation errors of both classical[7] and semi-empirical DFT[5] studies for four organic solvents: toluene ($C_6H_5CH_3$), chloroform ($CHCl_3$), acetonitrile (CH_3CN), and dimethyl sulfoxide ($(CH_3)_2SO$), respectively. Although the MD study performed classical dynamics simulations, both GAFF and AMOEBA polarizable force field exhibited more precise predictions than SMD or COSMO-RS solvation models in acetonitrile and DMSO since they consider explicitly implemented solvent. The random CV makes even more accurate predictions than MD simulations for all four solvents, which is consistent with comparisons in the previous section. The bottom line of cluster CV is if the dataset for train contains at least one side of the solvent-solvent mixture we want to estimate its solvation free energy, the expectation error of our model is within chemical accuracy 1.0 kcal/mol, which is the general error of computer simulation scheme. Also, results for four organic solvents demonstrates that predictions from the cluster CV have the accuracy that is comparable with MD simulations using AMOEBA force field.[7]

Results from the cluster CV highlight the necessity for discussion on the importance of database preparation. As described earlier, the cluster CV causes a considerable increase in prediction error, and we suspect that those degradations mainly come from the decline in the diversity of the training set. Namely, the number of substructures that the neural network learns in training process is not as many as the random CV if we use the cluster CV. To prove this speculation, we define *unique* substructures, which are substructures only exists in the test cluster. As shown in Figure 6, in the solute cluster CV, MAE for 1,226 mixtures which don't have any unique substructures in solutes is ± 0.54 kcal/mol, while the prediction error for the rest 1,269 solutions is ± 1.64 kcal/mol. The solvent cluster CV shows more

extreme results: the MAE for 374 aqueous mixtures is ±2:48 kcal=mol, while non-aqueous solutions exhibit ±0:52 kcal=mol in contrast. We believe that the outlying behavior of water is due to its distinctive nature. Water has only one, unique substructure since the oxygen atom does not have any neighbors. So the solvent clustering makes the network unable to learn water in indirect ways, results in prediction failure. This logic tells us that the most critical thing is securement of the training dataset which contains as many as possible kinds of solvents and solutes. Still, since there are 418 solutes and 91 solvents in the dataset we use[34] and they can make 38,038 possible mixtures, we expect our model and the MNSOL guarantee similar precision levels with the random CV for numerous mixture systems.

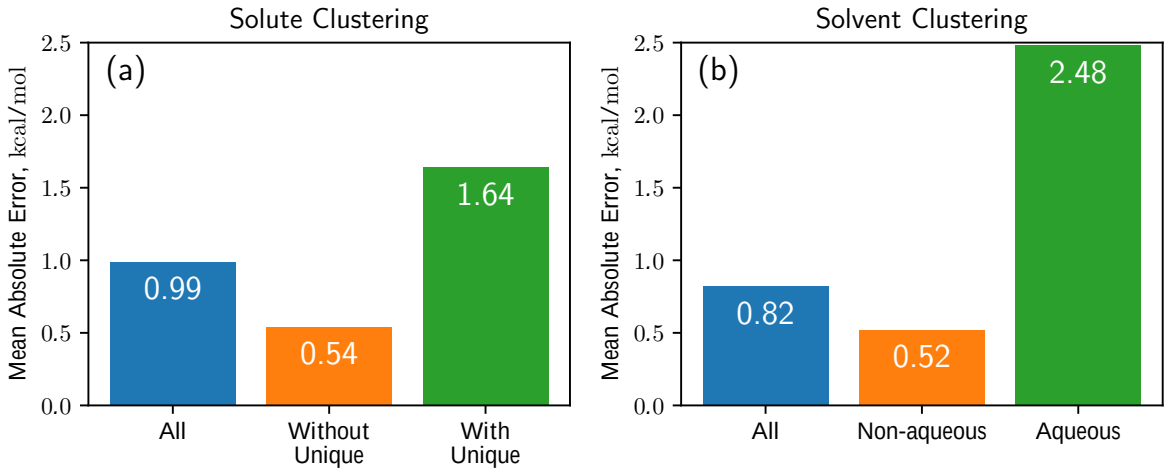

Figure 6: Results of cross-validation tasks using K-mean clustering algorithm for (a) solutes and (b) solvents. We conclude that unique substructures in the given compounds are the main cause of the decline in prediction accuracy. Each encoder network includes a BiLSTM layer and we use the same hyperparameters which are optimized in the random CV task.

4 Conclusions

In the present study, we introduced a QSPR regression neural network for solvation energy estimation that is inspired by NLP. The proposed model has two separate encoder neural networks for solvents and solutes and a predictor neural network. Each encoder neural network is designed to encode the chemical structure of an input compound into the feature

vector of a specific size. The encoding procedure is accomplished using Mol2Vec embedding model[30] and recurrent neural networks with the attention mechanism.[36–38] The predictor neural network with fully-connected MLP calculates the solvation free energy of a given solvent-solute mixture using the feature vectors from encoders.

We performed extensive calculations on 2495 experimental values of solvation energies taken from the MNSOL database.[34] We obtained mean averaged errors in solvation free energy of the BiLSTM model are ± 0.64 kcal=mol for aqueous solutions and ± 0.24 kcal=mol for non-aqueous solutions. Our results demonstrate that the encoder-predictor neural network exhibits excellent prediction accuracy which is more precise than several DFT calculations with SMx and SMD implicit solvent models,[2] while the MLP model without recurrent layers shows relatively deficient performances. The score matrix taken from the attention mechanism gives us an interaction map between atoms and substructure; our model does provide not only a simple estimation of target property but offers pieces of information about which substructures are play a dominant role in solvation processes. Decline of performances in the cluster CV suggestes the importance of preparation of the ML database even though it still performed comparable predictions with both MD[7] and *ab initio*[5] solvation models.

One of the most typical advantages of ML is flexibility, and a single model can be used to learn and predict various databases.[16] Also, our model may be applied to predict various chemical, physical, or biological properties especially focused on interactions between more than two different chemical species. One of the possible applications that we can consider is the prediction of chemical affinity and potentially the possibility of various chemical reactions.[48] Room-temperature ionic liquids might be another potential research topic as the interplay between molecular ions dominates their various properties, e.g., toxicity[49] or electrochemical properties in supercapacitors.[50,51] We expect the proposed model will be helpful for many further studies, not only localized in the prediction of solvation energies.

5 Conflicts of Interest

There are no conflicts to declare.

6 Acknowledgements

This research was supported by Creative Materials Discovery Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (NRF-2017M3D1A1039556).

References

(1) Cramer, C. J.; Truhlar, D. G.; Marenich, A. V.; Kelly, C. P.; Olson, R. M. *Journal of Chemical Theory and Computation* **2007**, *3*, 2011–2033.

(2) Marenich, A. V.; Cramer, C. J.; Truhlar, D. G. *Journal of Chemical Theory and Computation* **2013**, *9*, 609–620.

(3) Dupont, C.; Andreussi, O.; Marzari, N. *Journal of Chemical Physics* **2013**, *139*, 214110.

(4) Sundararaman, R.; Goddard, W. A. *Journal of Chemical Physics* **2015**, *142*, 064107.

(5) Kromann, J. C.; Steinmann, C.; Jensen, J. H. *Journal of Chemical Physics* **2018**, *149*, 104102.

(6) Shivakumar, D.; Williams, J.; Wu, Y.; Damm, W.; Shelley, J.; Sherman, W. *Journal of Chemical Theory and Computation* **2010**, *6*, 1509–1519.

(7) Mohamed, N. A.; Bradshaw, R. T.; Essex, J. W. *Journal of Computational Chemistry* **2016**, *37*, 2749–2758.

(8) Misin, M.; Fedorov, M. V.; Palmer, D. S. *The Journal of Physical Chemistry B* **2016**, *120*, 975–983.

(9) Genheden, S. *Journal of Computer-Aided Molecular Design* **2017**, *31*, 867–876.

(10) Cherkasov, A.; Muratov, E. N.; Fourches, D.; Varnek, A.; Baskin, I. I.; Cronin, M.; Dearden, J.; Gramatica, P.; Martin, Y. C.; Todeschini, R.; Consonni, V.; Kuz'min, V. E.; Cramer, R.; Benigni, R.; Yang, C.; Rathman, J.; Terfloth, L.; Gasteiger, J.; Richard, A.; Tropsha, A. *Journal of Medicinal Chemistry* **2014**, *57*, 4977–5010.

(11) Mitchell, J. B. O. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2014**, *4*, 468–481.

(12) Delaney, J. S. *Journal of Chemical Information and Computer Sciences* **2004**, *44*, 1000–1005.

(13) Cereto-Massagué, A.; Ojeda, M. J.; Valls, C.; Mulero, M.; Garcia-Vallvé, S.; Pujadas, G. *Methods* **2015**, *71*, 58–63.

(14) Kearnes, S.; Riley, P. *Journal of Computer-Aided Molecular Design* **2016**, *30*, 595–608.

(15) Coley, C. W.; Barzilay, R.; Green, W. H.; Jaakkola, T. S.; Jensen, K. F. *Journal of Chemical Information and Modeling* **2017**, *57*, 1757–1772.

(16) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. *Chemical Science* **2018**, *9*, 513–530.

(17) Lusci, A.; Pollastri, G.; Baldi, P. *Journal of Chemical Information and Modeling* **2013**, *53*, 1563–1575.

(18) Schütt, K. T.; Arbabzadah, F.; Chmiela, S.; Müller, K. R.; Tkatchenko, A. *Nature Communications* **2017**, *8*, 13890.

(19) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. *ACS Central Science* **2018**, *4*, 268–276.

(20) Chen, H.; Engkvist, O.; Wang, Y.; Olivecrona, M.; Blaschke, T. *Drug Discovery Today* **2018**, *23*, 1241–1250.

(21) Zheng, S.; Yan, X.; Yang, Y.; Xu, J. *Journal of Chemical Information and Modeling* **2019**,

(22) Schmidhuber, J. *Neural Networks* **2015**, *61*, 85–117.

(23) Okamoto, Y.; Kubo, Y. *ACS Omega* **2018**, *3*, 7868–7874.

(24) Faber, F.; Lindmaa, A.; von Lilienfeld, O. A.; Armiento, R. *International Journal of Quantum Chemistry* **2015**, *115*, 1094–1101.

(25) Sato, H. *Physical Chemistry Chemical Physics* **2013**, *15*, 7450.

(26) Skyner, R. E.; McDonagh, J. L.; Groom, C. R.; van Mourik, T.; Mitchell, J. B. O. *Physical Chemistry Chemical Physics* **2015**, *17*, 6174–6191.

(27) Barton, A. F. M. *Chemical Reviews* **1975**, *75*, 731–753.

(28) Ran, Y.; Yalkowsky, S. H. *Journal of Chemical Information and Computer Sciences* **2001**, *41*, 354–357.

(29) Goh, G. B.; Hodas, N. O.; Siegel, C.; Vishnu, A. *Arxiv preprint* **2017**, arxiv:1712.02034.

(30) Jaeger, S.; Fulle, S.; Turk, S. *Journal of Chemical Information and Modeling* **2018**, *58*, 27–35.

(31) Reichardt, C.; Welton, T. *Solvents and Solvent Effects in Organic Chemistry*; Wiley-VCH Verlag GmbH & Co. KGaA: Weinheim, Germany, 2010.

(32) Marenich, A. V.; Ho, J.; Coote, M. L.; Cramer, C. J.; Truhlar, D. G. *Physical Chemistry Chemical Physics* **2014**, *16*, 15068–15106.

(33) Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. *Arxiv preprint* **2013**, arxiv:1310.4546.

(34) Marenich, A. V.; Kelly, C. P.; Thompson, J. D.; Hawkins, G. D.; Chambers, C. C.; Giesen, D. J.; Winget, P.; Cramer, C. J.; Truhlar, D. G. Minnesota Solvation Database version 2012. University of Minnesota, Minneapolis, 2012.

(35) Pennington, J.; Socher, R.; Manning, C. Glove: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Stroudsburg, PA, USA, 2014; pp 1532–1543.

(36) Bahdanau, D.; Cho, K.; Bengio, Y. *Arxiv preprint* **2014**, arxiv:1409.0473.

(37) Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhutdinov, R.; Zemel, R.; Bengio, Y. *Arxiv preprint* **2015**, arxiv:1502.03044.

(38) Luong, M.-T.; Pham, H.; Manning, C. D. *Arxiv preprint* **2015**, arxiv:1508.04025.

(39) Asgari, E.; Mofrad, M. R. K. *PLOS ONE* **2015**, *10*, e0141287.

(40) Morgan, H. L. *Journal of Chemical Documentation* **1965**, *5*, 107–113.

(41) Rogers, D.; Hahn, M. *Journal of Chemical Information and Modeling* **2010**, *50*, 742–754.

(42) Schuster, M.; Paliwal, K. *IEEE Transactions on Signal Processing* **1997**, *45*, 2673–2681.

(43) Bengio, Y.; Simard, P.; Frasconi, P. *IEEE Transactions on Neural Networks* **1994**, *5*, 157–166.

(44) Hochreiter, S.; Schmidhuber, J. *Neural Computation* **1997**, *9*, 1735–1780.

(45) Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. *Arxiv preprint* **2014**, arxiv:1412.3555.

(46) Winter, R.; Montanari, F.; Noé, F.; Clevert, D.-A. *Chemical Science* **2019**, *10*, 1692–1701.

(47) Mayr, A.; Klambauer, G.; Unterthiner, T.; Steijaert, M.; Wegner, J. K.; Ceulemans, H.; Clevert, D.-A.; Hochreiter, S. *Chemical Science* **2018**, *9*, 5441–5451.

(48) Engkvist, O.; Norrby, P.-O.; Selmi, N.; hong Lam, Y.; Peng, Z.; Sherer, E. C.; Amberg, W.; Erhard, T.; Smyth, L. A. *Drug Discovery Today* **2018**, *23*, 1203–1218.

(49) Pham, T. P. T.; Cho, C.-W.; Yun, Y.-S. *Water Research* **2010**, *44*, 352–372.

(50) Jo, S.; Park, S.-W.; Shim, Y.; Jung, Y. *Electrochimica Acta* **2017**, *247*, 634–645.

(51) Noh, C.; Jung, Y. *Physical Chemistry Chemical Physics* **2019**, *21*, 6790–6800.