Artificial Intelligence in the Life Sciences 1 (2021) 100021

Contents lists available at ScienceDirect

Artificial Intelligence in the Life Sciences

journal homepage: www.elsevier.com/locate/ailsci

Research Article

Dealing with a data-limited regime: Combining transfer learning and transformer attention mechanism to increase aqueous solubility prediction performance

Magdalena Wiercioch [a,b,*], Johannes Kirchmair [b]

[a] Department of Applied Computer Science, Faculty of Physics, Astronomy and Applied Computer Science, Jagiellonian University, Łojasiewicza 11, Kraków 30-348, Poland

[b] Department of Pharmaceutical Chemistry, Faculty of Life Sciences, University of Vienna, Vienna 1090, Austria

ARTICLE INFO

Keywords:
Aqueous solubility
Deep Learning
Cheminformatics
Transformer model
Drug discovery
Regression

ABSTRACT

Aqueous solubility is a key chemical property that drives various processes in chemistry and biology. Its computational prediction is challenging, as evidenced by the fact that it has been a subject of considerable interest for several decades. Recent work has explored fingerprint-based, feature-based and graph-based representations with different machine learning and deep learning methodologies. In general, many traditional methods have been proposed, but they rely heavily on the quality of the rule-based, hand-crafted features. On the other hand, limitations in the quality of aqueous solubility data become a handicap when training deep models. In this study, we have developed a novel structure-aware method for the prediction of aqueous solubility by introducing a new deep network architecture and then employing a transfer learning approach. The model was proven to be competitive, obtaining an RMSE of 0.587 during both cross-validation and a test on an independent dataset. To be more precise, the method is evaluated on molecules downloaded from the Online Chemical Database and Modeling Environment (OCHEM). Beyond aqueous solubility prediction, the strategy presented in this work may be useful for modeling any kind of (chemical or biological) properties for which there is a limited amount of data available for model training.

1. Introduction

Significant progress has been made in the prediction of quantitative structure-property relationships (QSPRs) for small organic molecules, in particular in machine learning [1]. In the context of small-molecule drug discovery and the development of agrochemicals, cosmetics, functional foods, etc., aqueous solubility has been a field of active research for several decades now as it determines, among many other things, a compound's ability to cross the biological membranes and, hence, its capacity to induce desired biological effects (e.g. in the context of pharmacology) and/or undesired biological effects (in the context of toxicology) in organisms. The development of highly efficacious small-molecules that are sufficiently soluble in water remains a challenging task because aqueous solubility and biological activity are often observed to be in an indirect proportional relationship [2].

Aqueous solubility is usually reported as $\log S$, the logarithm of base 10 of the aqueous solubility in mol/L. However, aqueous solubility is not an easy molecular property to measure accurately, for which reason various approaches for measuring it are in existence and use. Data quality varies substantially between different data sources, and results obtained with different technologies are not always comparable. This is the reason why we can in part observe only a moderate reproducibility of aqueous solubility measurements. On the whole, there are varied sources of data, and the exact information connected with aqueous solubility methodology is not always given [3]. In consequence, the lack of high-quality datasets for computational aqueous solubility prediction is problematic [4,5]. Please note that the reported prediction root mean square errors (RMSE) ranges 0.6–1.4 log, with the average at 0.9 log [6].

Classical machine learning techniques have demonstrated remarkable performance in QSPR modeling, including aqueous solubility prediction. Generally, classical machine learning approaches can be split into two stages. The first stage aims to encode the input data and extract the most important features or properties connected with the molecule. In consequence, one gets a molecular representation. The second stage usually applies an algorithm that takes a calculated molecular representation as an input and returns some response. In addition, while building

* Corresponding author at: Department of Applied Computer Science, Faculty of Physics, Astronomy and Applied Computer Science, Jagiellonian University, Łojasiewicza 11, 30-348 Kraków, Poland.

E-mail addresses: magdalena.wiercioch@uj.edu.pl, mgkwiercioch@gmail.com (M. Wiercioch), johannes.kirchmair@univie.ac.at (J. Kirchmair).

https://doi.org/10.1016/j.ailsci.2021.100021
Received 4 October 2021; Received in revised form 20 November 2021; Accepted 23 November 2021
Available online 26 November 2021

machine learning models, one uses the train-test split as a method for evaluating the performance of a machine learning algorithm.

To address the problem of aqueous solubility prediction, several methods have been proposed [7,8]. The first technique appeared in 1924 in work by Fühner [9]. Fühner observed that the addition of successive methylene groups causes a decrease of aqueous solubility. Subsequently, scientists discovered that various molecular features affect aqueous solubility [10–12]. Over the years, different models were applied. Erickson used linear regression to investigate the aqueous solubility of homologous series of organic molecules [13]. He observed that $\log S$ was a negative rectilinear function of alkyl chain length of homologous series. Later, Hewitt et al. analyzed multilinear regressions performance to predict aqueous solubility [14]. Their conclusions were surprising since a simple linear regression approach proved to be superior over more complex modeling methods at that time. The authors noticed that the statistical fit of the training and validation data was reasonably good (R^2 values of 0.74 and 0.67, respectively). Finally, the root mean square error (RMSE) value for the test set of 0.95 was obtained. With Random Forest, Palmer et al. studied the prediction of aqueous solubility for a data set of compounds that are solid at room temperature [15]. Their method has been shown to perform comparably to other methods, including some that require 3D structure calculation. To specify, the authors predicted the log molar solubility values for the molecules in the test set with RMSE of 0.69. Also, Lind and Maltseva have demonstrated that Support Vector Machines with a Tanimoto similarity kernel detects aqueous solubility with accuracy, expressed in terms of root mean square error, produce comparable results than other reported approaches [16]. They reported that the cross-validation on a diverse data set resulted in a RMSE of 0.62 and R^2 of 0.88. Besides, artificial neural networks have been utilized to predict aqueous solubility [17]. The goal was to take advantage of their ability to approximate non-linear functions. For instance, Erić et al. proposed a methodology for the automatic adjustment of descriptor's relative importance in counter-propagation shallow neural networks in order to predict aqueous solubility [18]. The performance of their final model based on seven descriptors for prediction of aqueous solubility was satisfactory since the authors reported an RMSE of 0.679 on test dataset.

Very recently, deep neural networks (DNNs) [19–21] have yielded impressive performance in molecular and other materials property prediction [22–25]. While the classical machine learning approaches require hand-crafted molecular descriptors as inputs, DNNs can employ more lossless formats and train models in an end-to-end fashion in order to predict the target endpoints. Besides, there are many different formats of input representation [26]. The standard type is a topological graph that describes the connectivity of the atoms. Another example is SMILES strings (Simplified Molecular Input Line Entry System). Broadly speaking, SMILES is a text-based format where chemical species are mapped to single ASCII strings [27].

Deep learning systems have improved the state-of-the-art in aqueous solubility prediction. For instance, Lusci et al. proposed an architecture that uses a recurrent neural network and molecular structures converted into directed, acyclic graphs. Their methodology, called UG-RNN, sometimes outperformed current then state-of-the-art techniques [28]. For instance, UG-RNN achieves RMSE of 0.96 on the intrinsic solubility data set. Another example is the work of Wu et al., where the authors constructed a topology-based multi-task deep learning strategy (MT-DNN) and achieved some of the most accurate predictions of aqueous solubility (RMSE of 0.649) and the partition coefficient [29]. Also, Liu et al. presented a competitive approach (Chemi-Net) based on convolutional neural networks to predict aqueous solubility and other ADME properties [30]. They demonstrated that Chemi-Net beats the competitive models and achieves an R^2 of 0.585. Finally, in 2020 Tang et al. [31] proposed a self-attention-based message passing neural network to identify the relationship between molecular solubility and structure. Their method obtained an RMSE of 0.661 (for aqueous solubility prediction) on a small collection of compounds.

Table 1
Important notations used in this study.

Notation	Description
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	an undirected graph
$\mathcal{V}$	set of nodes in a graph $\mathcal{G}$
$\mathcal{E}$	set of edges in a graph $\mathcal{G}$
$\tilde{N}_i$	neighborhood set of vertex $v_i \in \mathcal{V}$
a_i	attributes for vertex $v_i \in \mathcal{V}$
$e_{i,j}$	attributes for edge $(v_i, v_j) \in \mathcal{E}$
d_g	input dimensionality of vertices
$\tilde{d}_g$	output dimensionality of vertices
$\tilde{e}$	dimensionality of edges
F	number of layers
h_i^t	state vector of node $v_i \in \mathcal{V}$ at layer t; $h_i^t \in \mathbb{R}^{d_g}$
$\alpha_{i,j}$	attention coefficient for edge $(v_i, v_j) \in \mathcal{E}$

Although the performance of computational models for the prediction of aqueous solubility has greatly improved by employing deep-learning architectures, there remains room for improvement. First, among the deep learning approaches, graph neural network (GNN)-based methods attract significant attention because of their ability to model interactions between atoms. The idea is to treat a molecule as a molecular graph where atoms are associated with nodes. However, simple operations such as summation and average may not capture various characteristics. Thus, more studies are required to analyze the contributions of different parts of the compound to make decisions. Second, a great majority of deep learning aqueous solubility prediction models that can be found in the literature are rather shallow networks having around seven layers [32]. Obviously, the performance could be enhanced by going wider or deeper [33,34]. Unfortunately, such modifications are constrained by the limited number of molecules with reliable aqueous solubility information.

To address the above-mentioned problems, in this study, we propose a deep learning architecture that employs both a text representation of molecules and a graph-based strategy with an attention mechanism to learn and predict aqueous solubility. The core contributions of this work are as follows. (1) We treat aqueous solubility prediction as a translation problem. Our architecture represents an encoder-decoder design. However, in order to learn a latent representation, our main encoder consists of two subencoders, i.e., a graph encoder and an encoder that employs a Transformer. We call this architecture M2M. (2) To address the problem of the availability of limited amounts of high-quality data and to increase the aqueous solubility prediction performance, transfer learning is incorporated. Therefore, we first pretrain the model on pKa dataset that consists of more than 6000 chemical compounds. Then, the learned knowledge is transferred to be used on a smaller water solubility dataset. The final architecture is called TunedM2M. (3) We demonstrate that the proposed method outperforms the state-of-the-art approaches in aqueous solubility prediction. Since we found the most recent and relevant work connected with aqueous solubility prediction was conducted by Tang et al., our model performance is evaluated against the dataset used in their paper.

2. Materials and methods

In this section, we describe the methodology presented in the work, the datasets, and the algorithms employed. Table 1 summarizes all notations used throughout this paper.

2.1. Model design

Fig. 1 provide schematic diagram of the workflow of the proposed M2M and TunedM2M that utilizes the pre-trained M2M.

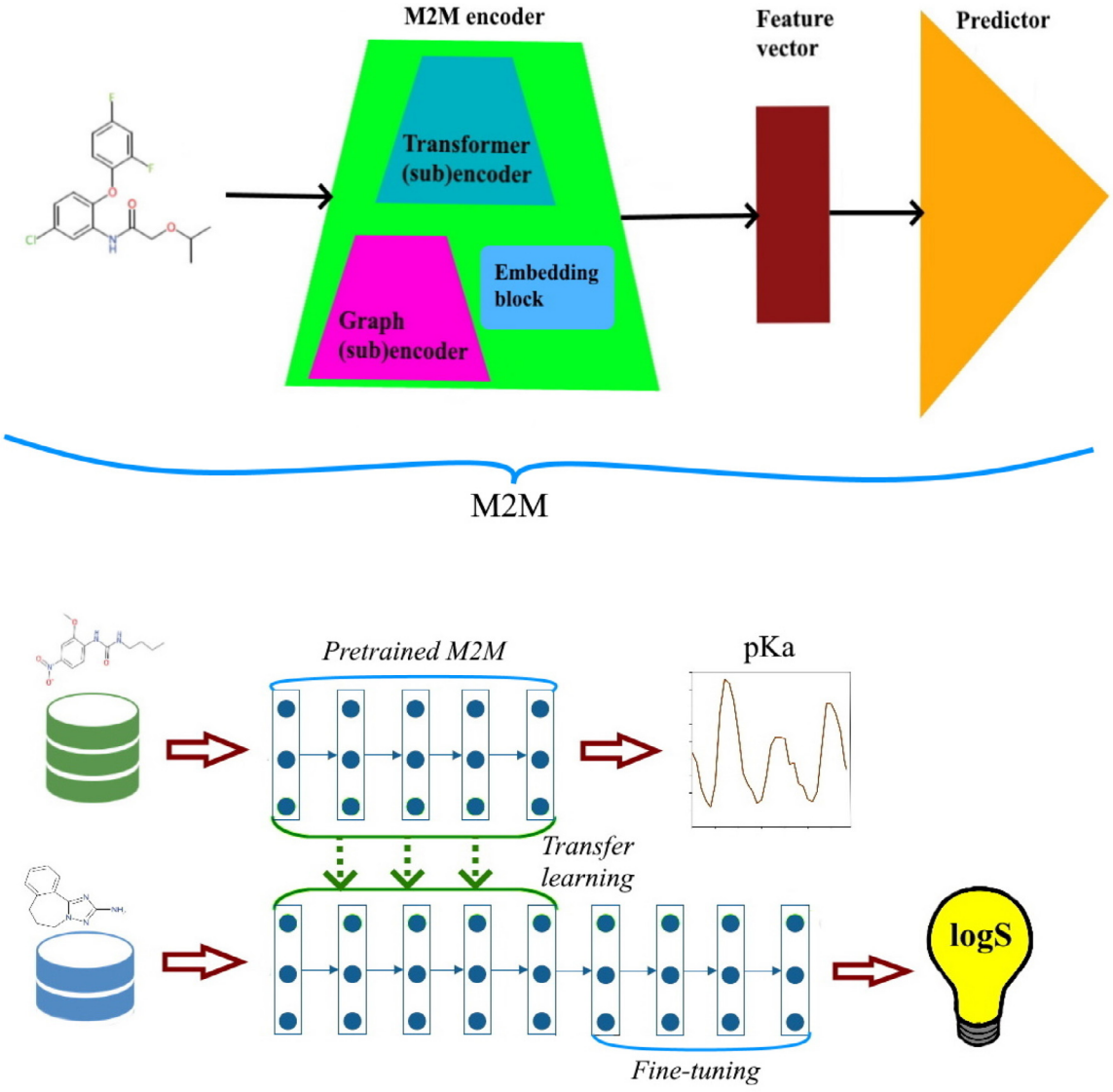


Fig. 1. Architecture of M2M and TunedM2M.

2.2. Datasets characteristics

Firstly, a pKa dataset was applied to pretrain our M2M model. Then, $\log S$ values related to aqueous solubility [35] were used to M2M transfer learning that we call TunedM2M.

Dataset for developing the pretrained M2M

We use dataset that contains pKa data for 7911 chemical compounds, and apply it to pretrain M2M. The data come as SMILES. One may obtain the chemicals from the DataWarrior [36] application folder. According to the analysis performed using Toxprint chemotypes [37], the chemical structures have a high diversity of functional groups. Therefore, they are sufficient to support our research, i.e. pretrain M2M. For the purpose of this study, preprocessing of the chemical structures was performed. More specifically, in the first step, the minor components of any multi-component compounds were stripped. Second, duplicated structures were removed from the set. Also, we excluded inorganic chemicals and mixtures. The final, processed pKa dataset consists of 6245 chemicals with measured pKa values. Fig. 3 illustrates the similarity map for pKa dataset. The measured values are highly diverse (Fig. 3).

Dataset for TunedM2M

The dataset we employ in this paper for M2M transfer learning was obtained from OCHEM [35]. It consists of 1311 molecules, which were randomly assigned to a training set (80% of all molecules), a validation set (10%), and test set (10%), following the procedure described in Tang's paper. In addition, in order to ensure that there was no overlap between the data used for pretraining and transfer learning, we checked for matches in both datasets. The corresponding chemical compounds were removed. In addition, Fig. 4 depicts the similarity map for solubility dataset. As one may note, the solubility properties are diverse among the molecules. Also, in order to explore the chemical space, we apply PCA to project the final representation obtained from TunedM2M into a three dimensional space. In addition, the Euclidean distances to the centroid were calculated. The result is shown in Fig. 5. The red star is the centroid of the space. Chemical compounds that are in the 75 percentiles closest to the centroid are colored in blue, whereas those further apart are colored in brown. The analysis indicate that the number of outliers is relatively small.

2.3. Graph neural networks

Graph neural networks (GNNs) were introduced by Scarselli et al. [38] in 2009. According to the concept, a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined by its vertices $\mathcal{V}$ and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Moreover, each vertex $v_i \in \mathcal{V}$ is naturally associated with a vector of attributes a_i, and each edge $(v_i, v_j) \in \mathcal{E}$

Table 2
The atom (graph node) attributes used for molecular representation.

Attributes	Explanation
atom type	C, O, H, N, O, P, … : one-hot encoding
hybridization	sp, sp^2, sp^3, sp^3d : one-hot encoding
chirality	R (Rectus), S (Sinister), N (None) : one-hot encoding
radical electrons	0 (no), 1 (yes)
aromatic	0 (no), 1 (yes)
implicit valence	0 (no), 1 (yes)
acceptor	0 (no), 1 (yes)
donor	0 (no), 1 (yes)

Table 3
The bond (graph edge) attributes used for molecular representation.

Attributes	Explanation
type	single, double, triple, aromatic : one-hot encoding
part of ring	0 (no), 1 (yes)
part of conjugation	0 (no), 1 (yes)
chirality	E, Z : one-hot encoding

is naturally associated with a vector of attributes $e_{i,j}$. There is also defined a neighborhood function $\tilde{N}_i = \{v_j : (v_j, v_i) \in \mathcal{E}\}$ that assigns a set of neighbors $\tilde{N}_i$ to each vertex $v_i \in \mathcal{V}$. The primarily goal of GNNs is to learn a state vector h_i that is associated with each vertex $v_i \in \mathcal{V}$. In the beginning the vector is initialized as $h_i^0 = a_i$. The state of all the vertices is then updated until the stopping criterion is reached. The update procedure is based on the assumption that each vertex communicates with its neighbors by sending and receiving messages. Therefore, the state h_i^t of vertex v_i at layer t depends on its state at the previous layer, h_i^{t-1}, and all messages which came from neighbors $v_j \in \tilde{N}_i$. In general, it can be defined as follows:

$$h_i^t = Up(h_i^{t-1}, Agg(\{M(v_i, v_j, t) : v_j \in \tilde{N}_i\})), \tag{1}$$

where Up is a function that updates the state, Agg denotes a neighborhood aggregation function, and M is a message function.

2.4. Graph attention mechanism

Attention mechanism in deep learning was first proposed in 2015 by Bahdanau et al. [39] to address the common problem in natural language processing, i.e., translation of long sequences. In fact, attention allows to assigning a learnable weight to pairs of elements such as nodes in a graph to focus on the most relevant parts of the graph. More formally, attention is defined as a function

$$g : v_i \times \tilde{N}_i \rightarrow [0, 1]$$

that for a given vertex $v_i \in \mathcal{V}$ projects each vertex in $\tilde{N}_i$ to a relevance score that informs how much attention is given to a particular neighbor vertex. If one considers an attention-based graph neural network, it means that the function Agg in Eq. (1) employs an attention mechanism. In this case, we add an extra attention coefficient $\alpha_{i,j}$ to modify the weight of h_j^t, where $v_j \in \tilde{N}_i$.

2.5. Initial featurization

The initial atom and bond features used in our study are shown in Tables 2 and 3, respectively. The M2M's initial attributes a_i are the atom features for the vertex $v_i \in \mathcal{V}$, while the M2M's initial edge features $e_{i,j}$ are the bond features for an edge $(v_i, v_j) \in \mathcal{E}$. In order to extract these features, RDKit [40] was used. Finally, the initial representation of the atom is a 130-dimensional vector, and the input representation of the edge is a 8-dimensional vector.

2.6. Molecular transformer

Transformer [41] architecture was originally proposed for various sequence-to-sequence tasks, including machine translation, and language understanding [42,43]. In fact, the huge success of transformer-based models is attributed to the multihead self-attention component that enables the network to capture contextual information from the entire sequence. A multihead attention layer includes several scaled dot attention layers that run in parallel and are concatenated at the end. In case of the ith attention head and for a matrix $H \in \mathbb{R}^{l \times d}$, the self-attention maps H into three matrices: the query matrix $Q = HW_Q$, the key matrix $K = HW_K$ and the value matrix $V = HW_V$, where l is the length of the input sequence, d is the dimension of the input sequence, W_K, W_K, W_V are learnable parameters. Then, the attention scores can be expressed as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V. \tag{2}$$

2.7. Molecular embedding

Mikolov et al. [44] introduced *Word2Vec* to learn word embeddings in the natural language processing field. The idea behind this method follows the assumption that words that appear frequently in similar contexts share a semantic relationship. The method achieves an embedding for each word in the training corpus by training a fully-connected shallow neural network to predict either the target word given the context (the CBOW model) or the context given a target word (the Skip-Gram model). It shows that *Word2Vec* has inspired much research in various domains such as bioinformatics [45]. It was also employed for the representation of chemical compounds [46].

2.8. M2M and TunedM2M architecture

According to the notion of transfer learning, we divide our approach into two parts: a source task and a target task. In our work, the source task is how to represent the molecules and then predict pKa. However, the main target task is to predict aqueous solubility expressed as $\log S$. In the following, we describe the proposed methodology in detail.

2.8.1. M2M

In order to accomplish the source and target tasks, we design an architecture that we call M2M. The model aims to learn feature representation using molecules as described in Subsection *Dataset for developing the pretrained M2M*, and then predict pKa. Fig. 2 illustrates the proposed model. As visualized in this figure, our architecture consists of three main components: the encoder, the embedding layer, and the prediction layer. With respect to representation learning, the core part of the model is the M2M encoder, which includes the embedding block and two (sub)encoders, namely graph (sub)encoder and transformer (sub)encoder. Each of these elements has an inevitable impact on the final form of the M2M encoder and involves several operations that can be summarized as follows.

(1) *Graph (sub)encoder* aims to generate a low-dimensional, hidden representation for atoms in a given chemical compound. Firstly, in order to construct a graph (sub)encoder, we assume that a molecule is represented as a graph, as discussed in Subsection *Graph neural networks*. In this setting, vertices are associated with atoms, and edges correspond to bonds. Additionally, we propose a set of features to provide initial molecular attributes, as explained in Subsection *Initial featurization*. To build the graph, we employ the deep graph library (DGL) [47]. Our graph (sub)encoder includes a stack of the single (sub)encoder layers. In addition, each layer aggregates the attribute information from the neighboring vertices of a target vertex. Fig. 2a illustrates a single layer of our graph (sub)encoder. Formally, it is defined as follows:

$$h_i^{t+1} = \sigma(\sum_{k \in \tilde{N}_t} \alpha_{ik} W h_k^t), \tag{3}$$

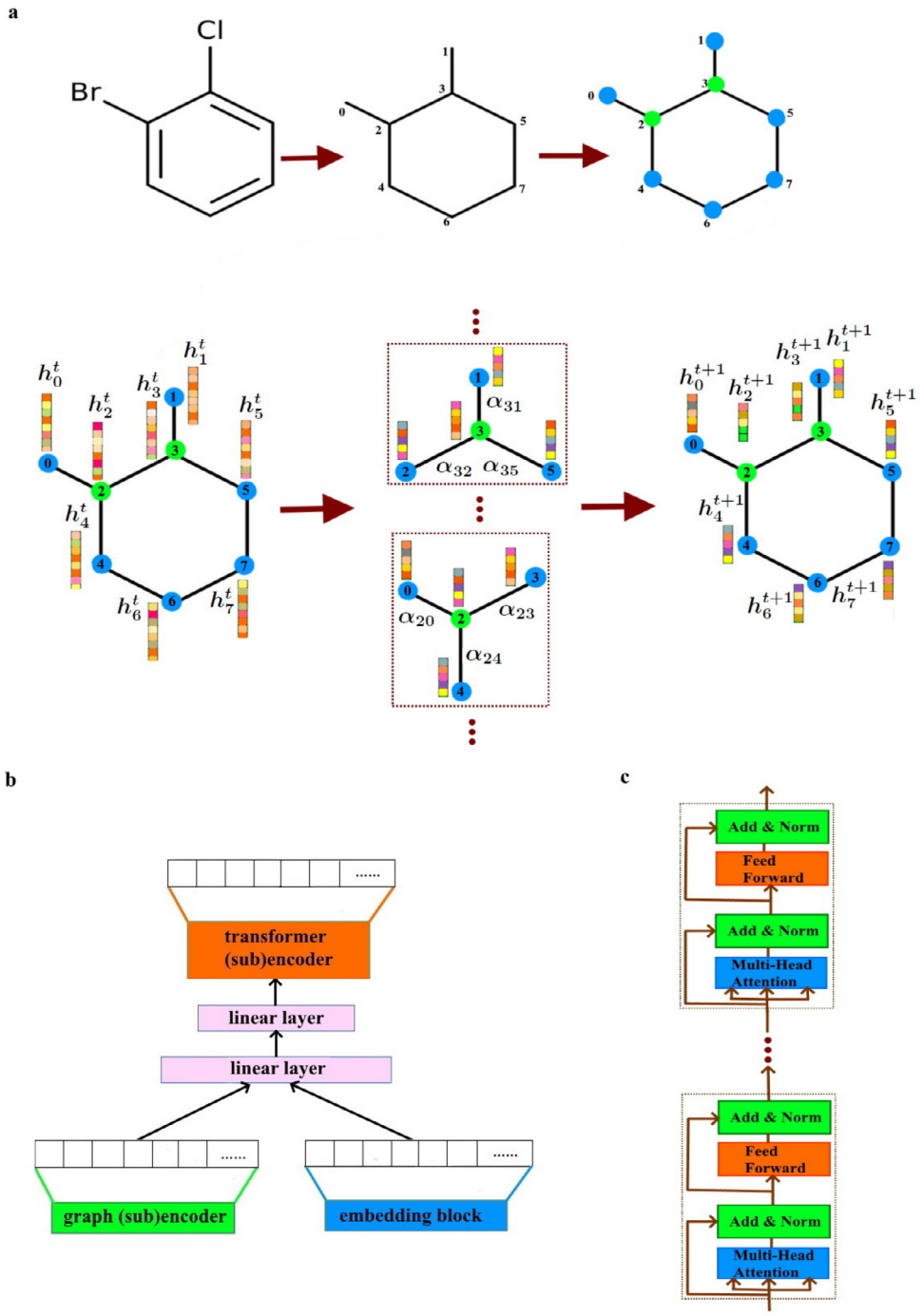


Fig. 2. M2M network architecture.

where $W \in \mathbb{R}^{\tilde{d}_g \times d_g}$ is a weight matrix. Furthermore, the vital component of each single layer is an attention coefficient that is associated with an attention mechanism introduced in Subsection *Graph attention mechanism*. More specifically, during the process, features are concatenated and parameterized by a weight vector $\vec{a} \in \mathbb{R}^{2\tilde{d}_g + \tilde{e}}$. Also, nonlinearity is provided by *LeakyReLU* function. It is interesting to note that we perform normalization using a softmax function denoted by σ. All these operations can be expressed as:

$$\alpha_{ij} = \frac{\exp(\sigma(\vec{a}^T[W\vec{h_i}||W\vec{h_j}||e_{ij}]))}{\sum_{k\in\tilde{N}_i}\exp(\sigma(\vec{a}^T[W\vec{h_i}||W\vec{h_k}||e_{ik}]))}. \tag{4}$$

What is more, to capture multiple types of relationships between vertices, multihead attention is used. As a result, we concatenate the different representations learned by different heads in the hidden layer. Then, they are averaged on the final layer of our graph (sub)encoder as follows:

$$h_i^{t+1} = \sigma\left(\frac{1}{M}\sum_{m=1}^{M}\sum_{k\in\tilde{N}_i}\alpha_{ik}^m W^m h_k^t\right), \tag{5}$$

where M is the number of heads involved in the multi-head attention mechanism.

(2) *Embedding block* is designed to obtain a vector representation of a molecule based on SMILES. Therefore, in order to transform the molecule SMILES to vector representation, we train a *Word2vec* model. Specifically, we follow a Skip-gram configuration. Our goal is to maxi-

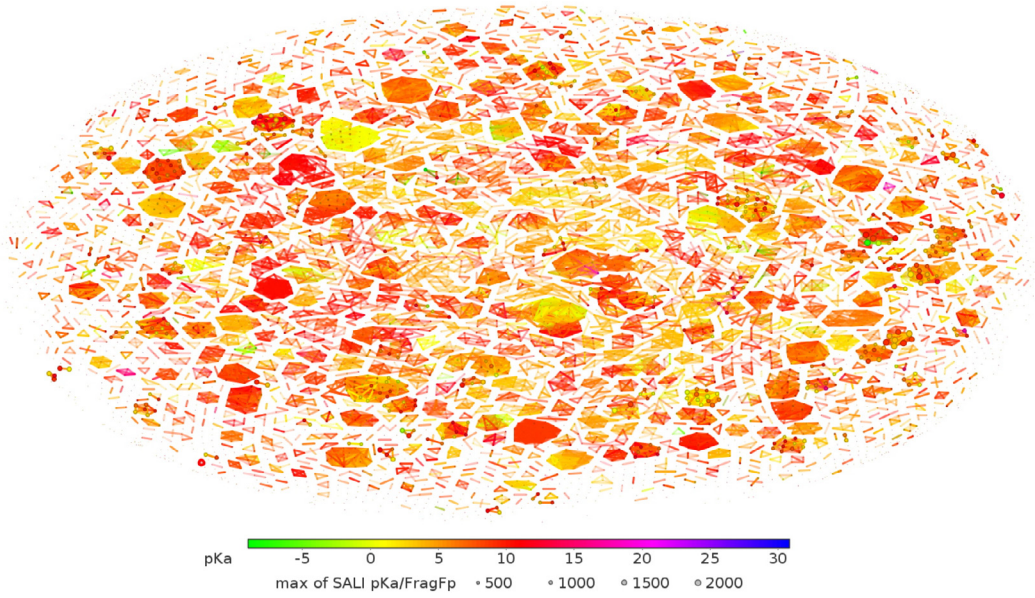


Fig. 3. Similarity map for pKa dataset.

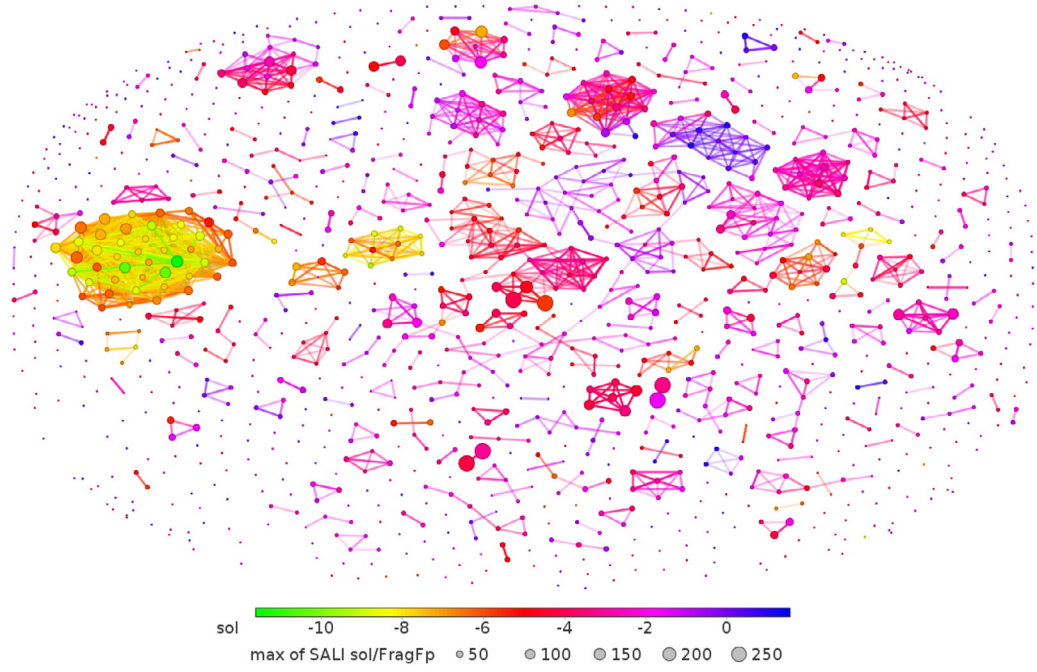


Fig. 4. Similarity map for solubility dataset.

mize the average log probability

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{-c\le m\le c, m\neq 0}\log P(s_{i+m}|s_i), \tag{6}$$

where N is the size of training set, $s_1, s_2, \ldots, s_N$ are the training SMILES and c is the size of the training context. Moreover, $P(s_{i+m}|s_i)$ is calculated using the softmax function

$$P(s_{i+m}|s_i) = \frac{\exp((\tilde{u}_{i+m})^T u_i)}{\sum_{k=1}^{S}\exp((\tilde{u}_k)^T u_i)},$$

where S is the number of SMILES in the vocabulary, U_s, $\tilde{U}_s$ are the input and output vector representations of s.

(3) *Transformer (sub)encoder* takes as input representations $\tilde{r}_i''$ (see Fig. 2b)

$$\tilde{r}_i'' = K_2\tilde{r}_i'$$

$$\tilde{r}_i' = K_1\tilde{r}_i$$

$$\tilde{r}_i = [s_i'||h_i^F]$$

where s_i' is a vector representation of SMILES returned by the embedding block, K_1 and K_2 are weight matrices to perform linear transformations on $\tilde{r}_i$ and $\tilde{r}_i'$ respectively. Then, we concatenate the outputs of k attention heads and get the output that is considered as a linear projection of the

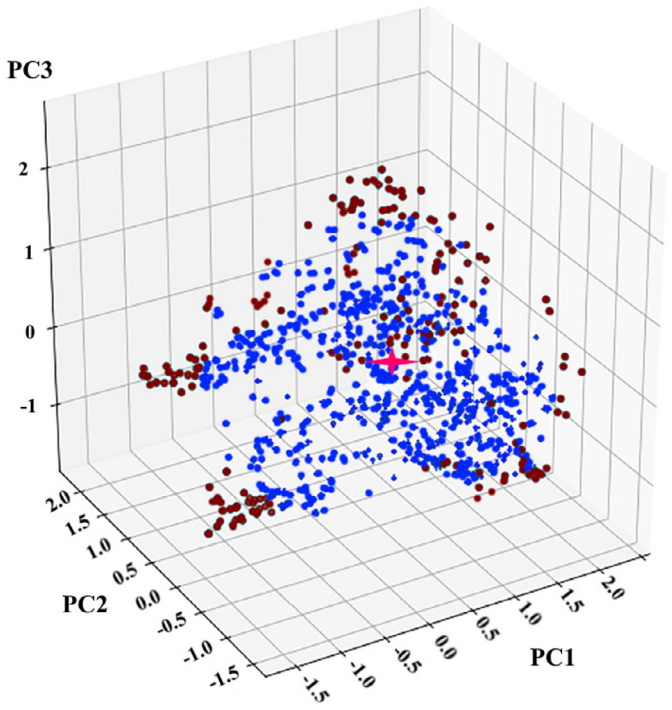


Fig. 5. PCA scores plot for the solubility data. The representation returned by TunedM2M is projected onto three axes.

concatenated representations as follows:

$$Multihead(Q, K, V) = Concat(head_1, \ldots, head_k)W_o, \quad (7)$$

where W_o is an output projection matrix, and $head_i = Attention(Q_i, K_i, V_i)$, as Eq. (2) indicates (see Fig. 2c). Finally, a position-wise feed-forward neural network is applied to the output of the network that can be expressed as follows:

$$FFN(out) = \max(0, outW_1 + b_1)W_2 + b_2, \quad (8)$$

where W_1, W_2, b_1, b_2 are kernel and bias parameters.

2.8.2. TunedM2M

In terms of the main target task of predicting aqueous solubility expressed as $\log S$, we opted to utilize the learned knowledge from a pretrained network (M2M) and apply the trained parameters to a new regression task in a process called transfer learning. Therefore, the idea is to transfer the weights calculated for pKa prediction. Here, we make use of the encoder and the embedding layer from M2M. Then, fine-tuning of the parameters is employed to accommodate aqueous solubility prediction. The resulting architecture is called TunedM2M.

2.9. Evaluation details

Root mean square error (RMSE), mean absolute error (MAE), and mean square error (MSE) are used to measure the prediction performance. Additionally, in some cases, to provide more reliable results, Pearson correlation coefficient (PCC) and coefficient of determination (R^2) are given as well. PCC is used to measure the linear association between two variables. In turn, (R^2) is seen as the proportion of the variance in the dependent variable that is predictable from the independent variables. Moreover, throughout this paper, unless otherwise stated, each dataset was randomly split into a training, validation, and test set (80%, 10%, and 10%, respectively).

Table 4
Model hyperparameters.

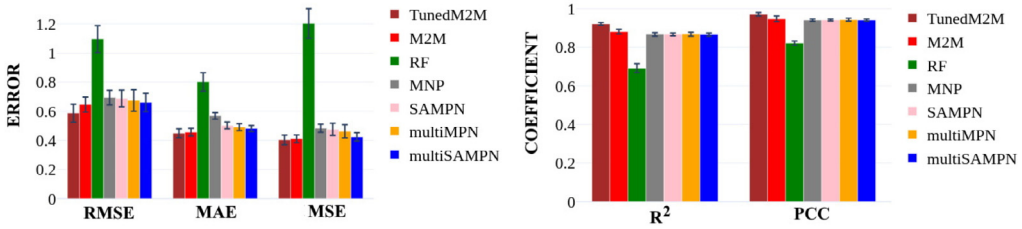
Hyperparameter	Values considered
no. of layers for the transformer	4–8
no. of multi-heads	6–12
dropout rate	0.00.6
initial learning rate	0.00015, 0.0015, 0.015, 0.15
α	0.25
dimension of the final representation	128, 256, 512

3. Results and discussion

To evaluate the performance of our method, we compared our approaches with the state-of-the-art models that have demonstrated superior performance in Tang's work on aqueous solubility prediction task. Specifically, random forests (RF), message passing network (MPN), self-attention-based message-passing neural network (SAMPN), multi message passing network (multiMPN), and multi self-attention-based message-passing neural network (multiSAMPN) are used. RF classifier [48] is one of the most popular and successful classification/regression methods in Euclidean spaces. It employs multiple decision trees to train and predict samples. MPN [49] is a deep neural network-based approach that operates on a graph and separates the prediction process into two phases: message passing phase and readout. SAMPN [31] can be seen as an extended version of MPN with an extra added attention mechanism. Tang et al. also introduce multiMPN and multiSAMPN that are variants of MPN and SAMPN, respectively, where the models learn both the relationship between chemical structures and properties and the relationship between intrinsic attributes of molecules. In addition, since our goal is to obtain a model that performs robustly, the same set of hyperparameters for fine-tuning was determined (see Table 4). We design tasks to answer a few questions.

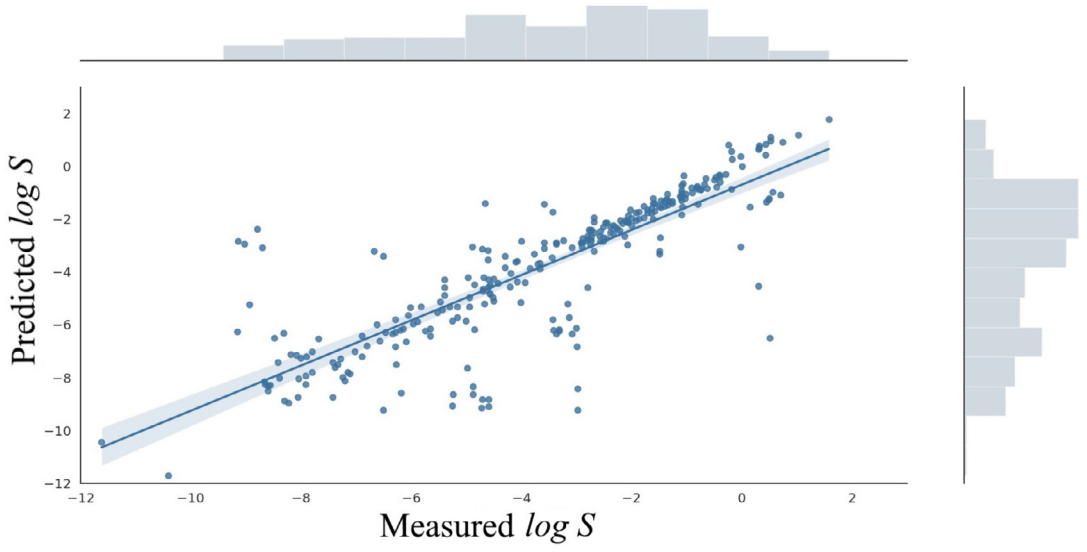
Q1: Do M2M and TunedM2M improve the aqueous solubility prediction results?

One may notice that either M2M or TunedM2M outperformed all competitive methods (see Figs. 6, 7 and Table S1 in the Supplementary Materials). TunedM2M achieves an overall RMSE of 0.587, MAE of 0.449, and MSE of 0.403, which indicates a high degree of accuracy. Crucially, although M2M is trained only on aqueous solubility dataset that comprises a small number of samples, it still achieves a reasonable performance. As shown in Fig. 6a, multiSAMPN achieves an RMSE of 0.661, MAE of 0.482, and MSE of 0.424. Compared to those scores, M2M obtains better prediction performance since it obtains an RMSE of 0.646, MAE of 0.456, and MSE of 0.411. Furthermore, the experiments indicate that selection of evaluation measure doesn't matter here since we arrive at the same conclusion. As summarized in Figs. 6b and 7, M2M leads to an overall $PCC = 0.94$ and $R^2 = 0.88$, while TunedM2M achieves $PCC = 0.97$ and $R^2 = 0.92$. All in all, one may observe that the M2M architecture has a positive impact on prediction performance since M2M is trained only on the aqueous solubility dataset and achieves the second-best performance among all models. However, these results also clearly indicate that the pretraining phase of M2M as performed in TunedM2M improves performance even more. Undoubtedly, TunedM2M produces fairly bad results for only a few molecules (see Fig. 7b). These molecules are diverse in structure, i.e., that they cannot be assigned to one or a few chemical classes. In contrast, the multiSAMPN produces many more bad predictions than our model (see Fig. 7a). More specifically, it produces too low $\log S$ values for a range of different compounds. Also, it produces too high $\log S$ values for some polyhalogenated compounds and some polycyclic aromatic hydrocarbon. Additionally, note that we evaluate statistical significance using a one-sided Wilcoxon signed-rank test and define the statistical significance as p-value less than 0.05. All in all, both M2M trained only on aqueous solubility dataset and TunedM2M significantly outperform the other models.

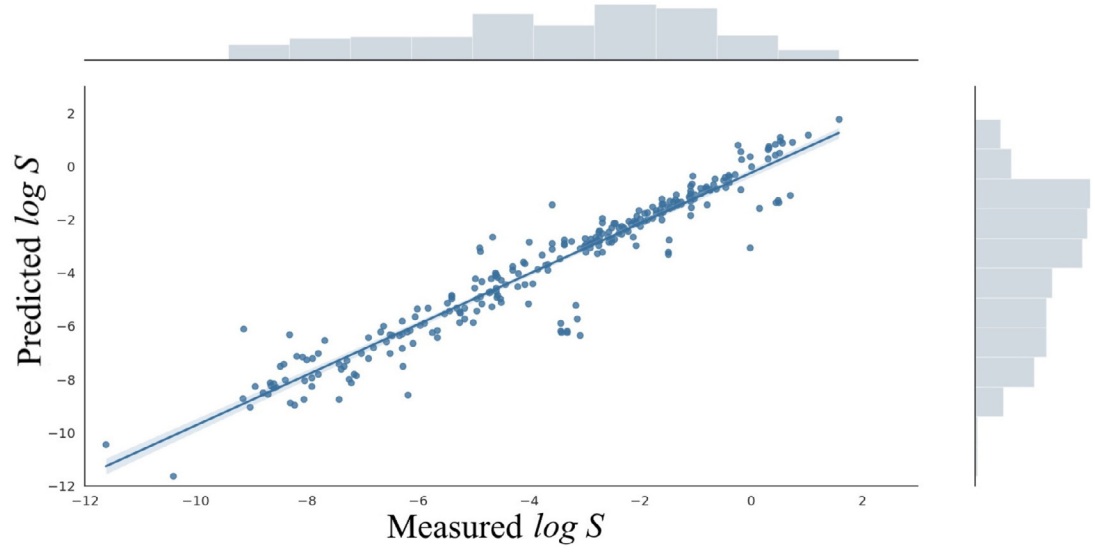


(a) Results for RMSE, MAE and MSE (b) Results for R^2 and PCC

Fig. 6. The scores of various methods on regression task and test set. We achieved the best results.



(a) multiSAMPN



(b) TunedM2M

Fig. 7. Scatter plot for measured and computational model output values of aqueous solubility in two different approaches: (a) multiSAMPN, (b) TunedM2M.

Q2: Is there any noticeable difference between M2M when learned only on the aqueous solubility dataset, and TunedM2M?

The computational tasks run to answer Question no. 1 suggest that TunedM2M works better than M2M trained on the aqueous solubility dataset. Nevertheless, to answer Q2, we conducted additional analyses: TunedM2M was again compared to M2M that is trained from scratch only on the aqueous solubility dataset. However, this time, both M2M and TunedM2M were trained on different numbers of training data and tested on the test set with the ratio of train:valid:test at 8:1:1 and random split. As Fig. 8 shows, with different numbers of train-

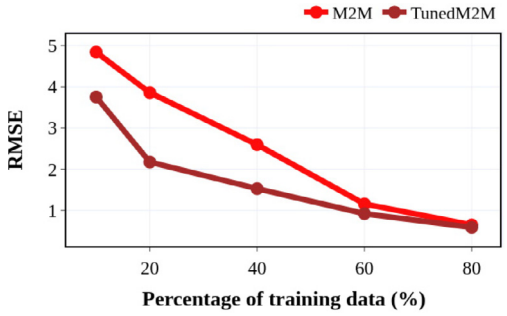


Fig. 8. Performance of M2M and TunedM2M on the different size of the training set.

Table 5
Evaluations of prediction for 32 molecules from the 2019 Solubility Challenge [6].

Method	RMSE
RF	**1.62**
MPN	3.14
SAMPN	3.35
multiMPN	3.24
multiSAMPN	3.48
TunedM2M	3.19

ing data, our TunedM2M outperformed in all cases our M2M trained from scratch on aqueous solubility dataset. This suggests that the transfer learning strategy that is applied in the TunedM2M architecture provides a robust improvement of the model. In other words, the outcomes mean that the training of pKa data benefits the prediction of aqueous solubility.

Q3: What is the chemical explainability of TunedM2M?

We propose a set of tasks to check how each atom contributes to the predictive performance gain. Fig. 9 shows the impact of atoms at the example of three selected chemical compounds. In general, TunedM2M

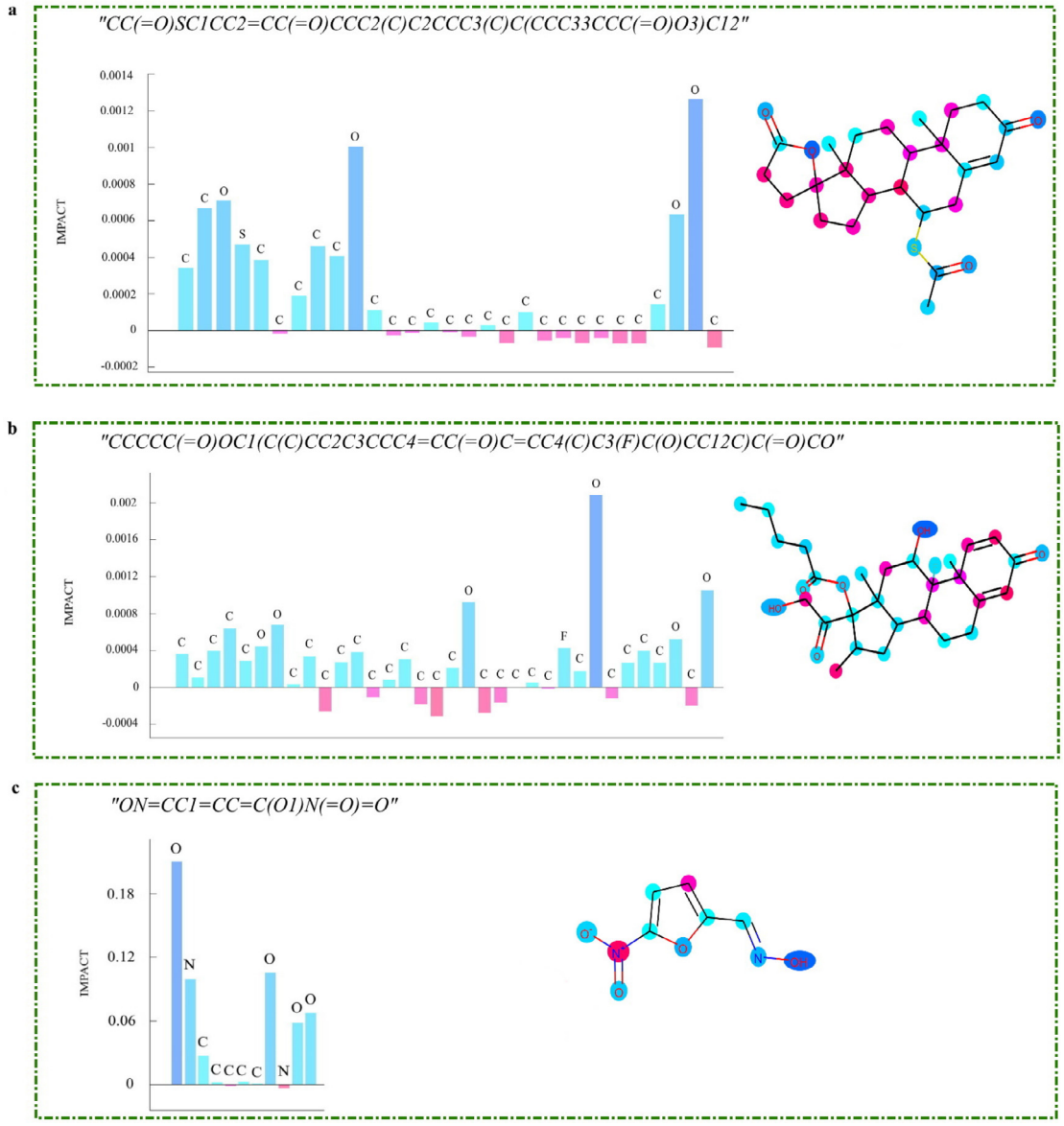


Fig. 9. Visualization of atoms and their calculated impact on aqueous solubility. Please note that TunedM2M accurately assigns negative scores to the hydrophobic atoms and positive importance scores to the atoms that help water dissolve the substance.

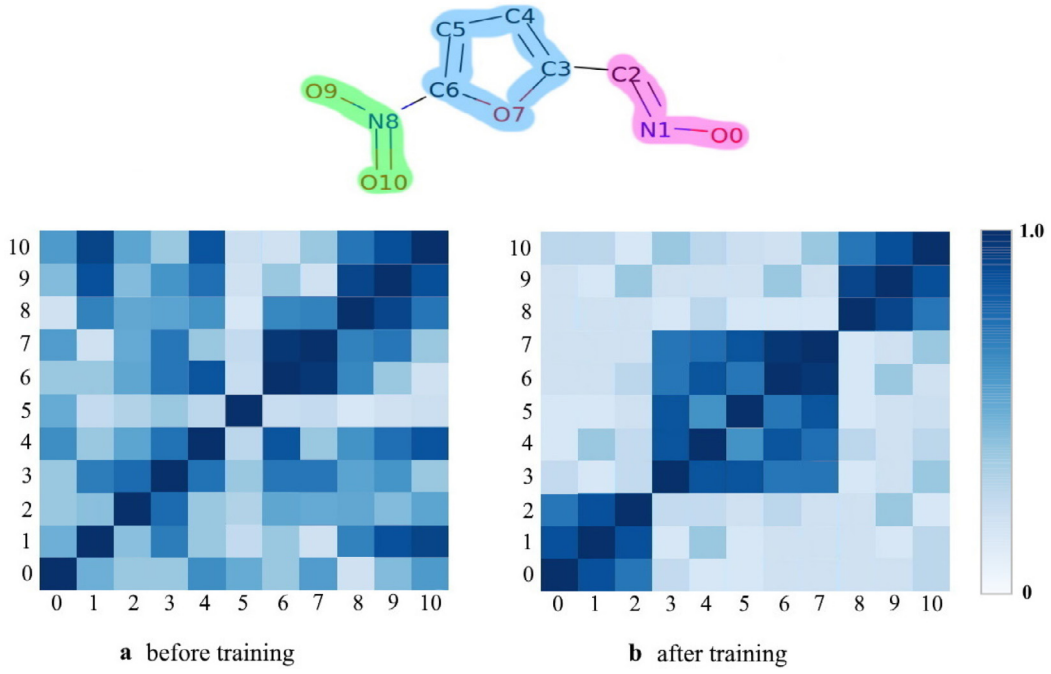


Fig. 10. Heat maps of the atom similarity matrix for the arbitrary selected molecule.

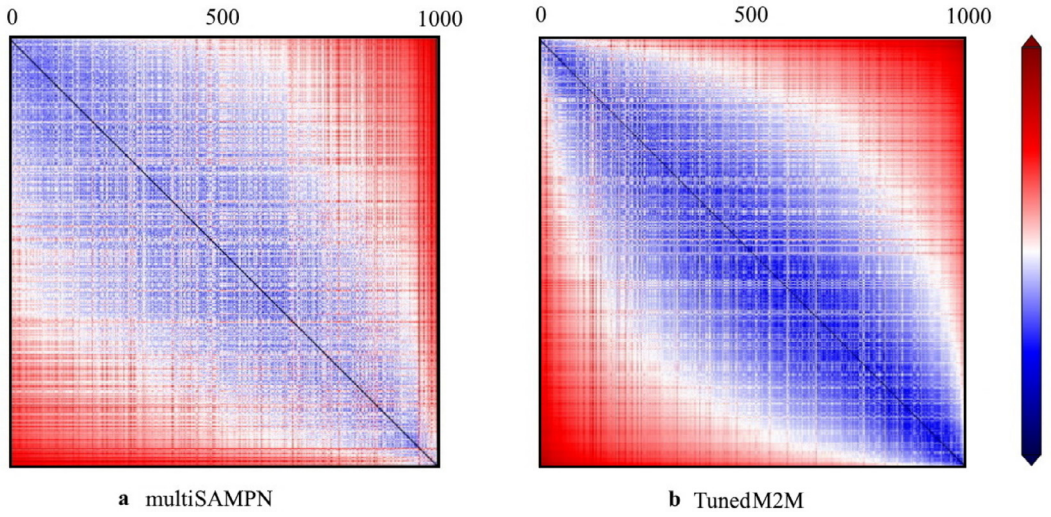


Fig. 11. 2D plot of the $L2$-norm distances between chemical compounds in the latent space associated with $\log S$. The scale bar on the right side is associated with the distance between two molecules labeled by column and row.

accurately assigned positive importance scores to atoms that contribute to aqueous solubility and negative scores to atoms that are hydrophobic. Besides, it can be observed that the atoms that have positive importance scores form a groups that can be associated with some functional groups that are hydrophilic, such as amines, ketones, aldehydes or ethers.

Q4: What is the interpretability of TunedM2M?

In order to get more insight into TunedM2M and to gain a better understanding of its performance, we selected an exemplary molecule presented in Fig. 9c and analyzed how the atom state vectors change during the learning process. Fig. 10 shows heat maps of the atom similarity matrix for the selected chemical compound. Indeed, one may notice that the pattern after training is different from the pattern observed before training. Fig. 10a demonstrates that the heat map of the similarity matrix reveals similar levels of randomness across different layers. In contrast, after training, we obtain three clusters of atoms that are associated with functional groups: an oxime group (atoms no. 0, 1, 2), an arene group (atoms no. 3, 4, 5, 6, 7) and a nitro group (atoms no. 8, 9, 10). In fact, this observation is strongly correlated with the chemical interpretation of the given structure. Therefore, one may infer that TunedM2M is able to learn a representation related to molecular aqueous solubility.

Q5: Do the models capture meaningful patterns in the distance mapping?

We also perform our analysis in latent space. We randomly chose 1000 molecules from the dataset and calculated the $L2$-norm distance between them. The normalized distance mapping is shown in Fig. 11. The scale bar on the right side refers to the distance between two molecules labeled by row and column. Furthermore, the diagonal elements colored in blue show the distance of a molecule from itself. In turn, the red parts mean the farthest distance. It seems that both multiSAMPN and TunedM2M reveal meaningful patterns in the distance mapping. It clearly indicates that for similar $\log S$ values the models are able to locate similar molecules in the latent space. Neverthe-

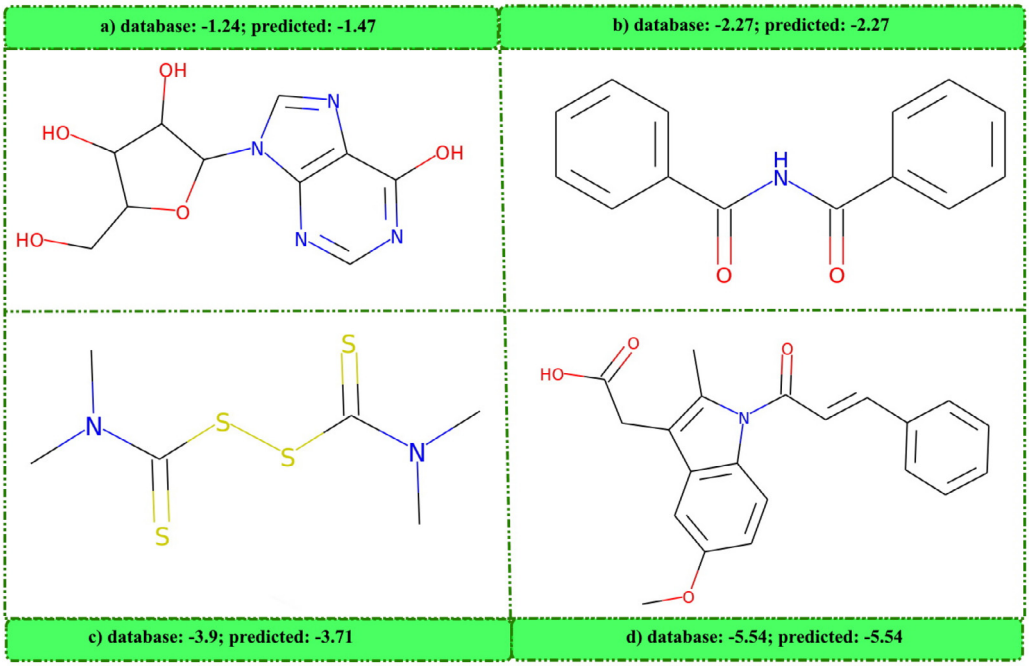


Fig. 12. Selection of chemical compounds and predicted values.

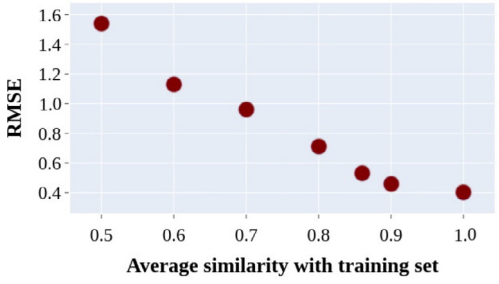


Fig. 13. Prediction error (measured as RMSE) versus the average similarity of the molecule to the rest of the training set.

less, TunedM2M puts chemical compounds that have similar properties closer to each other in space, and that explains the performance of TunedM2M (Figs. 12 and 13).

Q6: Does the usage of (sub)encoders affect the final performance of M2M?

To investigate the contribution of different factors that influence the performance of M2M, we conducted ablation studies. As shown in Fig. 14, decoupling knowledge-aware components such as a graph (sub)encoder and an embedding block in M2M encoder from our learning framework yields a significant drop of performance. Moreover, it seems that a graph (sub)encoder is crucial and effective for better results. In addition, the lack of transformer causes the worst performance.

Q7: Does scaffold-based splitting of the data affect the performance of M2M and TunedM2M?

We also evaluate the models on scaffold-based split. Scaffold splitting is performed using Bemis-Murcko [50] methodology that divides the dataset into a training set, a validation set, and a test set according to their two-dimensional molecular structures. As the Fig. 15 shows, both M2M and TunedM2M beat all the competitive approaches. To be more precise, TunedM2M reaches an RMSE of 0.672, MAE of 0.501, and MSE of 0.438. The second-best approach is M2M trained only on aqueous solubility dataset (RMSE = 0.721, MAE of 0.564 MSE of 0.471), while the third-best is multiSAMPN that obtains an RMSE of 0.735, MAE of 0.571, and MSE of 0.48. In fact, the results indicate that scaffold splitting is a more challenging setting since it yields less impressive scores than the random split approach. Nevertheless, the outcomes indicate that even when the training and the test set have different characteristics related to the molecular structures, the TunedM2M is able to learn the general representation of the molecules and returns a competitive prediction.

Fig. 14. Ablation studies. After removing a graph (sub)encoder, an embedding block and a transformer in M2M encoder from our learning framework, a significant drop of performance is observed.

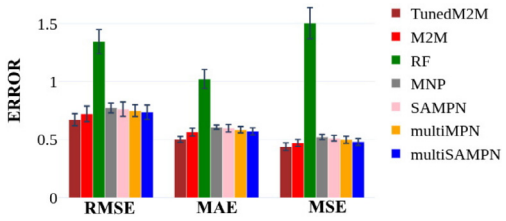


Fig. 15. Scaffold split. The scores of various methods on regression task and test set. We achieved the best results.

Q8: How does the model work for a dataset of intrinsic water solubility?

As part of an additional task, we apply the proposed model to the molecules provided by the organisers of the 2019 Solubility Challenge [6]. It is somewhat not surprising that RF obtains the best results, i.e., RMSE of 1.62 (see Table 5). It shows that for an extremely small dataset, including 100 molecules in the training set, deep learning-based approaches reveal very high error on a test set (32 molecules). Although TunedM2M's performance is not satisfactory, it achieves the third best result (RMSE of 3.19). We hope this observation will help to explore further the challenges connected with prediction of the intrinsic solubilities of molecules for small datasets.

4. Conclusions

To increase the accuracy of aqueous solubility prediction and overcome the difficulties of training deep neural networks caused by limited reliable aqueous solubility-related datasets, in this work, we introduced a transformer-based architecture. Moreover, we employed a transfer learning strategy and called the final architecture as TunedM2M. This paper presents the workflow and performance of our method. In fact, thanks to the well-designed encoder, TunedM2M provides insights on which fragments of the chemical structure have the greatest influence on the property of interest. Furthermore, several proposed tasks have shown our approach is competitive with the state-of-the-art models. The results also demonstrated that the accuracy of the aqueous solubility prediction was significantly improved. Therefore, the outcomes reveals that transfer learning is an effective way to solve the data-hungry problem and may be beneficial for improving the regression performance of other models.

4.1. Data and software availability

Our model was implemented using Python 3.7 and Pytorch 1.2.0. The files used in this study are available on Github (https://github.com/magdalenawi/TunedM2M).

4.1.1. Author contributions

M.W. contributed the concept, implementation and wrote the manuscript. M.W and J.K. contributed to the interpretation of results. All authors reviewed and approved the final manuscript.

Funding

This work is a part of broader research funded under the Iwanowska Program of the Polish National Agency for Academic Exchange (NAWA) - a decision no. PPN/IWA/2018/1/00094/DEC/1. This research was also supported by the National Science Centre (Poland) Grants No. 2016/21/N/ST6/01019. This publication has been partially funded from the SciMat Priority Research Area budget under the program "Excellence Initative - Research University" at the Jagiellonian University.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ailsci.2021.100021.

References

[1] Zhang L, Tan J, Han D, Zhu H. From machine learning to deep learning: progress in machine intelligence for rational drug discovery. Drug Discov Today 2017;22(11):1680–5.
[2] Hann MM. Molecular obesity, potency and other addictions in drug discovery. Medchemcomm 2011;2(5):349–55.
[3] Boobier S, Osbourn A, Mitchell JB. Can human experts predict solubility better than computers? J Cheminform 2017;9(1):1–14.
[4] Wang J, Hou T. Recent advances on aqueous solubility prediction. Comb Chem High Throughput Screening 2011;14(5):328–38.
[5] Palmer DS, Mitchell JB. Is experimental data quality the limiting factor in predicting the aqueous solubility of druglike molecules? Mol Pharm 2014;11(8):2962–72.
[6] Llinas A, Avdeef A. Solubility challenge revisited after ten years, with multilab shake-flask data, using tight (SD 0.17 log) and loose (SD 0.62 log) test sets. J Chem Inf Model 2019;59(6):3036–40.
[7] Dearden JC. In silico prediction of aqueous solubility. Expert Opin Drug Discov 2006;1(1):31–52.
[8] Jiménez-Luna J, Grisoni F, Schneider G. Drug discovery with explainable artificial intelligence. Nat Mach Intell 2020;2(10):573–84.
[9] Fühner H. Die wasserloslichkeit in homologen reihen. Berichte der deutschen chemischen Gesellschaft (A and B Series) 1924;57(3):510–15.
[10] Hansch C, Quinlan J, Lawrence G. The Linear Free-Energy Relationship between Partition Coefficients and the Aqueous Solubility of Organic Liquids. J Org Chem 1924.
[11] Kamlet MJ, Doherty RM, Abboud J-LM, Abraham MH, Taft RW. Linear solvation energy relationships: 36. Molecular properties governing solubilities of organic nonelectrolytes in water. J Pharm Sci 1986;75(4):338–49.
[12] Yalkowsky SH, Valvani SC. Solubility and partitioning i: solubility of nonelectrolytes in water. J Pharm Sci 1980;69(8):912–22.
[13] Erickson L.. The solubility of homologous series of organic compounds. 1952.
[14] Hewitt M, Cronin MT, Enoch SJ, Madden JC, Roberts DW, Dearden JC. In silico prediction of aqueous solubility: the solubility challenge. J Chem Inf Model 2009;49(11):2572–87.
[15] Palmer DS, O'Boyle NM, Glen RC, Mitchell JB. Random forest models to predict aqueous solubility. J Chem Inf Model 2007;47(1):150–8.
[16] Lind P, Maltseva T. Support vector machines for the estimation of aqueous solubility. J Chem Inf Comput Sci 2003;43(6):1855–9.
[17] Könczöl Á, Dargó G. Brief overview of solubility methods: recent trends in equilibrium solubility measurement and predictive models. Drug Discov Today 2018;27:3–10.
[18] Erić S, Kalinić M, Popović A, Zloh M, Kuzmanovski I. Prediction of aqueous solubility of drug-like molecules using a novel algorithm for automatic adjustment of relative importance of descriptors implemented in counter-propagation artificial neural networks. Int J Pharm 2012;437(1–2):232–41.
[19] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Commun ACM 2017;60(6):84–90.
[20] Chen L-C, Zhu Y, Papandreou G, Schroff F, Adam H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV); 2018. p. 801–18.
[21] Yang W, Zhang X, Tian Y, Wang W, Xue J-H, Liao Q. Deep learning for single image super-resolution: abrief review. IEEE Trans Multimedia 2019;21(12):3106–21.
[22] Duvenaud DK, Maclaurin D, Iparraguirre J, Bombarell R, Hirzel T, Aspuru-Guzik A, et al. Convolutional networks on graphs for learning molecular fingerprints. Adv Neural Inf Process Syst 2015;28:2224–32.
[23] Kearnes S, McCloskey K, Berndl M, Pande V, Riley P. Molecular graph convolutions: moving beyond fingerprints. J Comput Aided Mol Des 2016;30(8):595–608.
[24] Wójcikowski M, Kukiełka M, Stepniewska-Dziubinska MM, Siedlecki P. Development of a protein–ligand extended connectivity (PLEC) fingerprint and its application for binding affinity predictions. Bioinformatics 2019;35(8):1334–41.
[25] Louis S-Y, Zhao Y, Nasiri A, Wang X, Song Y, Liu F, et al. Graph convolutional neural networks with global attention for improved materials property prediction. PCCP 2020;22(32):18141–8.
[26] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521(7553):436–44.
[27] Weininger D. Smiles, a chemical language and information system. 1. Introduction to methodology and encoding rules. J Chem Inf Comput Sci 1988;28(1):31–6.
[28] Lusci A, Pollastri G, Baldi P. Deep architectures and deep learning in chemoinformatics: the prediction of aqueous solubility for drug-like molecules. J Chem Inf Model 2013;53(7):1563–75.
[29] Wu K, Zhao Z, Wang R, Wei G-W. TopP–S: persistent homology-based multi-task deep neural networks for simultaneous predictions of partition coefficient and aqueous solubility. J Comput Chem 2018;39(20):1444–54.

[30] Liu K, Sun X, Jia L, Ma J, Xing H, Wu J, et al. Chemi-Net: a molecular graph convolutional network for accurate drug property prediction. Int J Mol Sci 2019;20(14):3389.
[31] Tang B, Kramer ST, Fang M, Qiu Y, Wu Z, Xu D. A self-attention based message passing neural network for predicting molecular lipophilicity and aqueous solubility. J Cheminform 2020;12(1):1–9.
[32] Korotcov A, Tkachenko V, Russo DP, Ekins S. Comparison of deep learning with multiple machine learning methods and metrics using diverse drug discovery data sets. Mol Pharm 2017;14(12):4462–75.
[33] Ba J, Caruana R. Do deep nets really need to be deep? Adv Neural Inf Process Syst 2014;27:2654–62.
[34] Rajasegaran J, Jayasundara V, Jayasekara S, Jayasekara H, Seneviratne S, Rodrigo R. DeepCaps: going deeper with capsule networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2019. p. 10725–33.
[35] Sushko I, Novotarskyi S, Körner R, Pandey AK, Rupp M, Teetz W, et al. Online chemical modeling environment (OCHEM): web platform for data storage, model development and publishing of chemical information. J Comput Aided Mol Des 2011;25(6):533–54.
[36] Sander T, Freyss J, von Korff M, Rufener C. DataWarrior: an open-source program for chemistry aware data visualization and analysis. J Chem Inf Model 2015;55(2):460–73.
[37] Yang C, Tarkhov A, Marusczyk J, Bienfait B, Gasteiger J, Kleinoeder T, et al. New publicly available chemical query language, CSRML, to support chemotype representations for application to data mining and modeling. J Chem Inf Model 2015;55(3):510–28.
[38] Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G. The graph neural network model. IEEE Trans Neural Netw 2008;20(1):61–80.
[39] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. 2014; arXiv preprint arXiv:14090473.
[40] Landrum G, et al. Rdkit: open-source cheminformatics; 2006.
[41] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. 2017; arXiv preprint arXiv:170603762.
[42] Devlin J, Chang M-W, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. 2018; arXiv preprint arXiv:181004805.
[43] Dai Z, Yang Z, Yang Y, Cohen W, Carbonell J, Le Q, et al. Attentive language models beyond a fixed-length context. 2019; arXiv preprint arXiv:190102860.
[44] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. 2013; arXiv preprint arXiv:13013781.
[45] Asgari E, Mofrad MR. Continuous distributed representation of biological sequences for deep proteomics and genomics. PLoS ONE 2015;10(11):e0141287.
[46] Zhang Y-F, Wang X, Kaushik AC, Chu Y, Shan X, Zhao M-Z, et al. SPVec: a Word2vec-inspired feature representation method for drug-target interaction prediction. Front Chem 2020;7:895.
[47] Wang M, Yu L, Zheng D, Gan Q, Gai Y, Ye Z, et al. Deep graph library: towards efficient and scalable deep learning on graphs; 2019.
[48] Ho TK. Random decision forests. In: Proceedings of 3rd international conference on document analysis and recognition, vol. 1. IEEE; 1995. p. 278–82.
[49] Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. Neural message passing for quantum chemistry. In: International conference on machine learning. PMLR; 2017. p. 1263–72.
[50] Bemis GW, Murcko MA. The properties of known drugs. 1. Molecular frameworks. J Med Chem 1996;39(15):2887–93.