JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

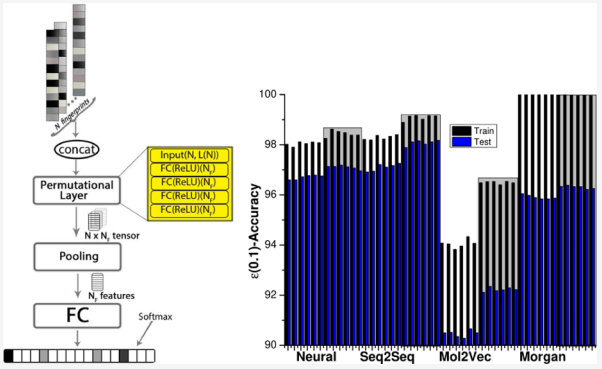
Permutationally Invariant Deep Learning Approach to Molecular Fingerprinting with Application to Compound Mixtures

Andrei Buin,* Hung Yi Chiang, S. Andrew Gadsden,* and Faraz A. Alderson

Cite This: *J. Chem. Inf. Model.* 2021, 61, 631−640 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Recent advancements in deep learning have led to widespread applications of its algorithms to synthetic planning and reaction predictions in the field of chemistry. One major area, known as supervised learning, is being explored for predicting certain properties such as reaction yields and types. Many chemical descriptors known as fingerprints are being explored as potential candidates for reaction properties prediction. However, there are few studies that describe the permutational invariance of chemical fingerprints, which are concatenated at some stage before being fed to deep learning architecture. In this work, we show that by utilizing permutational invariance, we consistently see improved results in terms of accuracy relative to previously published studies. Furthermore, we are able to accurately predict hydrogen peroxide loss with our own dataset, which consists of more than 20 ingredients in each chemical formulation.



1. INTRODUCTION

Recent applications of machine learning (ML) algorithms[1−5] to the field of drug discovery[6] and retrosynthetic planning[7] made it possible to explore novel compounds that have not been previously explored or sought after.[8−10] The majority of the material published still is in the realm of supervised learning. Many chemical descriptors are coming from traditional based[11] approaches such as Morgan,[12] QSAR,[13] and physical descriptors. With the advancement of ML methods, many chemical descriptors have emerged from the realm of unsupervised approaches. In particular, one of the areas of focus has been approaches which express complex, graph-like molecular representations as simpler vector representations that still retain rich information and involve SMILES[14] strings. These approaches include sequence-to-sequence (Seq2Seq),[6] molecule-to-vector (Mol2Vec)[15] NLP-inspired embedding, and variational autoencoder (VAE)-based encodings.[16] One should note, however, that there are also recent deep learning (DL) approaches which operate on graph-structured molecular data.[17] Given all those vector representations, or in other words molecular fingerprints (FPs), one could start combining them into one big FP for a variety of prediction problems at hand, as was done in the work of Sandfort et al.[18] Many of them use concatenation as a "glueing" operation at the molecular or reaction level.[1,4,18] However, no particular preferences for ordering have been given.

Concatenation alone represents a particular instance of ordering in the combined representation of items within some set. If order matters, as is the case for sequences, then recurrent neural networks (RNNs)[9] are usually the architecture of choice where unrolling usually captures either spatial or temporal dependence in a sequence. However, when order is irrelevant, such as in the case of reactions and mixtures, one can use ordinary feed-forward DL-neural networks (NNs). Quite often, the problem lies in the fact that there is no particular preference on how to concatenate molecular FPs or physical descriptors participating in a reaction, and choosing one ordering versus another would introduce a permutational bias. Thus, reactants and products can be viewed as a set of participating molecular entities and a set of resulting chemical formulations, which are presumably permutationally invariant. It turns out that the problem of sets, more particularly permutational invariant sets, as inputs to a DL architecture has recently gained much attention in theoretical and applied ML.[19−23]

One way to remove bias related to particular ordering is to have canonical ordering.[24] For example, for redox reactions, put the oxidizing agent first, followed by the reducing agent. Another way is by enlarging the dataset such that all possible permutations exist in the dataset without any preference given to any particular order. One can see that in a simple case of three unique reactants, it leads to a six-fold increase in the

Received: September 19, 2020
Published: February 4, 2021

https://dx.doi.org/10.1021/acs.jcim.0c01097
J. Chem. Inf. Model. 2021, 61, 631−640

dataset if we account for all of the permutations. In the case of 10 ingredients, there are over 3×10^6 permutations, thus making it impossible to account for all possible permutations as the number of reactants increases. This poses a significant scalability problem. Yet another way to remove permutational bias is to use permutationally invariant FPs such as Daylight structural/difference[25] where permutationally invariant FPs are constructed based on bitwise OR and XOR—operations on the individual FPs of reactants and products. The work of Schneider et al.[4] explores weighted difference reaction FPs (WDRFPs) where the difference of the FPs of products and reactants is constructed from a variety of chemical FPs, which is later used in reaction type prediction. Additionally, one could utilize a condensed graph of reaction (CGR) or SiRMS mixture approach.[26] One of the main problems with the Daylight FP, SiRMS, and CGR methods is that it is not entirely clear how to introduce relative weights of participating reactants, whereas, in the case of WDRFP, it is applicable only to relatively simple reactions.

In this work, we propose a novel DL neural-net architecture which is immune to permutational bias that combines physically meaningful descriptors of participating agents/reactants. The core fingerprinting methods in our work consist of unsupervised molecular fingerprinting methods based on a variety of DL architectures such as Seq2Seq,[6] VAE[16]—encoder−decoder-based, Mol2Vec[15]—NLP embedding-based, and possibly others.[27] These FPs are combined via permutational layers proposed by Guttenberg et al.[23] As a result, it does not require dataset augmentation nor does it scale linearly with the number of concatenated FPs. This is reflected in a significantly lower number of learnable parameters. Additionally, this method is not only applicable to reaction FPs. It has widespread applicability in property prediction of mixtures as demonstrated on our proprietary H_2O_2 loss prediction dataset, given a number of initial chemical ingredients mixed in each given experiment. Compared to mixtures which oftentimes do not result in products, reactions have products. For products to be incorporated in the reaction FP, one could potentially use another set of permutation layers dedicated to products only. The permutationally invariant encoding in these layers is subtracted/concatenated with the permutationally invariant reactant's latent representation, thus making reaction FPs permutationally invariant. We demonstrate better accuracies on the prediction of reaction types using Wei et al.'s[1] dataset of 16 basic reaction types of alkyl halides and alkenes. A regression task is also performed on our own dataset where H_2O_2 losses in mixtures after incubation are predicted, given the initial H_2O_2 concentrations.

This paper is organized as follows: the computational details on the proposed strategy are explained in Section 2. The results and discussion are provided in Section 3, divided into two tasks: classification with permutational layer and regression with permutationally invariant layers. Concluding remarks are provided in Section 4.

2. COMPUTATIONAL DETAILS

In order to prove the effectiveness of the proposed approach, two different tasks involving FPs were performed: a reaction classification task where the goal was to reproduce the results of Wei et al.[1] using their dataset of reactant/reagent FPs and an H_2O_2 regression task involving our own generated dataset.

2.1. Molecular FPs. For training purposes, all unsupervised FPs (Seq2Seq, VAE, and Mol2Vec) were set as dense 1D arrays with a length of 256, with the exception of Morgan and neural fingerprints (NFPs).[27] The former had a length of 768, while the latter had a length of 156. Additionally, for the regression task, we used Daylight reaction FPs. Daylight considers two types of reaction FPs: structural and difference. Structural reaction FP is the combination of structural FPs for reactants and products within the reaction using the bitwise-OR operator applied to individual reactant and product bit vectors (FPs).[25] Reaction or difference FPs, on the other hand, exploit the XOR operator[25] which is applied to reactants and products that reflect bond changes taking place in the reaction. In H_2O_2 prediction loss, we have found that the best performance for the baseline was achieved by using structural rather than difference 1024 bit FPs based on the reactants' Morgan FPs. Please note that in our case the product stays the same: H_2O_2. All of the chemical manipulations and Daylight with Morgan constituents FPs' constructions were generated in RDKit.[28] NFP[27] generation and a comparison with the work of Wei et al.[1] was partially completed with code from their paper and references within, and the rest was completed in Keras[29] with a TensorFlow[30] back-end.

To generate Seq2seq FPs, the original work of Xu[6] was utilized. An encoder−decoder DL architecture based on a RNN network with two stacked layers of (128) GRU cells with dropout were used to generate Seq2Seq FPs. Dropout was set at 0.5. To learn longer SMILES strings, the Bahdanau attention[31] mechanism was used. The overall process of generating Seq2Seq FPs was split into two stages: (1) learning the lower-dimensional latent representation of SMILES grammar by using the ChEMBL25[9] dataset with a SMILES max length of 80 characters and (2) using a transfer-learning (fine tuning) approach by training with weights obtained from the first stage on our dataset for 10 epochs. After that, Seq2Seq FPs were obtained by feeding SMILES into the encoder network, and then the context vector was extracted as a Seq2Seq FP.

We did not canonicalize SMILES strings due to recent findings suggesting that SMILES augmentation with non-canonical SMILES representation increases reconstruction accuracy.[32,33] As mentioned[9] previously, ChEMBL contains many products with complex scaffolds, such as peptides, which results in 72 million characters with only 51 being unique. We added the nine characters that were present in our dataset, but not in CHEMBL, to the vocabulary of the model. With enhanced vocabulary, we learnt latent representations of a reduced ChEMBL database with 2M SMILES entries in the first stage and then applied transfer learning (fine tuning of weights) on our dataset (262 unique SMILES entries) in the second stage. For now, we disregard all activity data from the ChEMBL database. When the original Seq2Seq model was trained on the reduced CHEMBL dataset and directly used on our dataset's SMILES strings, it gave an 86% reconstruction accuracy via the encoder−decoder Seq2Seq DL architecture. With further fine tuning performed on our dataset, trained over 10 epochs, it gave a 96% reconstruction accuracy.

The generation of VAE FPs is similar to that of Seq2Seq in the sense that, after learning completes, SMILES are fed into the encoder network and the latent representation is extracted. However, the VAE is parameterized in such a way that it learns a multidimensional normal distribution over latent degrees of freedom which best represents reconstructed SMILES, with

https://dx.doi.org/10.1021/acs.jcim.0c01097
J. Chem. Inf. Model. 2021, 61, 631−640

the VAE FPs being just a point in that space. The Gómez-Bombarelli et al.[16] implementation was utilized. Learning was performed in two stages as described above.

As for the Mol2vec approach, we utilized the work of Jaeger et al.[15] The Mol2vec approach utilizes the Morgan[34] algorithm to generate atom identifiers between radii 0 and 1. Those identifiers ("words") are then ordered into a "sentence" which serves as a basis for the Word2Vec[35] algorithm. In this case, however, the quality of each embedding is judged by how good it performs in a given supervised ML task, whereas in the case of Seq2Seq, it is additionally leveraged by reconstruction accuracy. This approach is technically unsupervised, as it does not require any labeled data. However, it is NLP-embedding[35]-based rather than an encoder−decoder architecture. Initial learned weights were used to generate Mol2Vec embeddings for our dataset.

After we generated various molecular FPs, all of them were considered for the classification and regression tasks. For the classification task, the prediction was done for one FP at a time. For the regression task, on the other hand, we tried multiple combinations when predicting H_2O_2 losses with permutational layers.

2.2. Datasets. For the classification task, we used the dataset from Wei et al.[1] for the classification of 16 reaction types involving alkyl halides and alkenes. For testing, we used 17,280 reactions, whereas for training we used 4320 reactions balanced across reaction types. It should be noted that this dataset is not particularly suitable for multiclass or multilabel classification tasks alone, as class 3 and class 4 are assumed to occur simultaneously. Therefore, each of these two classes was assigned a probability of 0.5, whereas the rest of the reaction types were mutually exclusive and could be represented by one-hot encodings. As a result, one can frame the problem as if each label is independent and treated with the sigmoid activation function and binary cross-entropy loss. However, since we only have uncertainty with classes 3 and 4, softmax can be used, provided that categorical cross-entropy loss is computed over the softmax layer. In other words, the softmax layer with categorical cross-entropy as a loss function is able to learn this dataset somewhat correctly in Keras. It learns somewhat correct, rather than fully correct, predictions due to the fact that categorical-entropy loss in this case cannot be strictly 0. The true and predicted labels belong to both classes 3 and 4 simultaneously. Accuracy in this case has to be modified in order to account for floating-point errors. Note that this modification is described in detail later when discussing Figure 2. On the other hand, treating each output neuron as a sigmoid with binary cross-entropy loss implies that each label is independent, which is not the case for other reaction types. Interestingly, such a problem was previously discussed in the YOLO9000 paper.[36] Experiments showed that the softmax layer approach was improved when computing both original[1] and modified accuracy. It should also be noted that reactions with mixed labels 3 and 4 constitute roughly 14% of the dataset used in this paper.

For the regression task, the inputs to our model contained our proprietary dataset collected over many years with the following properties: (1) number and chemical formula of ingredients mixed (expressed as SMILES); (2) weight concentration % w/w of each ingredient; (3) initial pH level of the resulting solution before incubation; and, (4) initial concentration of H_2O_2.

The significance of the H_2O_2 concentration loss lies in the fact that it directly affects antimicrobial properties of the solution. Data was collected over the years from a variety of mixtures and batches. The initial concentration at the time of mixing and final concentration after 1 month of incubation at 54 °C was recorded.

The training objective was to minimize mean squared error (MSE) between predicted H_2O_2 loss from the model and experimental data for H_2O_2 loss (this loss was expressed as a percentage loss from the initial concentration). Please note that although our final product is always H_2O_2, its degradation and antimicrobial/antiviral properties differ. We collected a total of 831 experimental data points, with 664 used for training and 167 for testing. This corresponds to an 80−20% train/test dataset split. Among the 831 data points, there were only 262 unique compounds. Initial data pruning was performed by removing outliers. As a result, we limited ourselves to formulations with a maximum of 20 ingredients mixed in (actual data had a maximum of 29 mixtures for some formulations). In the case where it was less than 20, we used zero padding. The problem of imbalanced datasets is a well-known problem in classification tasks with categorical variables in the ML field. However, only recently have imbalanced datasets gained attention[37] for regression tasks. As a result, proper balancing of a regression dataset was not considered except for the initial data pruning process.

2.3. Model. For the baseline classification task, we utilized the original Wei et al.[1] model with one hidden layer of 100 neurons, an output softmax layer, and the original learning rates. Training was done to minimize categorical cross-entropy. The Adam optimizer was used, with standard settings (normal initialization) from Segler et al.[9] All activation functions were ReLU[38] unless specified otherwise. For our classification and regression tasks, the permutational layers of Guttenberg et al.[23] were used as the main building blocks. Max-pooling was used when aggregating over pairwise interactions within all permutational layers considered. For our classification task specifically, we used a permutational layer with 4 dense layers of 100 neurons each. A Max-pooling layer was used to map permutationally equivariant layers into permutationally invariant output. Learning rate was set to 2.8×10^{-3}. All runs for the classification task were run 10 times over each permutation and average accuracies were collected.

For our regression task, the last layer was also ReLU as it was physically impossible for our H_2O_2 losses to be negative. Two permutationally equivariant layers were used: the first consisting of 4 ReLU (300 neurons each) and the second consisting of a single linear layer of 257 neurons followed by a pooling (average, maximum). The model was trained to minimize the MSE for H_2O_2 prediction loss. Some of the models used are shown in the Supporting Information. To prevent overfitting for any task, we used early stopping. For the regression task, we have additionally used a dropout[39] layer with a dropout rate of 0.5. In terms of batch sizes, we used batch sizes of 100 and 10 for the classification and regression tasks, respectively. Learning rate was set to 10^{-5} in the regression task.

3. RESULTS AND DISCUSSION

3.1. Classification Task with Permutational Layer. In the first step, we tried to reproduce the work of Wei et al.[1] using their compiled dataset and our proposed new DL-NN with novel permutational layers. Figure 1 shows both

https://dx.doi.org/10.1021/acs.jcim.0c01097
J. Chem. Inf. Model. 2021, 61, 631−640

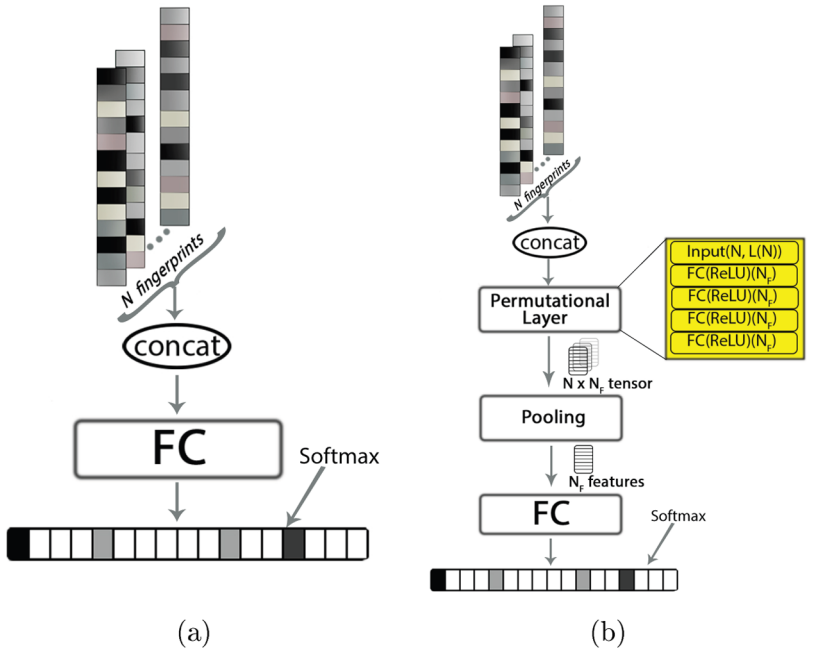


Figure 1. (a) Original Wei's architecture (b) proposed NN architecture (note that FC refers to a fully connected layer).

architectures. It can be seen from Figure 3a that there is uncertainty at the concatenation level where, for given reactants and reagents (3 in our case), there are 6 possible ways (permutations) on how reactants can be concatenated. On the other hand, with the case in Figure 3b, concatenation order is irrelevant and permutational invariance is handled at the permutational layer and the subsequent pooling layer.

In our work, we utilize the permutational layer of Guttenberg et al.,[23] where the permutational layer consists of several dense layers with parameter sharing. Each layer takes all possible pairwise combinations for each input element and produces a pooled output for each input paired with all of the others from the input. For intermediate pooling, we used max pooling as it was suggested to obtain improved accuracy.[23] As a result, after each permutational layer, one has an output tensor of (N, N_f) shape where N_f refers to the number of features and N is the number of input elements in the set as shown in Figure 3b. Subsequently, one needs to apply a final pooling layer so that we obtain full permutational invariant symmetry.

In our case, we tried maximum and average pooling functions, where summation was skipped due to its poor performance. It was mentioned previously[23] that such an approach has a striking resemblance to convolutional layers where long-range correlations can be learned across the entire depth of the network, as every single layer with a locally receptive field is only able to observe a close neighborhood of the area of interest. By introducing additional layers, information eventually percolates through previous layers, thus going from pixels to more meaningful abstractions (long-range correlations) such as object shapes.

Furthermore, in our case, we set $N_f = 100$. As mentioned previously, reactions belonging to classes 3 and 4 occur simultaneously; as a result, this is not strictly suitable for softmax classification. To handle this, we modified the original accuracy based on an argmax approach and introduced $\epsilon(\alpha)$ accuracy.

The original argmax approach returns a label with the maximum value from the last layer (softmax function in our case) as demonstrated in the second row of Figure 2. However, with 2 simultaneous classes, one has to consider more than just

[0, 0, 0, **0.500**, **0.5**, 0 , ... 0]	Ground Truth
[0, 0, 0, 0.493, **0.5** , 0.07, ... 0]	ArgMax
[0, 0, 0, **0.493**, **0.5** , 0.07, ... 0]	$\mathfrak{E}(0.1)$ – Accuracy

Figure 2. Demonstration of $\epsilon(\alpha)$ accuracy.

one possible label. Modified, the $\epsilon(\alpha)$ accuracy determines how far off from the value obtained by argmax is compared to the ground truth (considered an "argmax") and returns labels within α difference. This is what $\epsilon(\alpha)$ is doing, as shown in the third row of Figure 2. As shown for a given datapoint, it would give 100% accuracy. Basically, we are counting simultaneous occurrences of labels 3 and 4, rather than just giving 100% accuracy on either label 3 or 4 (argmax).

Results for both prediction accuracies, original and $\epsilon(\alpha)$, are shown in Figure 3. Note that the original accuracy limit approaching 0.86 is due to the fact that the original implementation could not distinguish between reactions belonging to classes 3 and 4, a portion which was 14% of the dataset. We used NFPs for the original NN and Seq2Seq/Mol2Vec/Morgan FPs with both the original and proposed NN architectures. Please note that in the case of NFPs, we extracted them after the model was originally trained on Wei's classification task. That is, we did not transfer the whole graph convolutional network (GCN) associated with NFPs to our model. All results were averaged over 10 runs for each permutation. One can see that accuracy is consistently higher for both the original and $\epsilon(0.1)$ accuracies in the case of permutational layers.

Confusion matrices for Morgan, Seq2Seq (no perm.), Seq2Seq (perm.), and NFP (see the Supporting Information) methods were generated. Morgan and NFP implementations were adapted from the code of Wei et al.[1] NFPs[27] are associated with the GCN as mentioned above and as a result are learnable from a particular dataset. For Morgan FPs, we

https://dx.doi.org/10.1021/acs.jcim.0c01097
J. Chem. Inf. Model. 2021, 61, 631−640

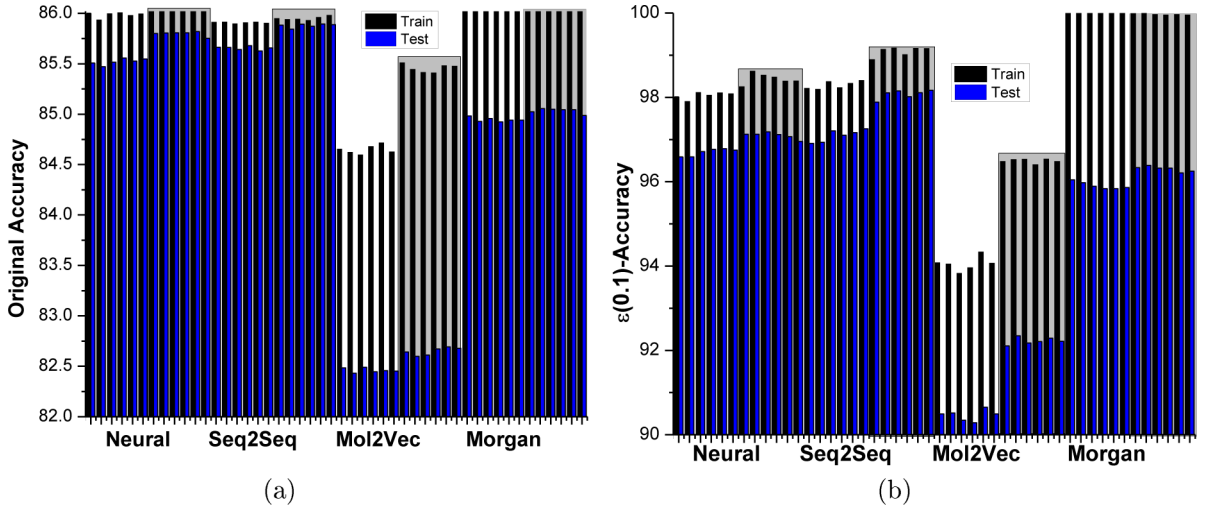


Figure 3. (a) Original with neural, Mol2Vec, Morgan, and Seq2Seq (no perm.) plain concatenation and neural, Mol2Vec, Morgan, Seq2Seq (perm.) (b) $\epsilon(0.1)$ accuracies for classification tasks where x-axis signifies each permutation. Gray shading signifies permutationally invariant architecture. Results were averaged over 10 runs for each permutation.

used a length of 768, whereas for NFP, a length of 156 was used. With the introduction of $\epsilon(\alpha)$ accuracy, confusion in reaction types 3 and 4 nearly disappeared. Additionally, the confusion matrices computed for Morgan and Seq2Seq (no perm.) FPs showed strong results; however, there is still room for improvement as demonstrated in Figure 4e,f with the introduction of permutational layers.

3.2. Regression Task with Permutationally Invariant Layers. For this task, we used our own generated dataset based on real experiments. The main objective was to minimize hydrogen peroxide loss while maintaining its antibacterial properties through the mixing variety of chemicals. Figure 5 shows the distribution of the number of chemical formulations mixed along with H_2O_2 losses for training and test data sets. In our computational experiments, we disregard all data with more than 20 reactants/reagents mixed. Additionally, one can see that the test set is representative of the training set as it has similar statistical properties. H_2O_2 loss values were taken after 1 month of incubation and recorded at room temperature.

Along with FPs, initial concentrations of mixed compounds, pH level of solution before incubation, and initial H_2O_2 concentration are used as inputs to the DL architecture, as shown in Figure 6. We used multiple FPs depicted as FP_1, FP_2, and structural FPs. Structural FP is a permutationally invariant type and, as such, does not require any permutational layers. Concatenation of FPs (FP_1, FP_2) does require explicitly imposed permutational symmetry. Note that we used multiple modalities expressed as FP_1 and FP_2 coming from either Seq2Seq, VAE, or Mol2Vec models. Interestingly, the multiple modalities approach is well known for audio and video data.[40−44] To our best knowledge, this has only been applied once previously[45] to chemistry-related ML research. Our approach has similarities with a chemical heteroencoder[46] in the sense that both approaches utilize multiple modalities as input. However, the goal of a heteroencoder is to build an autoencoder from one modality to another in a sequential manner. Whereas our approach takes multiple modalities at once. One possible drawback of our approach is that all FPs are built upon SMILES representation.

Figure 6 also highlights two baseline models. In baseline 1, only structural FPs (no concentration data is being considered) is used. In baseline 2, the structural FP is concatenated with an output of the permutational layer in which concentrations are being fed into. Additionally, we have considered baseline 3, not shown in Figure 6, where Seq2Seq FPs were concatenated with concentrations resulting in a total of 21 × 256 lengths and are fed to FC(2048) prior being propagated to FC(1024) layer, as per Figure 6. To make comparisons more complete, we also introduced deep layer networks using the best achieving Seq2Seq FP without permutational invariance, along with pooling in baseline 4 and baseline 5. All DL models are shown in the Supporting Information.

A number of combinations with various FPs have been attempted, and the results are shown in Table 1. A major contribution toward root MSE (RMSE) reduction comes from introducing the permutational layer that forces permutational symmetry on the dataset. Whereas the multiple modality approach gives notable benefits only if there is relatively poor latent data representation. This is the case for Mol2Vec FPs. However, for a given information-rich latent representation such as Seq2Seq, one does not benefit by introducing other modalities. Overall, average pooling seems to give slightly better R and RMSE values.

As mentioned previously, summation pooling was not considered due to its poor performance. It can be seen that Seq2Seq alone is performing much better than the other modalities. We also tried additional permutational layers as well as additional FC(ReLU) layers inside each layer. However, the performance seems to degrade as we introduce more layers (>10). The best RMSE and R achieved were with 2 permutational layers: the first one consisting of 10 × FC(ReLU) (300) layers rather than 4 and the second layer being the same.

Results of R for 2 baselines (1, 2) along with the best model are shown in Figure 7. As demonstrated, enforcing permutational invariance by introducing permutational layers significantly improves generalization properties of the model. It should be noted that, in our model, we included points beyond the 1.5 × interquartile range by attempting to capture larger

https://dx.doi.org/10.1021/acs.jcim.0c01097
J. Chem. Inf. Model. 2021, 61, 631−640

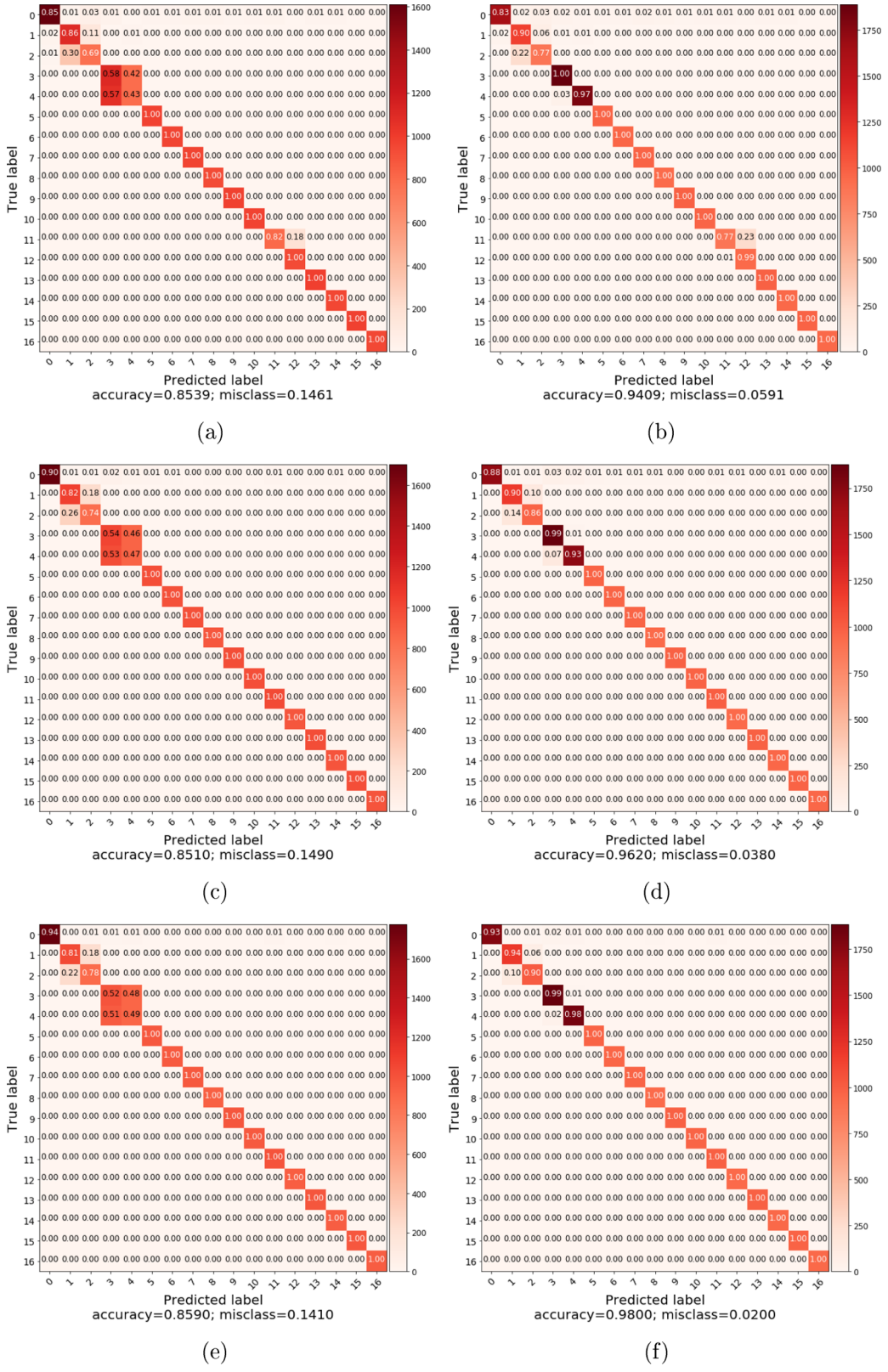


Figure 4. Confusion matrices for test data (a,b) Morgan FPs, (c,d) Seq2Seq (no perm.), and (e,f) Seq2Seq (perm.) with original and $\epsilon(0.1)$ accuracies. Predicted reaction type (label) is on horizontal axis, whereas true reaction type is on vertical axis. Numbers in matrices indicate the ratio of correctly predicted labels, whereas the vertical bar coloring scheme corresponds to the absolute number of reactions belonging to each reaction type.

H_2O_2 losses, as this is crucial for our model to recognize the mix of ingredients yielding those predictions. The inset shows points within the 1.5 × interquantile region; thus, statistically speaking, it is more representative of the dataset. Based on these points alone, one would be able to achieve higher accuracy.

https://dx.doi.org/10.1021/acs.jcim.0c01097
J. Chem. Inf. Model. 2021, 61, 631−640

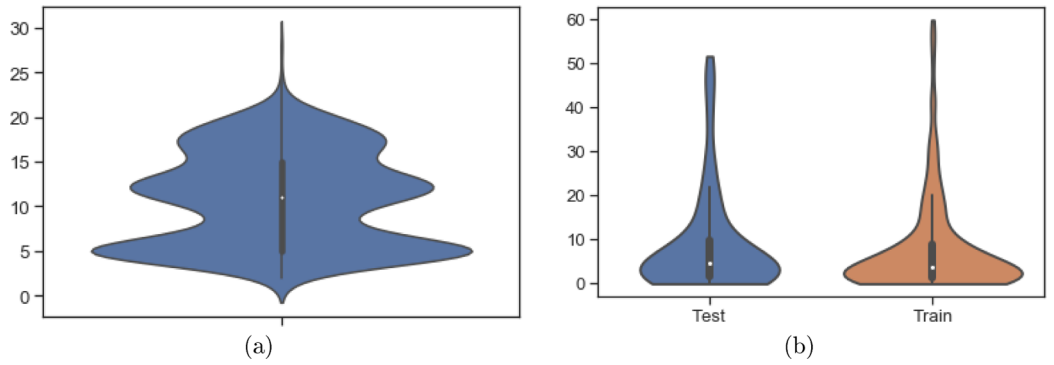


Figure 5. Violin-plots for (a) number of chemical formulations mixed and (b) H_2O_2 losses for training and test datasets.

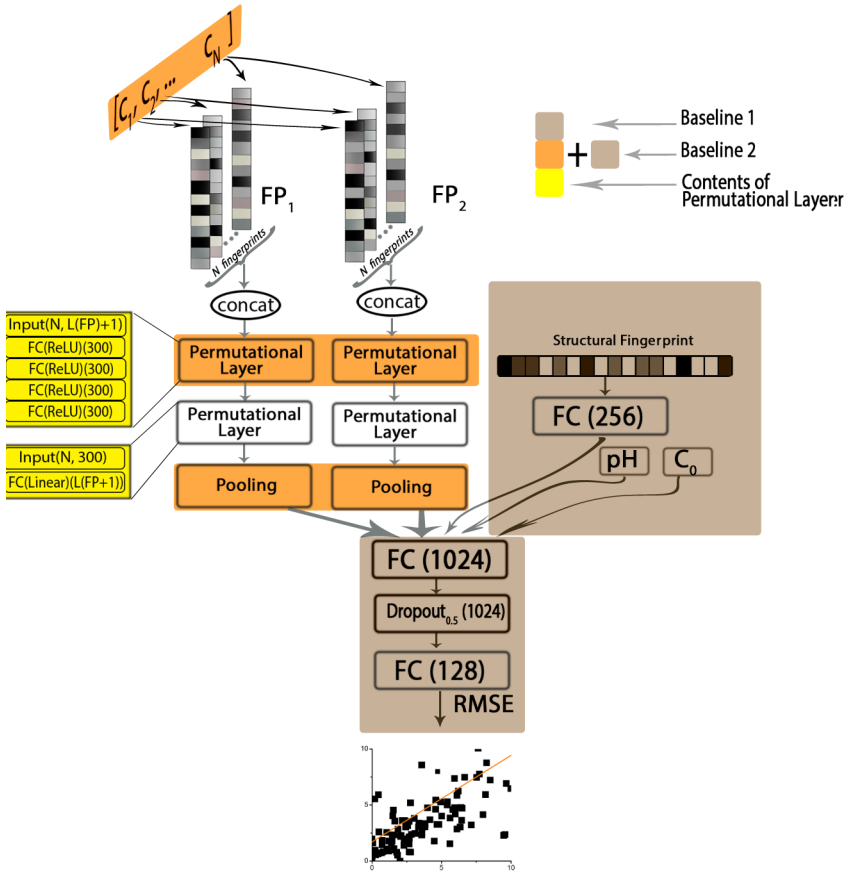


Figure 6. Proposed DL NN architecture used with multiple modalities.

4. CONCLUSIONS

In this paper, we demonstrated that permutational invariance plays a crucial role in molecular fingerprinting methods when applied to ML research where the concatenation of more than one chemical ingredient takes place, such as reactions. More specifically, we show that it is suitable for both classification and regression tasks and serves as an alternative to dataset augmentation whereby permuting input components, one eliminates ordering bias. It has a tremendous advantage when the number of components is modestly large, as permutation alone is not feasible in this case. It is also an alternative to the originally designed permutational invariant FPs such as Daylight structural/difference FPs.

In our case, we are able to have more than one FP (modality) simultaneously taken from other chemical representations to be concatenated in a permutationally invariant manner. Results show that in the case of originally information-rich representation, one does not gain a lot. This is the opposite for the case of information-poor representation, where one benefits from being able to learn latent space simultaneously from multiple representations. It also is interesting to note that given the information-rich latent representation and by introducing more modalities within the model, the performance degrades only slightly. Another advantage is that we are able to introduce more parameters such as partial charge, dipole moment, bioactivity, and others and apply them to each FP, which would not be possible using fingerprinting methods such as Daylight (unless specifically designed).

This paper presents a step toward analyzing and predicting the properties of chemical reactions/mixtures in a new and

https://dx.doi.org/10.1021/acs.jcim.0c01097
J. Chem. Inf. Model. 2021, 61, 631−640

Table 1. RMSE and Pearson Correlation Coefficients (R) with Various Approaches[a]

FPs	pooling	R (test)	RMSE (test)	R (train)	RMSE (train)
baseline 1	N/A	0.69	8.59	0.87	5.16
baseline 2	max	0.70	8.46	0.88	5.06
baseline 2	ave.	0.71	8.38	0.88	4.99
baseline 3	N/A	0.62	9.40	0.90	4.68
lightgray baseline 4	N/A	0.61	9.54	0.93	3.9
baseline 5	max	0.65	8.91	0.78	6.72
Mol2Vec	max	0.66	9.07	0.92	4.32
Mol2Vec	ave.	0.72	8.36	0.96	2.85
Mol2Vec + Fp	max	0.72	8.36	0.96	2.93
Mol2Vec + Fp	ave.	0.73	8.20	0.97	2.77
Seq2Seq	max	0.77	7.47	0.98	2.12
lightgray Seq2Seq	ave.	0.78	7.45	0.97	2.42
Seq2Seq + Fp	max	0.77	7.57	0.98	2.26
Seq2Seq + Fp	ave.	0.78	7.43	0.98	2.31
VAE	max	0.73	8.12	0.97	2.40
VAE	ave.	0.72	8.42	0.97	2.57
VAE + Fp	max	0.75	7.93	0.98	2.29
VAE + Fp	ave.	0.76	7.74	0.97	2.72
Seq2Seq + Mol2Vec + Fp	max	0.77	7.61	0.97	2.55
Seq2Seq + Mol2Vec + Fp	ave.	0.77	7.53	0.97	2.48
Seq2Seq + Mol2Vec + Fp*	ave.	0.80	6.93	0.98	2.25

[a]* indicates best achieved RMSE and R, with 10 FC(ReLU) (300) layers rather than 4 as indicated in Figure 6. Please note that the highlighted rows indicate the best performing Seq2Seq with permutational layers and the closest corresponding model without permutational layers.

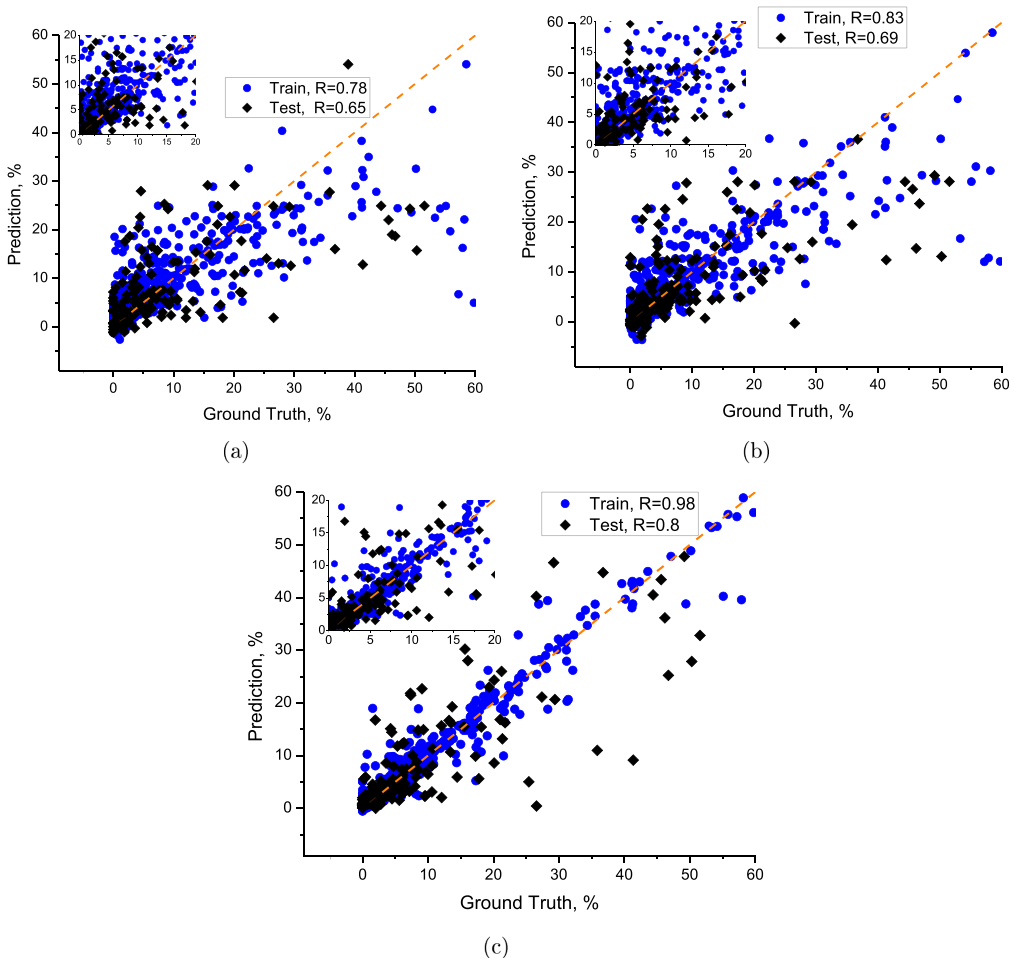


Figure 7. Prediction vs ground truth H_2O_2 losses for (a) baseline 1, (b) baseline 2, and (c) Seq2Seq + Mol2Vec + Fp* methods. Dashed orange line corresponds to perfect correlation ($y = x$).

https://dx.doi.org/10.1021/acs.jcim.0c01097
J. Chem. Inf. Model. 2021, 61, 631−640

systematic way. The model eliminates permutational bias and introduces multiple representations into the DL architecture. While the majority of these algorithms are established in the ML community, it is the first time that they have been effectively combined and utilized in the field of chemistry research for predicting reactions and compound mixtures.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.0c01097.

Wei's Original model with neural fingerprints; fingerprints with permutational layers used in modified classification task; fingerprints with permutational layers used in modified classification task; Baseline1 model; and confusion matrices using neural fingerprints on test data Code and data are available at: https://github.com/phquanta/DeepPermInvFp.git (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Andrei Buin − *College of Engineering and Physical Sciences, University of Guelph, Guelph, Ontario N1G 2W1, Canada*; Email: phquanta@gmail.com

S. Andrew Gadsden − *College of Engineering and Physical Sciences, University of Guelph, Guelph, Ontario N1G 2W1, Canada*; orcid.org/0000-0003-3749-0878; Email: gadsden@uoguelph.ca

Authors

Hung Yi Chiang − *College of Engineering and Physical Sciences, University of Guelph, Guelph, Ontario N1G 2W1, Canada*

Faraz A. Alderson − *College of Engineering and Physical Sciences, University of Guelph, Guelph, Ontario N1G 2W1, Canada*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.0c01097

Notes

The authors declare no competing financial interest.

■ REFERENCES

(1) Wei, J. N.; Duvenaud, D.; Aspuru-Guzik, A. Neural Networks for the Prediction of Organic Chemistry Reactions. *ACS Cent. Sci.* **2016**, *2*, 725−732.

(2) Coley, C. W.; Barzilay, R.; Jaakkola, T. S.; Green, W. H.; Jensen, K. F. Prediction of Organic Reaction Outcomes Using Machine Learning. *ACS Cent. Sci.* **2017**, *3*, 434−443.

(3) Arús-Pous, J.; Johansson, S. V.; Prykhodko, O.; Bjerrum, E. J.; Tyrchan, C.; Reymond, J.-L.; Chen, H.; Engkvist, O. Randomized SMILES Strings Improve the Quality of Molecular Generative Models. *J. Cheminf.* **2019**, *11*, 71.

(4) Schneider, N.; Lowe, D. M.; Sayle, R. A.; Landrum, G. A. Development of a Novel Fingerprint for Chemical Reactions and its Application to Large-Scale Reaction Classification and Similarity. *J. Chem. Inf. Model.* **2015**, *55*, 39−53.

(5) Schwaller, P.; Gaudin, T.; Lányi, D.; Bekas, C.; Laino, T. "Found in Translation": Predicting Outcomes of Complex Organic Chemistry Reactions Using Neural Sequence-to-Sequence Models. *Chem. Sci.* **2018**, *9*, 6091−6098.

(6) Xu, Z.; Wang, S.; Zhu, F.; Huang, J. Seq2seq Fingerprint: An Unsupervised Deep Molecular Embedding for Drug Discovery. *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 2017; pp 285−294.

(7) Schreck, J. S.; Coley, C. W.; Bishop, K. J. M. Learning Retrosynthetic Planning through Simulated Experience. *ACS Cent. Sci.* **2019**, *5*, 970−981.

(8) Gupta, A.; Müller, A. T.; Huisman, B. J. H.; Fuchs, J. A.; Schneider, P.; Schneider, G. Generative Recurrent Networks for De Novo Drug Design. *Mol. Inf.* **2018**, *37*, 1700111.

(9) Segler, M. H. S.; Kogej, T.; Tyrchan, C.; Waller, M. P. Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS Cent. Sci.* **2018**, *4*, 120−131.

(10) Harel, S.; Radinsky, K. Prototype-Based Compound Discovery Using Deep Generative Models. *Mol. Pharm.* **2018**, *15*, 4406−4416.

(11) Cereto-Massagué, A.; Ojeda, M. J.; Valls, C.; Mulero, M.; Garcia-Vallvé, S.; Pujadas, G. Molecular Fingerprint Similarity Search in Virtual Screening. *Methods* **2015**, *71*, 58−63.

(12) Morgan, H. L. The Generation of a Unique Machine Description for Chemical Structures—A Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* **1965**, *5*, 107−113.

(13) Cherkasov, A.; Muratov, E. N.; Fourches, D.; Varnek, A.; Baskin, I. I.; Cronin, M.; Dearden, J.; Gramatica, P.; Martin, Y. C.; Todeschini, R.; Consonni, V.; Kuz'min, V. E.; Cramer, R.; Benigni, R.; et al. QSAR Modeling: Where Have You Been? Where Are You Going To? *J. Med. Chem.* **2014**, *57*, 4977−5010.

(14) Weininger, D. SMILES, A Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31−36.

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

(16) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.* **2018**, *4*, 268−276.

(17) De Cao, N.; Kipf, T. MolGAN: An Implicit Generative Model for Small Molecular Graphs. **2018**, arXiv:1805.11973.

(18) Sandfort, F.; Strieth-Kalthoff, F.; Kühnemund, M.; Beecks, C.; Glorius, F. A Structure-Based Platform for Predicting Chemical Reactivity. *Chem* **2020**, *6*, 1379−1390.

(19) Zaheer, M.; Kottur, S.; Ravanbakhsh, S.; Poczos, B.; Salakhutdinov, R. R.; Smola, A. J. Deep Sets. *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2017; pp 3391−3401.

(20) Murphy, R. L.; Srinivasan, B.; Rao, V.; Ribeiro, B. Janossy Pooling: Learning Deep Permutation-Invariant Functions for Variable-Size Inputs. **2018**, arXiv:1811.01900. arXiv preprint.

(21) Sannai, A.; Takai, Y.; Cordonnier, M. Universal Approximations of Permutation Invariant/Equivariant Functions by Deep Neural Networks. **2019**, arXiv:1903.01939. arXiv preprint.

(22) Ravanbakhsh, S.; Schneider, J.; Poczos, B. Deep Learning with Sets and Point Clouds. **2016**, arXiv:1611.04500. arXiv preprint.

(23) Guttenberg, N.; Virgo, N.; Witkowski, O.; Aoki, H.; Kanai, R. Permutation-Equivariant Neural Networks Applied to Dynamics Prediction. **2016**, arXiv:1612.04530. arXiv preprint.

(24) Schwaller, P.; Probst, D.; Vaucher, A. C.; Nair, V. H.; Kreutter, D.; Laino, T.; Reymond, J.-L. Mapping the Space of Chemical Reactions using Attention-Based Neural Networks. **2020**, arXiv:2012.06051. arXiv preprint.

(25) *Daylight*; Daylight Chemical Information Systems, Inc.: 27401 Los Altos, Suite #370, Mission Viejo, CA 92691, USA.

(26) Polishchuk, P.; Madzhidov, T.; Gimadiev, T.; Bodrov, A.; Nugmanov, R.; Varnek, A. Structure−Reactivity Modeling Using Mixture-Based Representation of Chemical Reactions. *J. Comput.-Aided Mol. Des.* **2017**, *31*, 829−839.

(27) Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional Networks on Graphs for Learning Molecular Fingerprints. *Advances in Neural*

Information Processing Systems; Curran Associates, Inc., 2015; pp 2224−2232.
(28) Landrum, G. RDKit: Open-Source Cheminformatics. http://www.rdkit.org (accessed June 19, 2020).
(29) Chollet, F.; et al. Keras. 2015, https://github.com/fchollet/keras (accessed June 20, 2020).
(30) Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015, https://www.tensorflow.org/ (accessed June 07, 2020). Software available from tensorflow.org.
(31) Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. **2014**, arXiv:1409.0473. arXiv preprint.
(32) Arús-Pous, J.; Johansson, S.; Prykhodko, O.; Bjerrum, E. J.; Tyrchan, C.; Reymond, J.-L.; Chen, H.; Engkvist, O. Improving Deep Generative Models with Randomized SMILES. *Artificial Neural Networks Machine Learning—ICANN 2019: Workshop Special Sessions*: Cham, 2019; pp 747−751.
(33) Bjerrum, E. J. Smiles Enumeration as Data Augmentation for Neural Network Modeling of Molecules. **2017**, arXiv:1703.07076. arXiv preprint.
(34) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.
(35) Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. **2013**, arXiv:1301.3781. arXiv preprint.
(36) Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017; pp 7263−7271.
(37) Branco, P.; Torgo, L.; Ribeiro, R. P. SMOGN: a Pre-processing Approach for Imbalanced Regression. *Proceedings of the First International Workshop on Learning with Imbalanced Domains: Theory and Applications*; ECML-PKDD: Skopje, Macedonia, 2017; pp 36−50.
(38) Nair, V.; Hinton, G. E. Rectified Linear Units Improve Restricted Boltzmann Machines. *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010; pp 807−814.
(39) Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929−1958.
(40) Zhang, X.; Fu, Y.; Zang, A.; Sigal, L.; Agam, G. Learning Classifiers from Synthetic Data Using a Multichannel Autoencoder. **2015**, arXiv:1503.03163. arXiv preprint.
(41) Liu, Y.; Feng, X.; Zhou, Z. Multimodal Video Classification with Stacked Contractive Autoencoders. *Signal Process.* **2016**, *120*, 761−766.
(42) Guo, Q.; Jia, J.; Shen, G.; Zhang, L.; Cai, L.; Yi, Z. Learning Robust Uniform Features for Cross-Media Social Data by Using Cross Autoencoders. *Knowl.-Based Syst.* **2016**, *102*, 64−75.
(43) Feng, F.; Wang, X.; Li, R. Cross-Modal Retrieval with Correspondence Autoencoder. Proceedings of the 22nd ACM International Conference on Multimedia, 2014; pp 7−16.
(44) Ngiam, J.; Khosla, A.; Kim, M.; Nam, J.; Lee, H.; Ng, A. Y. Multimodal Deep Learning. *International Conference on Machine Learning*, 2011.
(45) Goh, G. B.; Sakloth, K.; Siegel, C.; Vishnu, A.; Pfaendtner, J. Multimodal Deep Neural Networks Using Both Engineered and Learned Representations for Biodegradability Prediction. **2018**, arXiv:1808.04456. arXiv preprint.
(46) Bjerrum, E.; Sattarov, B. Improving Chemical Autoencoder Latent Space and Molecular De Novo Generation Diversity with Heteroencoders. *Biomol* **2018**, *8*, 131.

https://dx.doi.org/10.1021/acs.jcim.0c01097
J. Chem. Inf. Model. 2021, 61, 631−640