JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

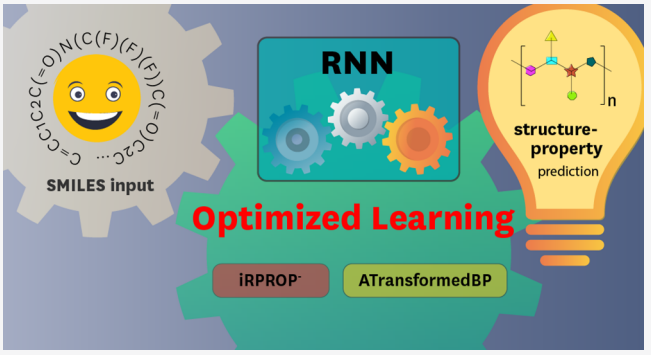
Dielectric Polymer Property Prediction Using Recurrent Neural Networks with Optimizations

Antonina L. Nazarova, Liqiu Yang, Kuang Liu, Ankit Mishra, Rajiv K. Kalia, Ken-ichi Nomura, Aiichiro Nakano,* Priya Vashishta, and Pankaj Rajak

Cite This: *J. Chem. Inf. Model.* 2021, 61, 2175–2186 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Despite the growing success of machine learning for predicting structure–property relationships in molecules and materials, such as predicting the dielectric properties of polymers, it is still in its infancy. We report on the effectiveness of solving structure–property relationships for a computer-generated database of dielectric polymers using recurrent neural network (RNN) models. The implementation of a series of optimization strategies was crucial to achieving high learning speeds and sufficient accuracy: (1) binary and nonbinary representations of SMILES (Simplified Molecular Input Line System) fingerprints and (2) backpropagation with affine transformation of the input sequence (ATransformedBP) and resilient backpropagation with initial weight update parameter optimizations (iRPROP$^-$ optimized). For the investigated database of polymers, the binary SMILES representation was found to be superior to the decimal representation with respect to the training and prediction performance. All developed and optimized Elman-type RNN algorithms outperformed nonoptimized RNN models in the efficient prediction of nonlinear structure–activity relationships. The average relative standard deviation (RSD) remained well below 5%, and the maximum RSD did not exceed 30%. Moreover, we provide a C++ codebase as a testbed for a new generation of open programming languages that target increasingly diverse computer architectures.



INTRODUCTION

In recent years, studies of structure–activity or structure–property relationships have been a focus of life science,[1−4] chemistry,[5] and materials research.[6,7] The computer-aided molecular design represents a milestone in industrial evolution and forces a shift from the time- and resource-intensive combinatorial search to *in silico* high-throughput screening sometimes in a matter of seconds.[8] New methods for digitalizing input data allowed the further design of computational tools to boost an evolution in the development of new materials. The fourth industrial revolution accelerated the digital transformation in material science, which is operating more and more with computationally obtained rather than experimental data.[9,10] Computational methods enabled the development of robust and reliable models that can predict the properties of materials to directly benefit experimental synthesis.[11,12]

Compared to the vast success of machine learning (ML) for predicting structure–property relationships in molecular and material research,[7] predicting the dielectric properties of polymers (i.e., dielectric polymer genome) is still in its infancy.[13−15] Recently, computationally synthesized polymer databases have been created,[16] which involved highly accurate computation of dielectric constants based on the new generation of first-principles-informed polarizable reactive force field methods.[17−19] These new computational databases provide an ideal testing ground for the quantitative assessment of ML models for the dielectric polymer genome within a well-controllable environment.

While building training models to predict the physical properties of new materials, various types of molecular fingerprints need to be considered. Descriptor functions can range from just a single atom to a net of algorithms expressing electron charge distribution, considering functional groups independently as well as other quantum mechanical features. Additional input parameters can be introduced, which however increase computational costs. Proposed by Wiswesser in 1952, the SMILES (Simplified Molecular Input Line System) representation remains one of the most commonly used and low-space-complexity descriptors.[20] SMILES is a linear string input representation of molecular fingerprints for a specific

Received: November 24, 2020
Published: April 19, 2021

https://doi.org/10.1021/acs.jcim.0c01366
J. Chem. Inf. Model. 2021, 61, 2175–2186

structure. While being also a vector-type, the SMILES input representation can be transformed into a digital sequence notation, such as binary, decimal, or ASCII symbols.[21,22] Different SMILES transformers have been reported, which allow a high level of contextualized text-based symbol extraction and encoding of chemical structures.[23] Utilizing SMILES for such molecular fingerprints became crucial in the development of suitable mathematical methods to model structure−property or structure−activity relationships.

Today, ML is considered the most versatile and convenient approach for the construction of such relationship models. A variety of ML methods have been used to predict the properties of compounds to avoid the need for complex molecular characterization.[24] Artificial neural networks of various architectures are a prominent platform for property prediction as well as materials design. In multiple reports as well as in our studies, it was shown that multilayer perceptron (MLP),[25] convolutional (CNN),[26] and graph (GNN)[27] neural networks perform well in mapping the physicochemical properties of complex polymeric systems. While deep neural networks (DNN) generally exhibit advanced performance compared to more classic ML models, a careful choice of the input representation, as well as optimization of model parameters, have shown the recurrent neural network (RNN) to be an equally effective alternative for polymer genome studies.[28]

In this work, we used the single-layer Elman RNN to identify correlations between the structure of polymers of the norbornene class and their permittivity while using the SMILES notation in binary and decimal representations. We compare two algorithms to implement RNN, the original backpropagation (BP) and its modification (ATransformedBP), developed herein with affine transformed input as well as resilient propagation (RPROP) with an optimized parameter of the initial weight update. The algorithms were compared in terms of their learning effectiveness when predicting dielectric parameters of complex polymeric structures.

In addition, to establishing RNN on SMILES input as a convenient but effective approach for the prediction of the structure-permittivity relationship in a well-defined computational dielectric polymer database, we provide a C++ codebase as a natural testbed for the new generation of open programming languages on increasingly diverse computer architectures. The most widely used ML software stack utilizes Python as a user interface with the backend written in the proprietary CUDA (Compute Unified Device Architecture) language[29] that runs exclusively on NVIDIA graphics processing units (GPUs).[30] To enable the universal use of ML on increasingly diverse GPUs from other vendors like Intel and AMD, there is a strong need for developing alternative ML software stacks based on open programming languages, most notably OpenCL,[31] which itself is written in C++. Also, the recently introduced data-parallel C++ (DPC++) language provides a unified interface to not only GPUs but also field-programmable gate arrays (FPGAs) and other heterogeneous accelerators.[32] Our customizable approach opens new venues for a thorough and flexible prediction of nonlinear structure−property relationships while effectively processing large databases in the exponentially growing chemical space in rapidly evolving programming environments.

METHODS

This section describes the learning algorithms used in this work and affine transformation method for the SMILES input that can be applied in the context of the NN optimization strategy.

Methods for Input Data Normalization. Choice of the learning algorithm determines the training efficiency of the weight parameters in an artificial neural network (ANN) model. Backpropagation (BP) is the most commonly used learning method. However, BP comes with several disadvantages that negatively affect training efficiency, including frequent trapping in local minima, ANN stagnation, and sluggish convergence.[33,34] A momentum term can be applied to overcome the issue of local minima trapping by proportionally changing the current weights.[35] However, in practice the identification of a suitable momentum term is challenging, especially when used in combination with other learning parameters.[36] Alternative methods are batch-size variations,[37] examination of diverse activation functions,[38] optimization of learning rates,[39] and initial weight parameters,[40] as well as input normalization.[41] We have addressed this challenge in our structure−property relationship model by means of a SMILES input affine transformation (AT).[42] The effectiveness of AT for input sequence scaling in ML has been demonstrated recently for multilayer perceptron (MLP),[43] convolutional neural networks (CNN),[44,45] and deep recurrent neural network architectures, such as long short-term memory (LSTM)[46] and fast−slow RNN (FS-RNN).[47] Our work identifies the optimal value of the scaling factor of AT, which significantly enhances the accuracy and efficiency of the parent BP algorithm (ATransformedBP). Due to ambiguity in the domain input, we validated AT for SMILES data in both binary and decimal formats.[37] The search for the optimal scaling factor was directly dependent on the initial values of the learning rate and weight parameters in a particular NN model. The working principle of the ATransformedBP algorithm can be described as follows:

Step 1: Random initialization of the weight parameters, followed by an iterative search for the AT factor α for the identification of a minimum in the loss function

Step 2: Performing learning and prognosis studies using the optimized AT factor α in Step 1.

Ready-to-Use vs Self-Developed Programming Frameworks. Riedmiller and Braun proposed modifications of the BP algorithm, i.e., resilient backpropagation (RPROP), to address the problem of convergence and local minima stacking. Comparative studies of the loss function and the duration of the learning process have proven RPROP to outperform its parent BP.[48−51] These results were partly confirmed by our investigations. Specifically, the superiority of the iRPROP$^-$ variant (a modification of RPROP with improved weight backtracking)[49,52] has been confirmed for the prediction of dielectric constants for computationally synthesized amorphous polymers. Also, iRPROP$^-$ and optimized ATransformedBP algorithms were found to perform nearly identically in the training and testing phases, particularly in the case of a binary format SMILES representation.

These somewhat unexpected observations demonstrate the benefit of the C++ codebase that accompanies this work,[53] which allows the in-depth investigation and analysis of such algorithm effectiveness when varying parameters of interest.[54] When using these easy-to-access programming tools, research-

https://doi.org/10.1021/acs.jcim.0c01366
J. Chem. Inf. Model. 2021, 61, 2175−2186

ers are not restricted by the plug-and-play functionality of ready-to-use NN frameworks (e.g., PyTortch[55] and TensorFlow[56]).

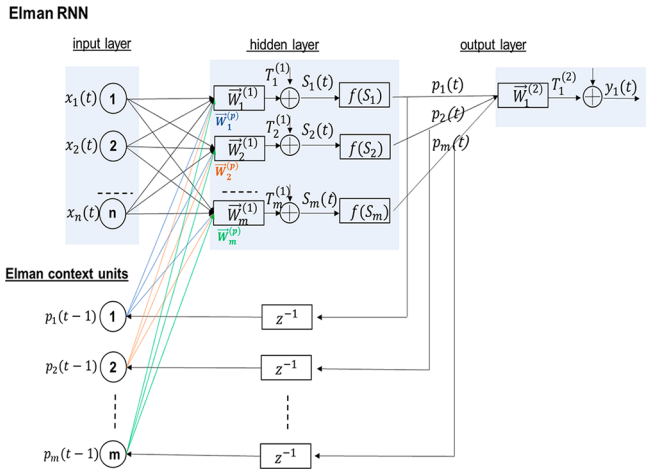
Motivation for Using RNN Algorithm. Data-driven approaches have spanned the field of drug discovery,[57] ligand−protein interaction studies,[58] and new materials design, when predicting properties of potential explosives,[59] superconductors,[60] superalloys,[61] and dielectric polymers.[62,63] Conventional ML techniques have been effective for molecular representation learning, with deep learning (DL) models outperforming in the nascent field of polymer informatics.[15,64] Multiple ML frameworks have been used as a key driver for structure−property studies of polymeric sequences, especially recurrent neural networks (RNN) that have become a popular technique in drug and peptide design due to an ability of training on pattern recognition.[65,66] RNN has considerable potential for prognosis studies, yet is under-represented in the polymer genome development.[15,67] In the current work, the effectiveness of recursive frameworks was found highly dependent on the format of the given input. Indeed, the learning set of the polymeric motifs consisted of several clusters of similar SMILES corresponding to the same range of dielectric constants. While employing a group of same-class polymeric structures possessing at least one repetitive framework moiety (identical SMILES fragment), we expect RNN to be a useful learning strategy. A similar type of related data was shown to be well treated by RNN when solving problems of speech recognition,[68] language translation,[69] and processing of the time-dependent sequential data.[70]

Computational Framework and Data Set Generation. Norbornene polynomial scaffolds have recently gained interest in optical, electronic, and theragnostic applications due to an ability to retain low scattering loss while possessing high glass transition temperatures, adhesion, and low values of dielectric constants.[71,72] The glass transition temperature (T_g) and dielectric constant (ε) are crucial parameters dictating a polymer's physical and chemical properties in dielectric polymer applications. The dielectric constant determines the permittivity characteristics of a polymer and can be changed significantly upon the incorporation of various functional groups. For instance, the introduction of siloxane groups leads to increased mechanical flexibility and gas permeability, while carboxylic acid esters alter the hydrophobicity and thermal stability of polymeric structures.[73,74] Cross-linking with electronically saturated aliphatic scaffolds allows polymers with increased thermal stability and solubility. On the basis of these features, a diverse library of 1200 oligomeric structures with unknown chirality was computationally generated and structurally characterized, using the valence-aware polarizable reactive force field (ReaxPQ-v) method—a recently introduced variant of the polarizable reactive force field (ReaxPQ) method.[17,75−77] Modifications on the polymeric subset were made to ensure covering a broad range of variable functionalities. The building blocks included organic fragments having electron-withdrawing (e.g., CF_3-, Cl-, F-), electron-donating (e.g., OMe-, tBu-, Me-), amide-type group substituents (NH−R−C═O), sterically hindered groups, and unsaturated methylene moieties. Together with the amide group mentioned above, structural motifs also included oxygen, sulfur, and halogen atoms. The training data consisted of the unique SMILES representations (with unspecified chirality) and computed dielectric constants (ε). The pool of SMILES fingerprints of the current set showed identical structural fragments like repetitive fused heterocycles or cycloalkane motifs since they belong to the same polymeric class. Electronically diverse substituents and sterically hindered functionalities were incorporated to capture all potential features that cause changes in ε.

General Architecture of RNN. In RNN, the output signals of the current layer are synaptically correlated with the outcome of the previous layer, thus being time (or sequence) dependent. Thus, the transformation of time-dependent inputs of the current node into an output that is later fed to the next node is defined as a time step. RNNs can be classified based on the connectivity between hidden, context, and output layers. In the Elman type of RNN, the output signal from the hidden layer is united with the neurons of the context layer (Scheme 1). The summator S_j (t) of the hidden layer is determined as

$$S_j(t) = \vec{W}_j^{(1)}\vec{x}^T(t) + \vec{W}_j^{(p)}\vec{p}^T(t-1) + T_j^{(1)},\ 1 \le j \le m \tag{1}$$

Scheme 1. Single-Layer Elman RNN Architecture[a]



[a] z^{-1} is a memory registry defining the context unit application for a one-time step delay.

where $\vec{x}^T$ (t) is input column vector with dimension n, $\vec{p}^T$ ($t-1$) is a column vector with dimension m from the activation function of a hidden layer, $\vec{W}_j^{(1)}$ is a hidden-layer row vector with dimension n, $\vec{W}_j^{(p)}$ is a context unit row vector with dimension m, and $T_j^{(1)}$ is a bias for the neuron j of a hidden layer. In general, $m \neq n$.

The output $p_j(t)$ from the activation function $f(x)$ is determined as

$$p_j(t) = f(S_j(t)),\ 1 \le j \le m \tag{2}$$

The RNN output is expressed as

$$y(t) = \vec{W}_1^{(2)}\vec{p}^T(t) + T_1^{(2)} \tag{3}$$

where $\vec{p}^T$ (t) is a column vector of m dimension with elements $p_j(t)$, $\vec{W}_1^{(2)}$ is a row vector with dimension m for the output layer, and $T_1^{(2)}$ is a bias for the output layer.

In this work, we developed and tested one layer Elman-type RNN architectures using binary and decimal representation of the generated SMILES input $\vec{x}(t)$ (Scheme 1).

Data Set Preparation. The effectiveness of the proposed RNN architectures was evaluated by performing studies on two types of polymeric data sets. The prognoses (testing) data set

https://doi.org/10.1021/acs.jcim.0c01366
J. Chem. Inf. Model. 2021, 61, 2175−2186

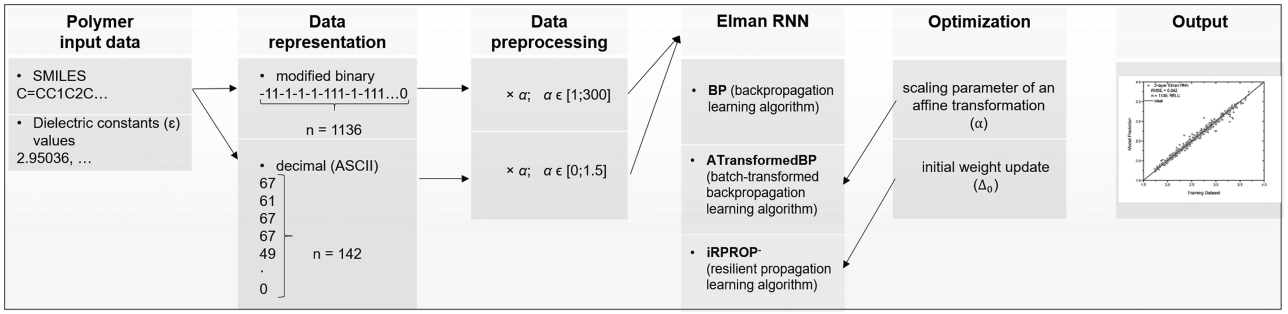


Figure 1. Flowchart of our RNN-based polymeric dielectric constant prediction. Batch-trained affine transformed BP (ATransformedBP) and epoch-trained resilient propagation (iRPROP$^-$) learning algorithms form the basis of the RNN developed in this work.

was composed of 5% of the network answers by taking every 20th sample from the original set of 1200 polymers. The learning (training) data set is composed of the remaining 1140 polymers. We applied both binary and decimal transformations on the SMILES data in the string input format. For the creation of fingerprints via the "modified binary" type representation, each SMILE notation was encoded as the sequence of 1 and −1 (zeroes were replaced with −1). The longest binary representation was used to define the size of the entire training set (fixed size), where zero padding was applied to shorter sequences. After the zero-padding procedure, the binary representation was transformed into a matrix with 1200 rows (number of samples) and 1136 columns (number of input neurons). For the decimal numerical representation of the SMILES input, a conversion of the string variables according to the ASCII decimal code was performed. In this case, the maximum length of the SMILES representation upon zero padding was 142 (eight binary digits were combined for corresponding decimal value). Figure 1 shows the SMILES representation of the polynomial training set before and after the transformation.

Activation Functions for RNNs. As activation functions $f(S)$ for RNNs, we considered hyperbolic tangent (Tanh), Sigmoid, and rectified linear unit functions (RELU)

$$f(S) = \mathrm{Tanh}(S) \tag{4}$$

$$f(S) = 1/(1 + e^{-S}) \tag{5}$$

$$f(S) = \begin{cases} 0, & S \le 0 \\ S, & S > 0 \end{cases} \tag{6}$$

where S is the output for each node of a hidden layer.

Normalized Backpropagation and Resilient Backpropagation Learning Algorithms. One of the most efficient learning algorithms is considered backpropagation, which is based on the iterative gradient refinement of the weighting parameters $\omega^{(t+1)}$ for the $(t + 1)$th tact in row-vectors $\vec{W}_j^{(2)}$ and $\vec{W}_j^{(p)}$ and for biases T in $T_j^{(1)}$ and $T_1^{(2)}$[78]

$$\omega^{(t+1)} = \omega^{(t)} - \gamma \cdot \frac{\partial L(y(\vec{x}_i), \varepsilon(\vec{x}_i))}{\partial \omega} \tag{7}$$

$$T^{(t+1)} = T^{(t)} - \gamma \cdot \frac{\partial L(y(\vec{x}_i), \varepsilon(\vec{x}_i))}{\partial T} \tag{8}$$

where γ is the convergence parameter of learning (the recommended value is 0.01). In eqs 7 and 8

$$L(y(\vec{x}_i), \varepsilon(\vec{x}_i)) = (y(\vec{x}_i) - \varepsilon(\vec{x}_i))^2 \ (i = 1, 2, \cdots, N) \tag{9}$$

is the objective function that is equivalent to the Euclidean metric, where N is the length of the training set pool, ε is the ground-truth dielectric constant, and y is predicted. The value of i is assigned randomly when computing eqs 7 and 8.

The following equations are applicable for the single-layer Elman RNN model using the standard backpropagation approach[79]

$$\frac{\partial L(y(\vec{x}_i), \varepsilon(\vec{x}_i))}{\partial \omega_j^{(2)}} = (y(\vec{x}_i) - \varepsilon(\vec{x}_i))p_j \tag{10}$$

$$\frac{\partial L(y(\vec{x}_i), \varepsilon(\vec{x}_i))}{\partial \omega_{jl}^{(1)}} = (y(\vec{x}_i) - \varepsilon(\vec{x}_i))\omega_j^{(2)} f'(S_j) x_l \tag{11}$$

$$\frac{\partial L(y(\vec{x}_i), \varepsilon(\vec{x}_i))}{\partial T_j^{(1)}} = (y(\vec{x}_i) - \varepsilon(\vec{x}_i))\omega_j^{(2)} f'(S_j) \tag{12}$$

$$\frac{\partial L(y(\vec{x}_i), \varepsilon(\vec{x}_i))}{\partial \omega_j^{(p)}} = (y(\vec{x}_i) - \varepsilon(\vec{x}_i))\omega_j^{(2)} f'(S_j) p_j \tag{13}$$

$$\frac{\partial L(y(\vec{x}_i), \varepsilon(\vec{x}_i))}{\partial T_1^{(2)}} = (y(\vec{x}_i) - \varepsilon(\vec{x}_i)) \tag{14}$$

Here, $1 \le j \le m$, $1 \le l \le n$ (Scheme 1).

In case of backpropagated recurrent networks, weights are usually updated by sample batch. Weighting parameters and biases are initialized randomly in the range $[-1/\sqrt{n}, +1/\sqrt{n}]$.

The disadvantage of the BP learning algorithm is the time-consuming calculation of the iterative system of eqs 10−14 and potentially only reaching a local root-mean-square error (RMSE) minimum of the $\langle L(y(\vec{x}_i), \varepsilon(\vec{x}_i))\rangle$ function, where $\langle\ \rangle$ represents averaging over the entire training set. This demands the pursuit of a more efficient and improved solution. For this purpose, we used the normalized backpropagation approach (ATransformedBP) with input size rationing, which significantly increases the convergence rate of the learning, while avoiding local minima. The developed ATransformedBP approach is based on the preprocessing of the SMILES input $\vec{x}_i$ (in either binary or decimal representation) according to the rule

$$\vec{x}_i' = \alpha \cdot (\vec{x}_i - \vec{x}_i), \ 1 \le i \le N \tag{15}$$

where α is the AT factor.

The optimal value of α is determined to minimize the objective function

https://doi.org/10.1021/acs.jcim.0c01366
J. Chem. Inf. Model. 2021, 61, 2175−2186

Table 1. RNN Models Internal Parameters

RNN internal parameters	ATransformedBP	iRPROP$^-$
Format of the SMILES input Training set pool	binary, decimal 1140	binary, decimal 1140
Testing set pool	60	60
Input nodes	142, 1136	142, 1136
Hidden nodes	142, 1136	142, 1136
Hidden layers	1	1
Context units layer	1	1
Activation function	Tanh, Sigmoid, and RELU	Tanh, Sigmoid, and RELU
Weight updates	by mini-batch	by epoch
Sample selection	random	sequential
Optimized parameters	α	Δ_0

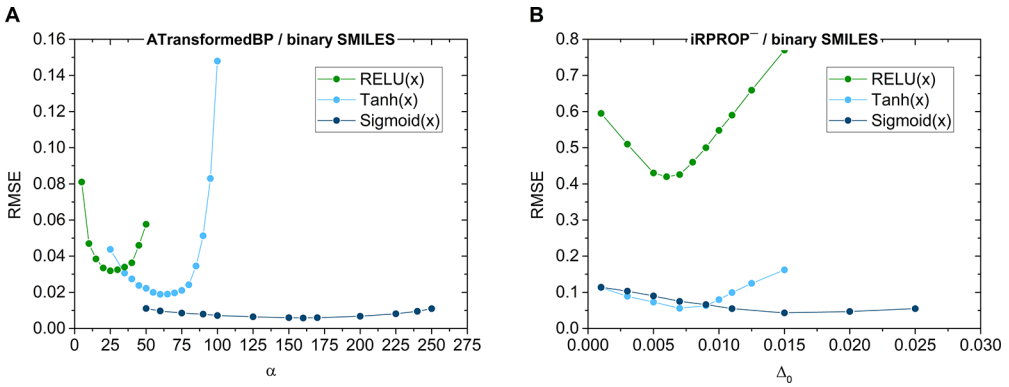


Figure 2. RMSE plots for the variation of (A) affine transform factor α in ATransformedBP learning algorithm and (B) initial weight update Δ_0 in iRPROP$^-$ learning algorithm.

$$\alpha_{\text{opt}} = \min_{\alpha}\langle L(y(\vec{x}_i), \varepsilon(\vec{x}_i))\rangle \tag{16}$$

with respect to the initial values of weights and biases. The initial weighting parameters of the hidden layer $\omega_{jl}^{(1)}$ $(1 \le j \le m,\ 1 \le l \le n)$ remain fixed when varying α and can be expressed as

$$\omega_{jl}^{(1)} = \alpha/\sqrt{n}\cdot(0.5 - \text{RAND}()/\text{RANDMAX}) \tag{17}$$

where RAND() is a random-number generator in the range [0, RAND_MAX] in the cstdlib.h library of the C++ programming language. Consequently, the initial values of $\omega_{jl}^{(1)}$ are in the range $\left[-\frac{\alpha}{2\sqrt{n}}, \frac{\alpha}{2\sqrt{n}}\right]$. The same initialization procedure is applied to the remaining weighting parameters of the context units $\omega_{jk}^{(p)}$ $(1 \le k \le m, 1 \le j \le m)$ output layer $\omega_{1j}^{(2)}$ and biases $T_j^{(1)}$ and $T_1^{(2)}$ of the input and output layers, respectively.

The results obtained using ATransformedBP were compared with those using resilient propagation learning algorithms (RPROP).[80] The basic difference of supervised learning approaches with different types of the RPROP (RPROP$^+$, RPROP$^-$, iRPROP$^+$, iRPROP$^-$) is the utilization of the sign of the gradient $\frac{\partial L^{(t)}}{\partial \omega^{(t)}}$ for the tth time step, where $L^{(t)} = \langle L^{(t)}(y(\vec{x}_i),\varepsilon(\vec{x}_i))\rangle$. Thus, the function $L^{(t)}$ is calculated as the mean value for the entire training set per one learning cycle, i.e., an epoch averaged.

In our developed RNN model, the iRPROP$^-$ algorithm was found to be most effective in solving the complex nonlinear problem of polymeric structure−property relationships. The iRPROP$^-$ algorithm is based on the fulfillment of the following "triple-step" conditions for the tth time step:

Step 1 condition: If $\frac{\partial L^{(t-1)}}{\partial \omega^{(t-1)}}\cdot\frac{\partial L^{(t)}}{\partial \omega^{(t)}} > 0$ then

$$\Delta^{(t)} = \min(\Delta^{(t-1)}\Delta\eta^+, \Delta_{\max}),\quad R^{(t)} = \frac{\partial L^{(t)}}{\partial \omega^{(t)}} \tag{18}$$

Step 2 condition: If $\frac{\partial L^{(t-1)}}{\partial \omega^{(t-1)}}\cdot\frac{\partial L^{(t)}}{\partial \omega^{(t)}} < 0$ then

$$\Delta^{(t)} = \max(\Delta^{(t-1)}\Delta\eta^-, \Delta_{\min}),\quad R^{(t)} = \frac{\partial L^{(t)}}{\partial \omega^{(t)}} = 0 \tag{19}$$

Step 3 condition: If $\frac{\partial L^{(t-1)}}{\partial \omega^{(t-1)}}\cdot\frac{\partial L^{(t)}}{\partial \omega^{(t)}} = 0$ then

$$\Delta^{(t)} = \Delta^{(t-1)},\quad R^{(t)} = \frac{\partial L^{(t)}}{\partial \omega^{(t)}} \tag{20}$$

The weights updates are defined through

$$\Delta\omega^{(t)} = -sign(R^{(t)})\cdot\Delta^{(t)}$$

$$\omega^{(t)} = \omega^{(t-1)} + \Delta\omega^{(t)}$$

$$\text{sign}(x) = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0 \end{cases} \tag{21}$$

where $\eta^+ = 1.2$, $\eta^- = 0.5$, $\Delta_{\max} = 50$, and $\Delta_{\min} = 10^{-6}$ are recommended literature values.[80]

$\Delta^{(0)}$ parameters in eqs 18−21 are initialized as $\Delta^{(0)} = \Delta_0$. The optimal value $\Delta_0(opt)$ of the initial weight update should satisfy the following condition

$$\Delta_{0,opt} = \min_{\Delta_0}\langle L(y(\vec{x}_i), \varepsilon(\vec{x}_i))\rangle \tag{22}$$

https://doi.org/10.1021/acs.jcim.0c01366
J. Chem. Inf. Model. 2021, 61, 2175−2186

Table 2. Optimized Parameters α and Δ_0 for ATransformedBP and iRPROP$^-$ Learning Algorithms, Respectively[a]

ATransformedBP								
SMILES format	decimal (n = 142)		binary (n = 1136)					
Activation function	Tanh		Tanh		RELU		Sigmoid	
	α_{opt}	RMSE	α_{opt}	RMSE	α_{opt}	RMSE	α_{opt}	RMSE
Training set pool	1.0	0.104	70.0	0.00073	25.0	0.0187	160.0	0.0031
Testing set pool	1.0	0.201	70.0	0.164	25.0	0.167	160.0	0.154
iRPROP$^-$								
SMILES format	decimal (n = 142)		binary (n = 1136)					
Activation function	Tanh		Tanh				Sigmoid	
	$\Delta_{0,opt}$	RMSE	$\Delta_{0,opt}$	RMSE			$\Delta_{0,opt}$	RMSE
Training set pool	0.002	0.078	0.007	0.00074			0.015	0.0103
Testing set pool	0.002	0.198	0.007	0.165			0.015	0.181

[a]RMSE parameters for training and testing sets.

Table 3. Statistical Analysis of RNN Based on ATransformedBP and iRPROP$^-$ Algorithms[a]

ATransformedBP								
SMILES format	decimal (n = 142)		binary (n = 1136)					
Activation function	Tanh		Tanh		RELU		Sigmoid	
	α_{opt}	up to 10% of RSD	α_{opt}	up to 10% of RSD	α_{opt}	up to 10% of RSD	α_{opt}	up to 10% of RSD
Training set	1.0	96%	70.0	100%	25.0	100%	160.0	100%
Testing set	1.0	80%	70.0	85%	25.0	90%	160.0	91%
iRPROP$^-$								
SMILES format	decimal (n = 142)		binary (n = 1136)					
Activation function	Tanh		Tanh				Sigmoid	
	$\Delta_{0,opt}$	up to 10% of RSD	$\Delta_{0,opt}$	up to 10% of RSD			$\Delta_{0,opt}$	up to 10% of RSD
Training set	0.002	99%	0.007	100%			0.015	100%
Testing set	0.002	85%	0.007	88%			0.015	88%

[a]Relative volumes of training and predicted sampling sets do not exceed the 10% limit of statistical fluctuation.

RESULTS AND DISCUSSION

Software Used for Comparing RNN Models. The single-layer Elman type RNN models were written in C++, and a schematic representation of the implementation is depicted in Scheme 1. The learning efficiency as well as the prediction performance of the developed models were statistically analyzed using the RMSE and the relative standard deviation (RSD) parameters. The averaged (eq 23) and maximum (eq 24) forms of the RSD parameter were defined as

$$\mathrm{RSD(av)} = \frac{\sum_{i=0}^{n} (y(\vec{x_i}) - \varepsilon(\vec{x_i}))}{n} \quad (23)$$

$$\mathrm{RSD(max)} = \max_i (y(\vec{x_i}) - \varepsilon(\vec{x_i})) \quad (24)$$

For the RNN model using the ATransformedBP learning algorithm, a random pattern selection was utilized for training and prediction. Using the iRPROP$^-$ learning protocol, the samples were selected consistently from the learning set to accomplish epoch averaging. Details of the developed RNN models and their parameters are given in Table 1.

Determination of Optimized Parameters: Affine Transformation Factor α and Initial Weight Update Parameter Δ_0. Modeling studies were performed to find the optimal parameters for α (ATransformedBP) and Δ_0 (iRPROP$^-$). The results were obtained by using a learning sample pool (training data set) which was trained for 100 epochs.

Figure 2 shows the dependency of the RMSE on α in case of ATransformedBP learning or the initial weight update Δ_0 in case of the iRPROP$^-$ algorithm. The corresponding optimal values of α or Δ_0 representing a minimal RMSE are given in Table 2. The lowest RMSE value in the Sigmoid activation function was determined to be at α_{opt} = 160 (ATransformedBP) or at $\Delta_{0,opt}$ = 0.015 (iRPROP$^-$).

Similar plots were made for decimal SMILE input format (142 nodes) using the most effective activation function—Tanh (the RELU and Sigmoid activations demonstrated significantly higher values in the loss function) (Figure S1 in Supporting Information). The optimal values of α_{opt} and $\Delta_{0,opt}$ for all studied learning procedures are given in Table 3. The use of the binary SMILES notation was identified to be more effective compared to the decimal format in terms of RMSE for the training and testing pools using the ATransformedBP algorithm. However, in the case of iRPROP$^-$ learning, in the RMSE using the decimal and binary SMILES representations showed a close resemblance during the testing phase. Also, ATransformedBP and iRPROP$^-$ were nearly equivalent for the testing set pools using the decimal notation of the SMILE input, yet iRPROP$^-$ performed marginally better.

One notable observation is that the RMSE levels for both learning algorithms do not drop below a value of 0.150 in the training sets. The values of the learning rate Δ_0 and affine

https://doi.org/10.1021/acs.jcim.0c01366
J. Chem. Inf. Model. 2021, 61, 2175−2186

transformation factor α are crucial in the RNN model development. Figure 3 illustrates that an optimal α_{opt} allows

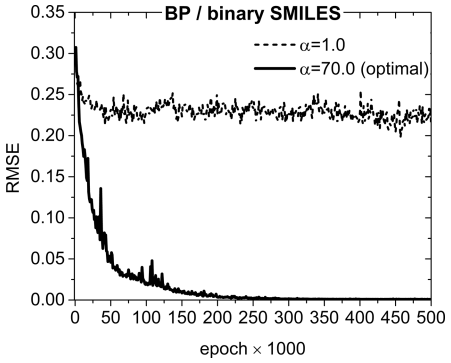


Figure 3. Backpropagation learning algorithm with α = 1.0 and ATransformedBP with α_{opt} = 70.0.

the convergence of learning to reach a zero level when using a binary SMILES input format. When applying the BP algorithm under nonoptimized conditions, the RMSE saturates at 0.23, thereby prohibiting reaching the global minimum.

Examination of Prognostic Capabilities of RNN Architectures with Optimization. Using the optimized values of the learning parameters, the prediction pattern was extracted for the Elman RNN with ATransformedBP and iRPROP$^-$ learning algorithms (Figures 4–7). The training sampling pool consisted of 95% of an original database (1140 units). The remaining 5% (60 examples) was used to study the prediction capabilities of the obtained models. The results were obtained for the learning and testing sample pools after 500 epochs. The optimized parameter of the trained and predicted outputs dependent on the choice of the activation function are given in Tables 2 and 3.

The RMSE evolution for the training and testing sample sets for the binary SMILES representation using the designed learning methods (1136 hidden nodes RNN architecture) are shown as a function of the number of epochs in Figures 4(A) and 5(A) for ATransformedBP and iRPROP$^-$ learning algorithm, respectively. The training-dependent loss function for iRPROP$^-$ learning converges (to zero) after about 120 epochs, while ATransformedBP learning converges only after a larger number of epochs. In addition, the maximum RSD for iRPROP$^-$ learning (Figure 5(B)) was six times greater than that for ATransformed (Figure 4(B)). The predicted values of the dielectric constants using ATransformedBP were in better agreement with the original data than those derived using iRPROP$^-$. The average RSD, in this case, was less than 3% (Figure 4(C)), whereas the maximum of RSD never exceeded 30%. On the contrary, for the iRPROP$^-$ learning strategy, the average RSD was found to be 3.1%. Yet, the maximum RSD parameter did not exceed 29% (Figure 5(C)). While activation with the RELU function was applied only for ATransformed BP algorithm, similar learning and prognostic results were obtained for Tanh activation when comparing the performance of the ATransformedBP and iRPROP$^-$ in the training and testing phases (Figures S2–S5). These findings indicate slightly superior prognosing capabilities of TransformedBP compared to resilient iRPROP$^-$.

It was found that resilient backpropagation iRPROP$^-$ with optimized Δ_0 and ATransformedBP exhibit similar prediction trends for the decimal SMILES representation RNN model (142 hidden nodes). As shown in Figures 6(A) and 7(A) for ATransformedBP and iRPROP$^-$ learning algorithms, respectively, iRPROP- learning leads to higher performance while being noisier. Thus, the RMSE in the training phases differed nearly 1.3-fold. The maximum RSD for ATransformedBP learning was 25.4% (Figures 6(B)), which is nearly twice greater than that for iRPROP$^-$ (Figure 7(B)). Taking into account prognostic results, the average RSD in the case of ATransformedBP was about 4.3% (Figure 6(C)) with a maximum RSD of 28.6%. On the contrary, for the iRPROP$^-$ learning strategy, the average RSD was found to be slightly higher (4.85%), whereas the maximum RSD of 21.7% was lower (Figure 7(C)).

These results demonstrated sufficient prediction capabilities of RNN models with decimal SMILES notation (142 hidden nodes), conceding to RNN with binary SMILES representation (1136 hidden nodes).

A detailed analysis of all polymers of interest revealed several clusters grouped by SMILES similarity and, consequently, certain ranges of values deviation of molecular weights as well as dielectric constants. The individual clusters correlate only moderately with each other, yet the values of ε within the clusters are quite scattered. The prediction accuracy can be further improved when the test sample is preliminarily assigned from the appropriate cluster. The RMSE value of the testing set consisting of samples corresponding to only one cluster decreased almost 1.5 times in comparison to using the testing set generated from samples of different clusters (Figure S4 in Supporting Information). When considering training and

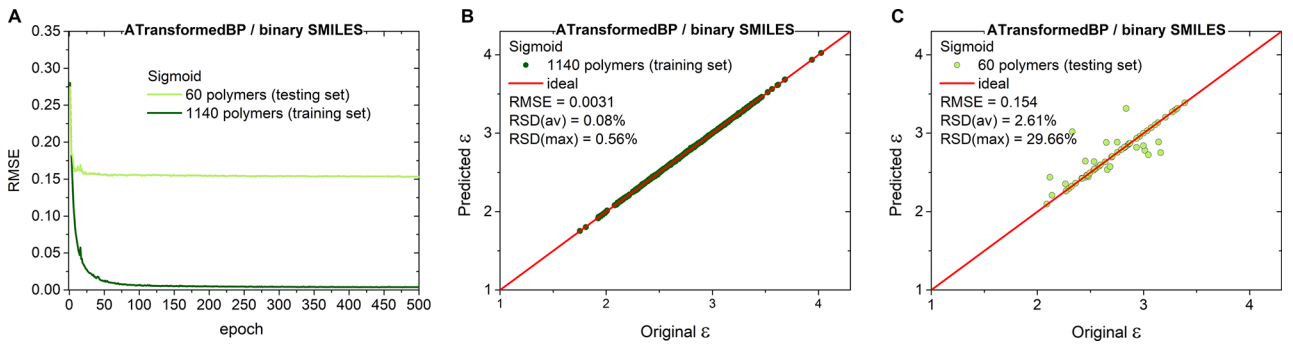


Figure 4. Single-layer Elman RNN prediction model with ATransformedBP learning algorithm for binary SMILES representation. Sigmoid activation is used, and α = 160. (A) RMSE as a function of epoch for training and testing data sets. (B) Parity plot for the training set. (C) Parity plot for the testing set.

https://doi.org/10.1021/acs.jcim.0c01366
J. Chem. Inf. Model. 2021, 61, 2175–2186

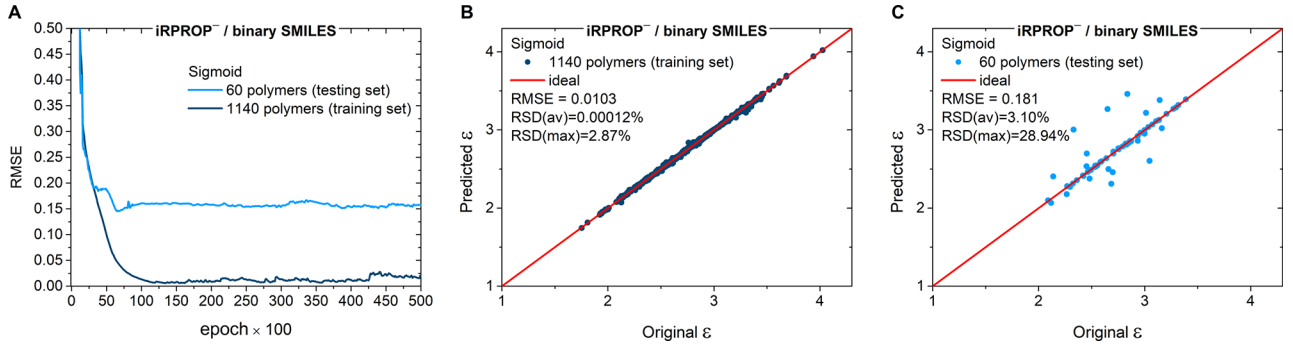


Figure 5. Single-layer Elman RNN prediction model with iRPROP⁻ learning algorithm for binary SMILES representation. Sigmoid activation was used, and $\Delta_0 = 0.015$. (A) RMSE as a function of epoch for training and testing data sets. (B) Parity plot for the training set. (C) Parity plot for the testing set.

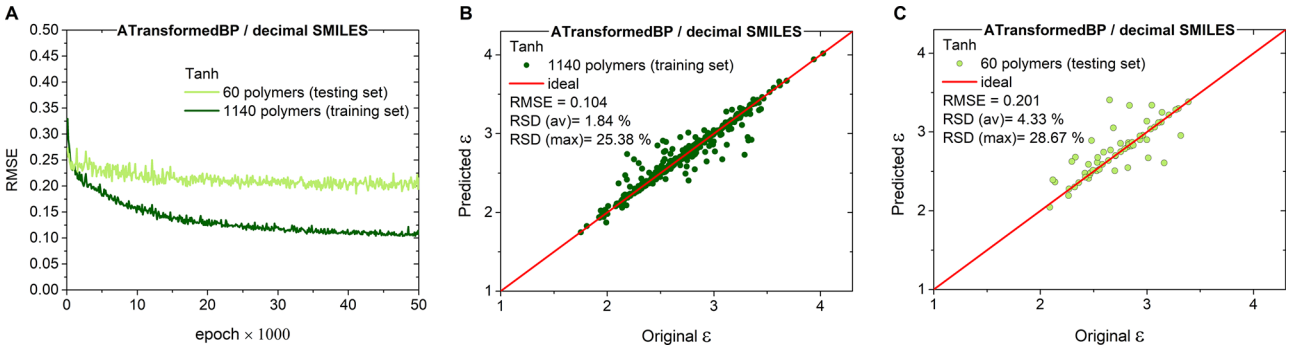


Figure 6. Single-layer Elman RNN prediction model with ATransformedBP learning algorithm for decimal SMILES representation. Tanh activation was used, and $\alpha = 1.0$. (A) RMSE as a function of epoch for training and testing data sets. (B) Parity plot for the training set. (C) Parity plot for the testing set.

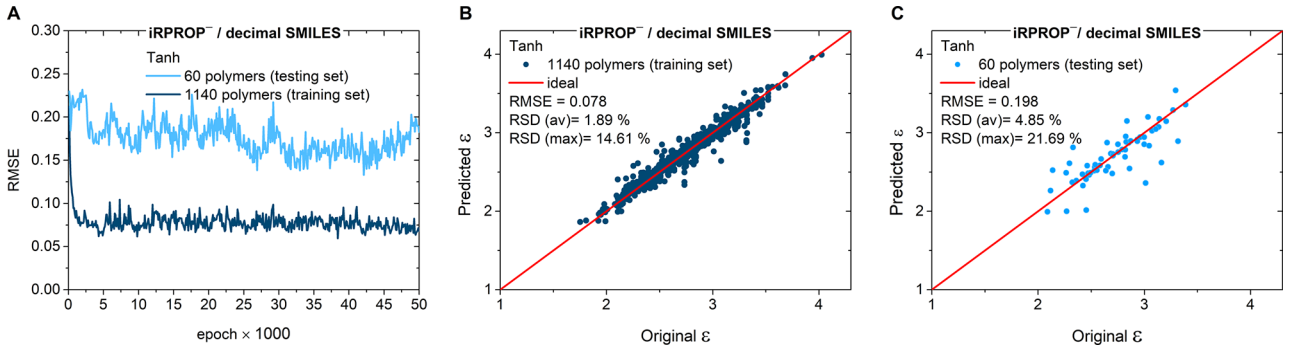


Figure 7. Single-layer Elman RNN prediction model with iRPROP⁻ learning algorithm for decimal SMILES representation. Tanh activation was used, and $\Delta_0 = 0.002$. (A) RMSE as a function of epoch for training and testing data sets. (B) Parity plot for the training set. (C) Parity plot for the testing set.

testing for the original data set with multiple clusters, the accuracy was found to be satisfactory when using a 5% testing set. The increase of the testing set size from 5% to 10% of the original data set did not significantly influence the prediction performance. Namely, in the case of the RNN model with ATransformedBP learning, the maximum value of RSD(max) for a 10% testing set was only 0.5% higher than that of a 5% testing set (Figures S2 and S3 in Supporting Information).

To examine the error distribution of RNN prediction models, error distributions for the training and testing sets were calculated for ATransformedBP and iRPROP⁻ learning algorithms. Figure 8 shows the relative number of samples as a function of RSD. We see that most samples have small RSD values. As shown in Table 3, 85% of the prognostic results indeed do not exceed 10% RSD using the binary SMILES input representation (1136 hidden nodes). For the training sampling pool, this parameter was close to 100% regardless of the used activation function.

CONCLUSION

We have developed a simple, customizable, and user-friendly RNN codebase for the prediction of material properties based on nonlinear structure−property relationships. High-throughput ML models were implemented for solving the problem of molecular structure−property relationships. In particular, the developed RNN models were verified with the example of the dielectric property prediction of computationally created polymeric sequences. A polymer genome is challenging due

https://doi.org/10.1021/acs.jcim.0c01366
J. Chem. Inf. Model. 2021, 61, 2175−2186

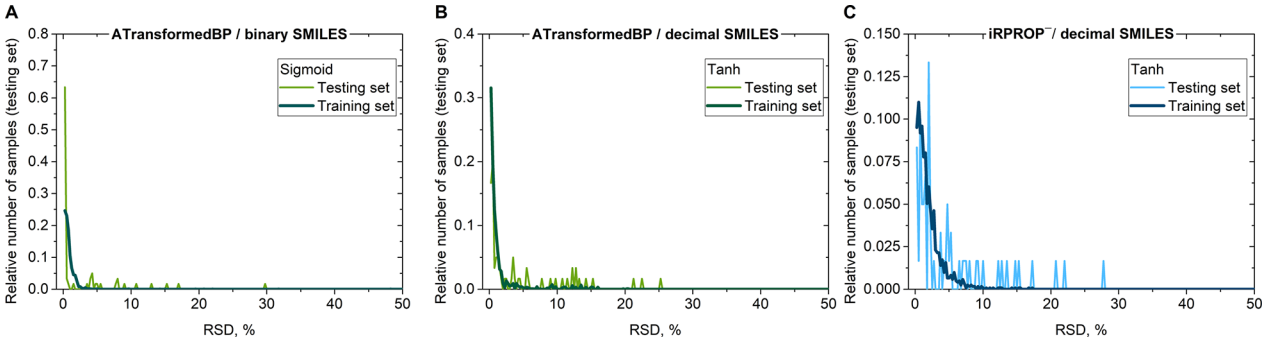


Figure 8. Prediction error distribution of RNN prediction model with ATransformedBP and iRPROP$^-$ learning algorithms for binary (1136 nodes) and decimal (142 nodes) SMILES representation. Activation function and optimized parameter used are (A) Sigmoid, α = 160.0, (B) Tanh, α = 1.0, and (C) Tanh, Δ_0 = 0.002.

to uncontrollable morphologies (e.g., amorphous vs semi-crystalline) to which different properties are sensitive and correlate differently. Our quantum-mechanically validated computational data set for a specific property of a dielectric constant in polymers with a well-controlled amorphous morphology offers a much needed test bed,[77] for which this article presents ground optimizations of several advanced algorithms as an open software to facilitate the development of further algorithms using the well-controlled dielectric polymer data set.

A detailed description of the developed backpropagation algorithm with input affine transformation, ATransformedBP, was provided. Optimization studies were performed for the iRPROP$^-$ learning algorithm with respect to the initial weight update parameter. The prediction accuracy of the dielectric constant ε using the ATransformedBP algorithm was found to be similar or in some instances slightly superior to resilient propagation learning algorithms, specifically iRPROP$^-$. The decimal and binary numerical input formats of the SMILES representation were implemented and tested. The binary format of SMILES was found to be more informative compared to the decimal format for the investigated polymers. The average RSD parameter of the designed algorithms did not exceed 5%, while the maximum RSD value was never higher than 30%. This suggests a superior efficiency of the optimized RNN models in utilizing the scrutiny of the structure−activity nonlinear relationships. Apart from predicting the dielectric permeability, a wide range of possible property prognoses will likely benefit from the developed protocols. Ongoing efforts include a comparison of RNN architectures with long short-term memory (LSTM) and gated recurrent units (GRU) neural networks.

While many researchers remain skeptical about whether to rely on external frameworks or to come up with self-developed software, the aspect of the time expenditure might bias toward the ready-to-use option. In the current work, we demonstrated that the workload of managing a custom-made program is well worth it when being able to further develop or optimize an available market solution.

DATA AND SOFTWARE AVAILABILITY

The following program and data files are available free of charge at https://zenodo.org/record/4280446#.X7qBlM1KiUk: (1) RNN_ATransformedBP_opt_pol file: RNN code using affine transformed backpropagation learning algorithm. (2) RNN_iRPROP-_opt_pol file: RNN code using resilient iRPROP$^-$ learning algorithm with optimization of the parameter of initial weights update. (3) smiles_mod.dat and eps.dat: Input files containing polymer structure information (SMILES) and dielectric property values (ε).

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.0c01366.

Figures S1−S8 (PDF)

AUTHOR INFORMATION

Corresponding Author

Aiichiro Nakano − *Collaboratory of Advanced Computing and Simulations, Department of Computer Science, Department of Physics & Astronomy, Department of Chemical Engineering & Materials Science, and Department of Biological Sciences, University of Southern California, Los Angeles, California 90089, United States;* orcid.org/0000-0003-3228-3896; Email: anakano@usc.edu

Authors

Antonina L. Nazarova − *Department of Chemistry, Loker Hydrocarbon Research Institute, and USC Bridge Institue, University of Southern California, Los Angeles, California 90089, United States*

Liqiu Yang − *Collaboratory of Advanced Computing and Simulations, Department of Computer Science, Department of Physics & Astronomy, Department of Chemical Engineering & Materials Science, and Department of Biological Sciences, University of Southern California, Los Angeles, California 90089, United States*

Kuang Liu − *Collaboratory of Advanced Computing and Simulations, Department of Computer Science, Department of Physics & Astronomy, Department of Chemical Engineering & Materials Science, and Department of Biological Sciences, University of Southern California, Los Angeles, California 90089, United States*

Ankit Mishra − *Collaboratory of Advanced Computing and Simulations, Department of Computer Science, Department of Physics & Astronomy, Department of Chemical Engineering & Materials Science, and Department of Biological Sciences, University of Southern California, Los Angeles, California 90089, United States*

Rajiv K. Kalia − *Collaboratory of Advanced Computing and Simulations, Department of Computer Science, Department of Physics & Astronomy, Department of Chemical Engineering & Materials Science, and Department of Biological Sciences,*

University of Southern California, Los Angeles, California 90089, United States

Ken-ichi Nomura − *Collaboratory of Advanced Computing and Simulations, Department of Computer Science, Department of Physics & Astronomy, Department of Chemical Engineering & Materials Science, and Department of Biological Sciences, University of Southern California, Los Angeles, California 90089, United States*

Priya Vashishta − *Collaboratory of Advanced Computing and Simulations, Department of Computer Science, Department of Physics & Astronomy, Department of Chemical Engineering & Materials Science, and Department of Biological Sciences, University of Southern California, Los Angeles, California 90089, United States;* orcid.org/0000-0003-4683-429X

Pankaj Rajak − *Argonne Leadership Computing Facility, Argonne National Laboratory, Lemont, Illinois 60439, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.0c01366

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported by the Office of Naval Research through Multi-University Research Initiative (MURI) Grant N00014-17-1-2656. The simulations were performed at the Argonne Leadership Computing Facility under DOE INCITE and Aurora Early Science programs as well as at the Center for Advanced Research Computing of the University of Southern California.

■ REFERENCES

(1) Hasegawa, K.; Deushi, T.; Yaegashi, O.; Miyashita, Y.; Sasaki, S. Artificial Neural Network Studies in Quantitative Structure-Activity Relationships of Antifungal Azoxy Compounds. *Eur. J. Med. Chem.* **1995**, *30*, 569−574.

(2) Le, T.; Epa, V. C.; Burden, F. R.; Winkler, D. A. Quantitative Structure-Property Relationship Modeling of Diverse Materials Properties. *Chem. Rev.* **2012**, *112*, 2889−2919.

(3) Sliwoski, G.; Kothiwale, S.; Meiler, J.; Lowe, E. W. Computational Methods in Drug Discovery. *Pharmacol. Rev.* **2014**, *66*, 334−395.

(4) Idakwo, G.; Thangapandian, S.; Luttrell, J.; Zhou, Z.; Zhang, C.; Gong, P. Deep Learning-Based Structure-Activity Relationship Modeling for Multi-Category Toxicity Classification: A Case Study of 10k Tox21 Chemicals with High-Throughput Cell-Based Androgen Receptor Bioassay Data. *Front. Physiol.* **2019**, *10*, 1−13.

(5) von Lilienfeld, O. A.; Burke, K. Retrospective on a Decade of Machine Learning for Chemical Discovery. *Nat. Commun.* **2020**, *11*, 4895.

(6) Jain, A.; Persson, K. A.; Ceder, G. Research Update: The Materials Genome Initiative: Data Sharing and the Impact of Collaborative Ab Initio Databases. *APL Mater.* **2016**, *4*, 053102.

(7) Butler, K. T.; Davies, D. W.; Cartwright, H.; Isayev, O.; Walsh, A. Machine Learning for Molecular and Materials Science. *Nature* **2018**, *559*, 547−555.

(8) Van Drie, J. H. Computer-Aided Drug Design: The Next 20 years. *J. Comput.-Aided Mol. Des.* **2007**, *21*, 591−601.

(9) Gil, Y.; Greaves, M.; Hendler, J.; Hirsh, H. Amplify Scientific Discovery with Artificial Intelligence. *Science* **2014**, *346*, 171−172.

(10) Himanen, L.; Geurts, A.; Foster, A. S.; Rinke, P. Data-Driven Materials Science: Status, Challenges, and Perspectives. *Adv. Sci.* **2019**, *6*, 1900808.

(11) Steinhauser, M.; Hiermaier, S. A Review of Computational Methods in Materials Science: Examples from Shock-Wave and Polymer Physics. *Int. J. Mol. Sci.* **2009**, *10*, 5135−5216.

(12) Spakowitz, A. J. Polymer Physics across Scales: Modeling the Multiscale Behavior of Functional Soft Materials and Biological Systems. *J. Chem. Phys.* **2019**, *151*, 230902.

(13) Sharma, V.; Wang, C. C.; Lorenzini, R. G.; Ma, R.; Zhu, Q.; Sinkovits, D. W.; Pilania, G.; Oganov, A. R.; Kumar, S.; Sotzing, G. A.; Boggs, S. A.; Ramprasad, R. Rational Design of All Organic Polymer Dielectrics. *Nat. Commun.* **2014**, *5*, 4845.

(14) Kim, C.; Pilania, G.; Ramprasad, R. From Organized High-Throughput Data to Phenomenological Theory Using Machine Learning: The Example of Dielectric Breakdown. *Chem. Mater.* **2016**, *28*, 1304−1311.

(15) Kim, C.; Chandrasekaran, A.; Huan, T. D.; Das, D.; Ramprasad, R. Polymer Genome: A Data-Powered Polymer Informatics Platform for Property Predictions. *J. Phys. Chem. C* **2018**, *122*, 17575−17585.

(16) Mishra, A.; Baradwaj, N.; Bassman, L.; Horton, B. K.; Tiwari, S.; Hong, S.; Krishnamoorthy, A.; Moen, E.; Rajak, P.; Kalia, R.; Nakano, A.; Nomura, K.-i.; Shimojo, F.; Vashishta, P. Materials Genome Software Framework: Scalable Parallel Simulation, Virtual Reality Visualization and Machine Learning. In *International Conference on Scientific Computing*; CSREA Press, 2019; pp 125−131.

(17) Naserifar, S.; Brooks, D. J.; Goddard, W. A.; Cvicek, V. Polarizable Charge Equilibration Model for Predicting Accurate Electrostatic Interactions in Molecules and Solids. *J. Chem. Phys.* **2017**, *146*, 124117.

(18) Liu, K.; Hong, S.; Kalia, R. K.; Nakano, A.; Nomura, K.; Rajak, P.; Tiwari, S.; Vashishta, P.; Luo, Y.; Romero, N. A.; S. Naserifar, S.; Goddard, W. A.; Kunaseth, M., Shift-Collapse Acceleration of Generalized Polarizable Reactive Molecular Dynamics for Machine Learning-Assisted Computational Synthesis of Layered Materials. In *2018 IEEE/ACM 9th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems (scalA)*; IEEE, 2018; pp 41−48.

(19) Li, Y.; Nomura, K.; Insley, J.; Morozov, V.; Kumaran, K.; Romero, N. A.; Goddard III, W. A.; Kalia, R. K.; Nakano, A.; Vashishta, P. Scalable Reactive Molecular Dynamics Simulations for Computational Synthesis. *Comput. Sci. Eng.* **2019**, *21*, 64−75.

(20) Wiswesser, W. J. The Wiswesser Line Formula Notation. *Chem. Eng. News* **1952**, *30*, 3523−3526.

(21) Zhang, Y.-F.; Wang, X.; Kaushik, A. C.; Chu, Y.; Shan, X.; Zhao, M.-Z.; Xu, Q.; Wei, D.-Q. Spvec: A Word2vec-Inspired Feature Representation Method for Drug-Target Interaction Prediction. *Front. Chem.* **2020**, *7*, 1−11.

(22) Jaeger, S.; Fulle, S.; Turk, S. Mol2vec: Unsupervised Machine Learning Approach with Chemical Intuition. *J. Chem. Inf. Model.* **2018**, *58*, 27−35.

(23) Honda, S.; Shi, S.; Ueda, H. Smiles Transformer: Pre-Trained Molecular Fingerprint for Low Data Drug Discovery. *arXiv:1911.04738*, 2019. https://arxiv.org/abs/1911.04738 (accessed April 2021).

(24) Deng, T.; Jia, G.-z. Prediction of Aqueous Solubility of Compounds Based on Neural Network. *Mol. Phys.* **2020**, *118*, No. e1600754.

(25) Heidari, E.; Sobati, M. A.; Movahedirad, S. Accurate Prediction of Nanofluid Viscosity Using a Multilayer Perceptron Artificial Neural Network (Mlp-Ann). *Chemom. Intell. Lab. Syst.* **2016**, *155*, 73−85.

(26) Xie, T.; Grossman, J. C. Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. *Phys. Rev. Lett.* **2018**, *120*, 145301.

(27) Liu, K. C.; Nomura, K.-i.; Kalia, R. K.; Nakano, A.; Vashishta, P.; Rajak, P. Graph Neural Network Analysis of Layered Material Phases; SpringSim-HPC, 2019.

(28) Bernazzani, L.; Duce, C.; Micheli, A.; Mollica, V.; Sperduti, A.; Starita, A.; Tiné, M. R. Predicting Physical-Chemical Properties of Compounds from Molecular Structures by Recursive Neural Networks. *J. Chem. Inf. Model.* **2006**, *46*, 2030−2042.

(29) Cook, S. *Cuda Programming: A Developer's Guide to Parallel Computing with Gpus*; Morgan Kaufmann: Waltham, MA, 2012.

(30) *NVIDIA DGX A100 System Architecture*; Technical White Paper WP-10083-001_v01; NVIDIA, 2020.
(31) Munshi, A. The Opencl Specification. In *2009 IEEE Hot Chips 21 Symposium (HCS)*, 2009.
(32) Ashbaugh, B.; Bader, A.; Brodman, J. C.; Hammond, J.; Kinsner, M.; Pennycook, J.; Schulz, R.; Sewall, J. D. Data Parallel C++: Enhancing Sycl through Extensions for Productivity and Performance. In *IWOCL '20: Proceedings of the International Workshop on OpenCL*, 2020.
(33) Rehman, M. Z.; Nawi, N. M. Studying the Effect of Adaptive Momentum in Improving the Accuracy of Gradient Descent Back Propagation Algorithm on Classification Problems. *Int. J. Mod. Phys.: Conf. Ser.* **2012**, *09*, 432−439.
(34) Van Ooyen, A.; Nienhuis, B. Improving the Convergence of the Back-Propagation Algorithm. *Neural Netw.* **1992**, *5*, 465−471.
(35) Attoh-Okine, N. O. Analysis of Learning Rate and Momentum Term in Backpropagation Neural Network Algorithm Trained to Predict Pavement Performance. *Adv. Eng. Softw.* **1999**, *30*, 291−302.
(36) Jaeger, S. The Golden Ratio of Learning and Momentum 2020, Preprint at https://arxiv.org/abs/2006.04751.
(37) Paeedeh, N.; Ghiasi-Shirazi, K. Improving the Backpropagation Algorithm with Consequentialism Weight Updates over Mini-Batches. *arXiv:2006.04751*, 2020. https://arxiv.org/abs/2003.05164 (accessed April 2021).
(38) Olgac, A.; Karlik, B. Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks. *Int. J. Artificial Intelligence Expert Systems* **2011**, *1*, 111−122.
(39) Nawi, N. M.; Khan, A.; Rehman, M. Z.; Chiroma, H.; Herawan, T. Weight Optimization in Recurrent Neural Networks with Hybrid Metaheuristic Cuckoo Search Techniques for Data Classification. *Math. Probl. Eng.* **2015**, *2015*, 1−12.
(40) Heravi, A. R.; Abed Hodtani, G. A New Correntropy-Based Conjugate Gradient Backpropagation Algorithm for Improving Training in Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* **2018**, *29*, 6252−6263.
(41) Bishop, C. M. *Neural Networks for Pattern Recognition*; Oxford University Press: New York, 1995.
(42) Nomizu, K. *Affine Differential Geometry: Geometry of Affine Immersions*; Cambridge University Press: Cambridge, U.K., 1994.
(43) Kůrková, V. Kolmogorov's Theorem and Multilayer Neural Networks. *Neural Netw.* **1992**, *5*, 501−506.
(44) Dai, J.; Xiong, S. An Adversarial Attack against Stacked Capsule Autoencoder. *arXiv:2010.07230*, 2020. https://arxiv.org/abs/2010.07230 accessed April 2021).
(45) Zhong, L.; Bai, C.; Li, J.; Chen, T.; Li, S. Facial Expression Recognition Method Based on a Part-Based Temporal Convolutional Network with a Graph-Structured Representation. In *Artificial Neural Networks and Machine Learning - ICANN 2020*; Farkaš, I., Masulli, P., Wermter, S., Eds.; Springer: Cham, 2020; pp 609−620.
(46) Choi, H. Persistent Hidden States and Nonlinear Transformation for Long Short-Term Memory. *Neurocomputing* **2019**, *331*, 458−464.
(47) Mujika, A.; Meier, F.; Steger, A. Fast-Slow Recurrent Neural Networks. *arXiv:1705.08639*, 2017. https://arxiv.org/abs/1705.08639 (accessed April 2021).
(48) Reed, R. D.; Marks, R. J., II *Neural Smithing*; MIT Press: Cambridge, MA, 1999.
(49) Igel, C.; Hüsken, M. Empirical Evaluation of the Improved Rprop Learning Algorithms. *Neurocomputing* **2003**, *50*, 105−123.
(50) Xinxing, P.; Lee, B.; Chunrong, Z. A Comparison of Neural Network Backpropagation Algorithms for Electricity Load Forecasting. In *2013 IEEE International Workshop on Inteligent Energy Systems (IWIES)*, 2013; pp 22−27.
(51) Erfiani, A. S. B. Comparison of Backpropagation and Resilient Backpropagation Algorithms in Non-Invasive Blood Glucose Measuring Device. *Int. J. Eng. Res.* **2017**, *8*, 153−157.
(52) Igel, C.; Hüsken, M. Improving the Rprop Learning Algorithm. In *Proceedings of the Second International Symposium on Neural Computation, NC 2000*; Bothe, H., Rojas, R., Eds.; ICSC Academic Press: Canada/Switzerland, 2000; pp 115−121.
(53) Nazarova, A. Polymeric Property Prediction Using Recurrent Neural Networks with Optimizations; Zenodo, 2020. https://zenodo.org/record/4280446#.YHnH5j8pDmE (accessed April 2021).
(54) Gerchev, I. Front-End Frameworks: Costom vs Ready-to-Use Solutions. SitePoint. https://www.sitepoint.com/front-end-frameworks-custom-vs-ready-use-solutions/ (accessed October 23, 2020).
(55) Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A. Pytorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv:1912.01703*, 2019. https://arxiv.org/abs/1912.01703 (accessed April 2021).
(56) Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; Kudlur, M.; Levenberg, J.; Monga, R.; Moore, S.; Murray, D. G.; Steiner, B.; Tucker, P.; Vasudevan, V.; Warden, P.; Wicke, M.; Yu, Y.; Zheng, X. Tensorflow: A System for Large-Scale Machine Learning. In *12th USENIX Symposium on Operating Systems Design and Implementation*; USENIX Association, 2016; pp 265−283.
(57) Jiménez-Luna, J.; Grisoni, F.; Schneider, G. Drug Discovery with Explainable Artificial Intelligence. *Nat. Mach. Intell.* **2020**, *2*, 573−584.
(58) Cui, Y.; Dong, Q.; Hong, D.; Wang, X. Predicting Protein-Ligand Binding Residues with Deep Convolutional Neural Networks. *BMC Bioinf.* **2019**, *20*, 93.
(59) Elton, D. C.; Boukouvalas, Z.; Butrico, M. S.; Fuge, M. D.; Chung, P. W. Applying Machine Learning Techniques to Predict the Properties of Energetic Materials. *Sci. Rep.* **2018**, *8*, 9059.
(60) Stanev, V.; Oses, C.; Kusne, A. G.; Rodriguez, E.; Paglione, J.; Curtarolo, S.; Takeuchi, I. Machine Learning Modeling of Superconducting Critical Temperature. *npj Comput. Mater.* **2018**, *4*, 29.
(61) Conduit, B. D.; Jones, N. G.; Stone, H. J.; Conduit, G. J. Design of a Nickel-Base Superalloy Using a Neural Network. *Mater. Des.* **2017**, *131*, 358−365.
(62) Mannodi-Kanakkithodi, A.; Pilania, G.; Huan, T. D.; Lookman, T.; Ramprasad, R. Machine Learning Strategy for Accelerated Design of Polymer Dielectrics. *Sci. Rep.* **2016**, *6*, 20952.
(63) Pilania, G.; Wang, C.; Jiang, X.; Rajasekaran, S.; Ramprasad, R. Accelerating Materials Property Predictions Using Machine Learning. *Sci. Rep.* **2013**, *3*, 2810.
(64) Yamada, H.; Liu, C.; Wu, S.; Koyama, Y.; Ju, S.; Shiomi, J.; Morikawa, J.; Yoshida, R. Predicting Materials Properties with Little Data Using Shotgun Transfer Learning. *ACS Cent. Sci.* **2019**, *5*, 1717−1730.
(65) Müller, A. T.; Hiss, J. A.; Schneider, G. Recurrent Neural Network Model for Constructive Peptide Design. *J. Chem. Inf. Model.* **2018**, *58*, 472−479.
(66) Amabilino, S.; Pogány, P.; Pickett, S. D.; Green, D. V. S. Guidelines for Recurrent Neural Network Transfer Learning-Based Molecular Generation of Focused Libraries. *J. Chem. Inf. Model.* **2020**, *60*, 5699−5713.
(67) Mozaffar, M.; Bostanabad, R.; Chen, W.; Ehmann, K.; Cao, J.; Bessa, M. A. Deep Learning Predicts Path-Dependent Plasticity. *Proc. Natl. Acad. Sci. U. S. A.* **2019**, *116*, 26414−26420.
(68) Shewalkar, A.; Nyavanandi, D.; Ludwig, S. A. Performance Evaluation of Deep Neural Networks Applied to Speech Recognition: Rnn, Lstm and Gru. *J. Artif. Intell. Soft Comput. Res.* **2019**, *9*, 235−245.
(69) Karita, S.; Chen, N.; Hayashi, T.; Hori, T.; Inaguma, H.; Jiang, Z.; Someki, M.; Soplin, N. E. Y.; Yamamoto, R.; Wang, X.; Watanabe, S.; Yoshimura, T.; Zhang, W. A Comparative Study on Transformer Vs Rnn in Speech Applications. *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* **2019**, 449−456.
(70) Keren, G.; Schuller, B. Convolutional Rnn: An Enhanced Model for Extracting Features from Sequential Data. In *2016 International Joint Conference on Neural Networks (IJCNN)*; IEEE, 2016; p 3412.

(71) Kong, P.; Drechsler, S.; Balog, S.; Schrettl, S.; Weder, C.; Kilbinger, A. F. M. Synthesis and Properties of Poly(Norbornene)S with Lateral Aramid Groups. *Polym. Chem.* **2019**, *10*, 2057−2063.
(72) Li, Q.; Hu, J.; Lv, J.; Wang, X.; Shao, S.; Wang, L.; Jing, X.; Wang, F. Through-Space Charge-Transfer Polynorbornenes with Fixed and Controllable Spatial Alignment of Donor and Acceptor for High-Efficiency Blue Thermally Activated Delayed Fluorescence. *Angew. Chem., Int. Ed.* **2020**, *59*, 20174−20182.
(73) Tetsuka, H.; Isobe, K.; Hagiwara, M. Synthesis and Properties of Addition-Type Poly(Norbornene)S with Siloxane Substituents. *Polym. J.* **2009**, *41*, 643−649.
(74) Shin, B.-G.; Jang, M.-S.; Yoon, D. Y.; Heitz, W. Vinyl-Type Polymerization of Norbornene Dicarboxylic Acid Dialkyl Esters. *Macromol. Rapid Commun.* **2004**, *25*, 728−732.
(75) Li, Y.; Nomura, K.-I.; Insley, J. A.; Morozov, V.; Kumaran, K.; Romero, N. A.; Goddard, W. A.; Kalia, R. K.; Nakano, A.; Vashishta, P. Scalable Reactive Molecular Dynamics Simulations for Computational Synthesis. *Comput. Sci. Eng.* **2019**, *21*, 64−75.
(76) Liu, K.; Nazarova, A.; Mishra, A.; Chen, Y.; Lyu, H.; Xu, L.; Yin, Y.; Zhao, Q.; Kalia, R. K.; Nakano, A.; Nomura, K.; Vashista, P.; Rajak, P. Dielectric Polymer Genome: Integrating Valence-Aware Polarizable Reactive Force Fields and Machine Learning. In *Proceedings of International Conference on Scientific Computing, CSC'20*, 2020.
(77) Mishra, A.; Chen, L.; Li, Z.; Nomura, K.-i.; Krishnamoorthy, A.; Fukushima, S.; Tiwari, S. C.; Kalia, R. K.; Nakano, A.; Ramprasad, R.; Sotzing, G.; Cao, Y.; Shimojo, F.; Vashishta, P. Computational Framework for Polymer Synthesis to Study Dielectric Properties Using Polarizable Reactive Molecular Dynamics *arXiv:2011.09571*, 2020. https://arxiv.org/abs/2011.09571 (accessed April 2021).
(78) Pojas, R. *Neural Networks*; Springer-Verlag: Berlin, 1996.
(79) Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. Learning Representations by Back-Propagating Errors. *Nature* **1986**, *323*, 533−536.
(80) Riedmiller, M.; Braun, H. A Direct Adaptive Method for Faster Backpropagation Learning: The Rprop Algorithm. In *IEEE International Conference on Neural Networks*; IEEE, 1993; pp 586−591.

https://doi.org/10.1021/acs.jcim.0c01366
J. Chem. Inf. Model. 2021, 61, 2175−2186