Chemical Research in Toxicology

pubs.acs.org/crt

Article

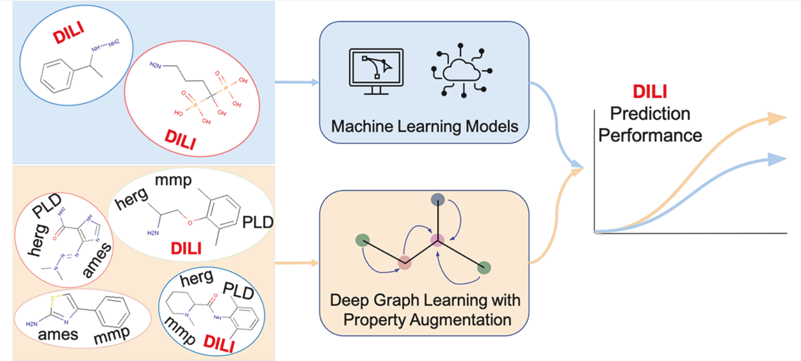
Deep Graph Learning with Property Augmentation for Predicting Drug-Induced Liver Injury

Hehuan Ma, Weizhi An, Yuhong Wang, Hongmao Sun, Ruili Huang, and Junzhou Huang*

Cite This: *Chem. Res. Toxicol.* 2021, 34, 495−506

Read Online

ACCESS | Metrics & More | Article Recommendations | SI Supporting Information



ABSTRACT: Drug-induced liver injury (DILI) is a crucial factor in determining the qualification of potential drugs. However, the DILI property is excessively difficult to obtain due to the complex testing process. Consequently, an *in silico* screening in the early stage of drug discovery would help to reduce the total development cost by filtering those drug candidates with a high risk to cause DILI. To serve the screening goal, we apply several computational techniques to predict the DILI property, including traditional machine learning methods and graph-based deep learning techniques. While deep learning models require large training data to tune huge model parameters, the DILI data set only contains a few hundred annotated molecules. To alleviate the data scarcity problem, we propose a property augmentation strategy to include massive training data with other property information. Extensive experiments demonstrate that our proposed method significantly outperforms all existing baselines on the DILI data set by obtaining a 81.4% accuracy using cross-validation with random splitting, 78.7% using leave-one-out cross-validation, and 76.5% using cross-validation with scaffold splitting.

■ INTRODUCTION

Drug discovery has been a critical research area for years. The development process of new drugs is extremely time-consuming and resource costly since it usually requires a series of complicated *in vitro* and *in vivo* experiments.[1−3] One major challenge is to identify the safety of the potential drug candidates, for example, filtering the drugs that may cause human toxicity. Drug-induced liver injury (DILI) is one of the most fundamental toxicity concerns that is undesirable and unpredictable. Research indicates that traditional hepatotoxicity testings on animal models may have distinct outcomes from humans.[4−6] Since animal or human model testings are usually conducted in the late stage of drug development, the withdrawal or termination of such disqualified drug candidates would sacrifice lots of previous efforts. Therefore, a precise and accurate model to better predict DILI in the early stages would be a promising approach to facilitate the development progress.

Human toxicity data are extremely hard to collect, since *in vivo* and *in vitro* toxicological studies cannot provide adequate assessment when the drug candidates are applied on humans.[4−7] Several labeling schemes[8−11] have been developed to annotate the DILI label for certain drugs to provide predictive models with labeled data. Sakatis et al. is based on the Physician's Desk Reference, while others[8−10] are from case reports and literature. Although labeled DILI data sets are available to the public, such data sets only contain 100 or 200 drugs, and what is worse, the labeling standards are inconsistent. To tackle this problem, the Food and Drug Administration (FDA) has developed an annotation scheme to label the DILI risk of 1036 FDA-approved drugs and announced the DILIrank[12] data set in 2016. The previous version of DILIrank annotates the drugs with Most-DILI concern, Less-DILI concern, and No-DILI concern, based on regulatory professionals assessment.[13] The new scheme establishes a more detailed verification process dividing the

Special Issue: Computational Toxicology

Received: August 1, 2020
Published: December 21, 2020

ACS Publications

https://dx.doi.org/10.1021/acs.chemrestox.0c00322

drugs into four categories: Most-DILI concern, Less-DILI concer, No-DILI concern, and Ambiguous DILI concern.[12] DILIrank is the most widely used data sets to develop predictive models of DILI and has been used in various studies.[14−17] Lately, the FDA further augments DILIrank to DILIst[4] with another four literature data sets by applying concordance analysis across these five datsets. To date, DILIst is the largest data set with DILI classification, which contains 1279 drugs. These efforts[4,12] provide an invaluable resource for predicting DILI risk.

DILI prediction can be considered as the application of molecular property prediction, which is one of the oldest chem-informatics tasks. Many *in silico* methods have been applied to solve the molecular property prediction problem.[18−21] These approaches generally convert the molecule into a vector representation via different procedures and then through different machine learning models to predict the label information. The vector representation of a molecule is called fingerprints. Traditionally, fingerprints are either manually constructed by experts (hand-crafted biologist-guided fingerprints) or calculated by a fixed hash function (hash-based fingerprints). The former one is designed by specialists based on biological experiments and chemical knowledge. Specific substructures of the compounds are considered as functional groups, and their corresponding local features are determined based on their properties revealed during experiments or different states.[18,19] For example, CC(OH)-CC appears to have a solubility relevant characteristic, thus it has been isolated as local features to produce fingerprints on solubility-related tasks. Hash-based fingerprints such as circular fingerprints employ a fixed hash function to extract each layer's feature of a molecule based on the concatenated features of the neighborhood in the previous layer.[20] This type of the fingerprint is non-invertible, so there is no way to check back and modify the quality of the fingerprints if the hash function cannot capture enough information, which might lead to poor performance in further predictive tasks. To tackle this problem, Le et al. recently proposed a reverse-engineering method to reconstruct the molecular structure from hash-based fingerprints such as ECFP.[22]

With the rapid increase of deep learning techniques, recent studies tend to address molecular property prediction with such novel models. One promising research interest is considering a molecule as a graph, since the atoms of the molecules can be referred as the vertexes, and the bonds between atoms as the edges. Neural fingerprints[21] are the first attempt to learn molecular vector representation based on its graph structure. The difference between neural fingerprints and hash-based fingerprints is the replacement of the hash function. Neural fingerprints apply a nonlinear activated densely connected layer to generate the fingerprints. Many other graph-based deep learning models can also be applied to represent a molecule by embedding the graph features to a continuous vector.[23,24] Within them, the Message Passing Neural Networks (MPNN)[25,26] have achieved notable prediction performance. MPNN models recursively update the atom or bond features by aggregating message/information from its adjacent atoms or bonds, then employ a readout function to pool all updated features of atoms to deliver the global representation of the molecule. However, these methods only focus on one single view of the graph topology, either atom central or bond central. Taking Figure 1 as an example, the left graph is the atom-oriented structure of caffeine, and the right one is its bond-oriented representation. It is observed that both atom and bond features should be taken into account when embedding a molecule graph, e.g., the double bond within the benzene N═C is distinct from bond C═O, atoms N and C are notably different. Inspired by this insight, we propose a fresh perspective of viewing the graph from two aspects in our recent work MV-GNN^{cross},[27] which involves both atom messages and bond messages. The MV-GNN^{cross} model takes the molecular SMILES as input and uses RDKit[28] to extract the graph structure and the local features associated with each atom and bond. A graph encoder network then learns and converts such information into a vector representation of the input molecular SMILES. After that, the vector representation is fed into a prediction network to predict the property label. Our method outperforms current state-of-the-art methods on 11 commonly used moleclar property prediction tasks. Therefore, we employed our graph-based deep learning model on the DILIrank data set to classify the DILI label and achieved superior prediction performance compared with other models including both graph-based deep learning models and traditional fingerprints-based models.

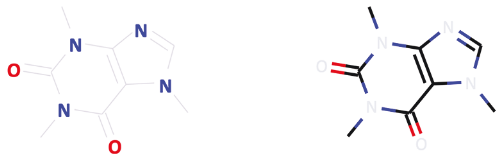


Figure 1. Atom-oriented graph vs bond-oriented graph.

Available labeled DILI drugs are still quite limited for data-hungry deep learning models. In order to get better and more stable prediction performance, research has been done from different aspects. Thakkar et al. developed a new annotation scheme to augment the drug list with DILI risk. Minerali et al. employed different machine learning models on different human toxicity data sets to investigate the corresponding prediction performance. Ancuceanu et al. and Mora et al. propose to obtain better prediction results with ensemble computational models and various molecular descriptors. These attempts have earned certain achievements, but may still be restricted by the available labeled DILI data. To tackle this bottleneck and reinforce the expressive power of deep learning models, we propose a property augmentation strategy to utilize MV-GNN^{cross} models along with more data by taking advantage of other property information. In particular, we create a larger training data set by combining more drugs with other toxic properties, such as phospholipidosis (PLD)[29] which measures the organism-level toxicity of compounds. Since a graph neural network is able to learn molecular vector representation only based on its graph structure and the underlying atom/bond level features, more input data would help generate a more accurate molecular representation. Moreover, for those properties with more available data, deep learning techniques are more likely to obtain better performance. Thus, the correct prediction would help promote the entire training including those properties with only a few samples, such as DILI. In this fashion, we are able to increase the accuracy of DILI to 81.4% using cross-validation with random splitting, 78.7% using leave-one-out cross-validation, and 76.5% using cross-validation with scaffold splitting, which is regarded as a remarkable boost considering the challenges on DILI risk prediction. Detailed methodologies and experimental procedures are described in later sections.

METHODOLOGIES

We take our recent work with the MV-GNN^{cross} model as the backbone to implement proposed property augmentation

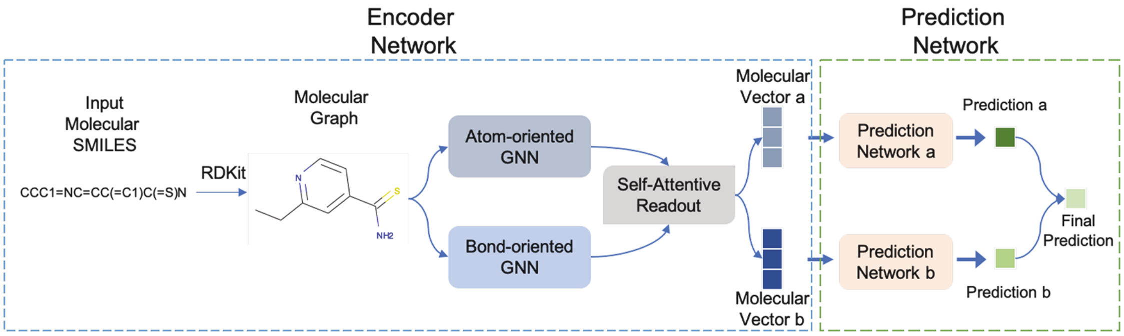


Figure 2. Overview of MV-GNN^{cross} models.

methods, since MV-GNN^{cross} outperforms other baseline models on DILI data sets in extensive experiments. As shown in Figure 2, MV-GNN^{cross} contains two principal parts, the encoder network and the prediction network. The encoder network transforms the input molecular SMILES into a vector representation based on its graph structure, and the prediction network is responsible for classifying the binary label of certain properties, such as DILI. We also employ deep multilabel learning to establish proposed methods while involving more properties information along with DILI.

Molecular Graph Preliminaries. A molecule can be naturally represented as a graph based on its chemical structure, in particular, by taking the atoms as the nodes and the bonds between atoms as the edges. Thus, the molecular graph is denoted as $G_m = (\mathcal{A}, \mathcal{B})$, where $\mathcal{A}$ is a set of the atoms, and $\mathcal{B}$ is a set of the bonds. Based on such a graph structure, the initial features of atoms and bonds are extracted as the learning information, and referred as x_a and y_b. Figure 3 takes ethionamide as an example to illustrate how a molecule converts to its corresponding computational graph.

The initial features selected for each atom and bond follow the same protocol of Yang et al., as shown in Tables 1 and 2.[26] All of the features are one-hot encodings except the atomic mass and are extracted using the RDKit.[28]

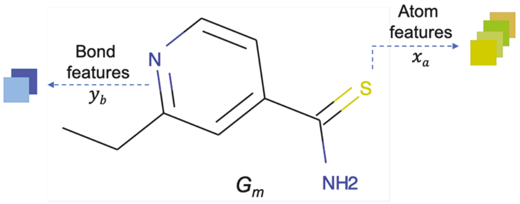


Figure 3. Graph definition of ethionamide. G_m represents the entire graph structure, and x_a and y_b refer to the atom and bond features that associate with each atom and bond, respectively.

Table 1. Atom Features Selection

features	size	descriptions
atom type	100	type of atom (e.g., C, N, O), in the order of atomic number
formal charge	5	integer electronic charge assigned to atom
number of bonds	6	number of bonds the atom is connected
chirality	4	unspecified, tetrahedral CW/CCW, or other.
number of Hs	5	number of bonded hydrogen atoms
atomic mass	1	mass of the atom, divided by 100
aromaticity	1	whether this atom is part of an aromatic system
hybridization	5	sp, sp^2, sp^3, sp^{3}d, or sp^{3}d^2

Table 2. Bond Features Selection

features	size	descriptions
bond type	4	single, double, triple, or aromatic
stereo	6	*E*/*Z*, *cis*/*trans*, any, or none
in ring	1	whether the bond is part of a ring
conjugated	1	whether the bond is conjugated

Encoder Network. Molecules can be observed from two perspectives: One is taking the atoms as the centers and the bonds as the connections,[25] while the other one is to consider bonds as the centers and atoms as connections.[26] Inspired by multiview learning,[30] MV-GNN^{cross} takes advantage of the two perspectives and designs a multiview framework to generate more informative molecular representation. Specifically, the encoder network is constructed by two streams, atom-oriented and bond-oriented, where each contains one graph neural network (GNN). Next, a self-attentive readout mechanism is employed to convert the learned molecular feature matrix to a vector representation.

Atom-Oriented GNN and Bond-Oriented GNN. The atom-oriented GNN learns the molecular representation by aggregating neighbor atoms recursively for several steps, while bond-oriented GNN establishes a similar procedure via a bond-central fashion. The generalized GNN can be defined as

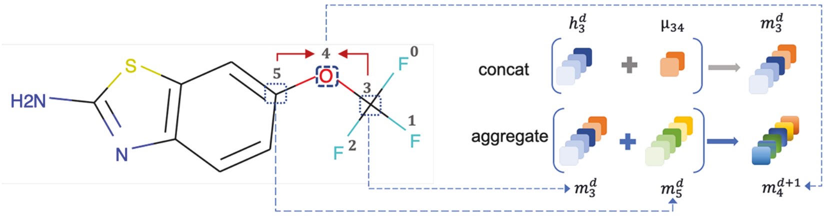
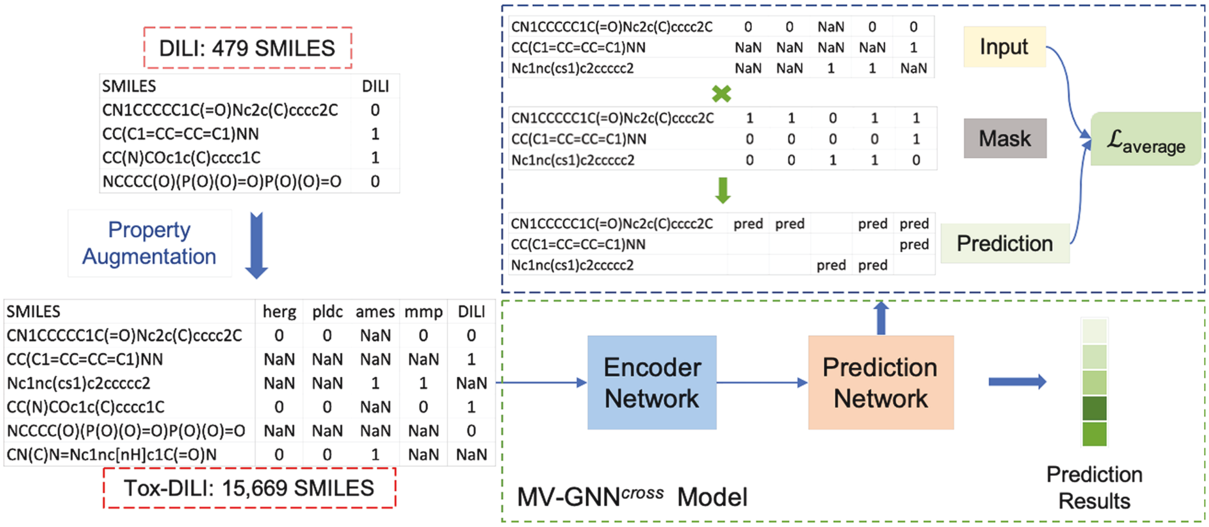


Figure 4. Message passing aggregation phase. Taking atom 4 as an example, atoms 3 and 5 are its neighbors. In the passing process, the message of atoms 3 and 5 from previous passing steps will be aggregated to atom 4. For the message construction, we take atom 3 as an example. The message m_3^d of atom 3 is concatenated by the initial atom features h_3^d of atom 3 as well as the initial bond features μ_{34} of the connected bond 34.



SMILES	DILI
CN1CCCCC1C(=O)Nc2c(C)cccc2C	0
CC(C1=CC=CC=C1)NN	1
CC(N)COc1c(C)cccc1C	1
NCCCC(O)(P(O)(O)=O)P(O)(O)=O	0

SMILES	herg	pldc	ames	mmp	DILI
CN1CCCCC1C(=O)Nc2c(C)cccc2C	0	0	NaN	0	0
CC(C1=CC=CC=C1)NN	NaN	NaN	NaN	NaN	1
Nc1nc(cs1)c2ccccc2	NaN	NaN	1	1	NaN
CC(N)COc1c(C)cccc1C	0	0	NaN	0	1
NCCCC(O)(P(O)(O)=O)P(O)(O)=O	NaN	NaN	NaN	NaN	0
CN(C)N=Nc1nc[nH]c1C(=O)N	0	0	1	NaN	NaN

CN1CCCCC1C(=O)Nc2c(C)cccc2C	0	0	NaN	0	0
CC(C1=CC=CC=C1)NN	NaN	NaN	NaN	NaN	1
Nc1nc(cs1)c2ccccc2	NaN	NaN	1	1	NaN

CN1CCCCC1C(=O)Nc2c(C)cccc2C	1	1	0	1	1
CC(C1=CC=CC=C1)NN	0	0	0	0	1
Nc1nc(cs1)c2ccccc2	0	0	1	1	0

CN1CCCCC1C(=O)Nc2c(C)cccc2C	pred	pred		pred	pred
CC(C1=CC=CC=C1)NN					pred
Nc1nc(cs1)c2ccccc2			pred	pred	

Figure 5. Property augmentation procedure. Original DILI data set is augmented to the Tox-DILI data set. Tox-DILI is then fed into the MV-GNN^{cross} model for prediction. During the training period of the prediction network, a mask scheme is applied to handle the back-propagation of missing labels, and an average loss across all properties is used to restrain the entire training.

$$m_o^{d+1} = \sum_{\eta \in \mathcal{N}(o)} \mathcal{A}_d(h_\eta^d, \mu_{\text{attached}})$$
$$h_o^{d+1} = \mathcal{U}_d(h_o^d, m_o^{d+1}) \quad (1)$$

In eq 1, $\mathcal{A}_d$ and $\mathcal{U}_d$ represent the neighbor aggregation function and state update function, respectively. m_o^{d+1} and h_o^{d+1} are the aggregated message and states vector for entity o at $d + 1$ step, respectively. Entity o can be either atoms or bonds. $\mathcal{N}(o)$ is the neighborhood entity set of entity o, and μ_{attached} is the attached features of entity o during aggregation. In atom-oriented GNN, entity o represents the atoms, and μ_{attached} denotes the features for the connected bonds. Figure 4 illustrates the message passing phase in atom-oriented GNN. The bond-oriented GNN is formed with a similar implementation by considering the bonds as passing centers and atom features as attached. In particular, the entity o represents the bonds, and the corresponding bond messages m_o^{d+1} are constructed by bond states vector h_o^{d+1} and attached atom features μ_{attached}.

Self-Attentive Readout. The outputs of the two GNN models are the learned feature matrices by regarding the molecular graph as atom-oriented and bond-oriented. As demonstrated in Figure 2, in order to obtain the fixed length of molecular vector representation, a readout transformation is needed to eliminate the obstacle of size variance and permutation variance. Other than commonly used mean-pooling or max-pooling, a self-attentive readout is employed here to generate molecular representation associated with different attention weights.[31,32] Formally, take an output of atom-oriented GNN $\mathbf{H}_n$ as an example, the self-attention over atoms is defined as

$$S = \text{softmax}(\mathbf{W}_2 \tanh(\mathbf{W}_1 \mathbf{H}_n)), \quad \xi_n = (S\mathbf{H}_n^{\text{T}}) \quad (2)$$

where n is the number of atoms in the molecule, and $\mathbf{W}_1$ and $\mathbf{W}_2$ are learnable matrices, which are shared between the two streams to enable message circulation during the multiview training process. Thus, two molecular vectors are generated in a multiview manner.

Prediction Network. In MV-GNN^{cross}, we have generated two vectors from the two submodules: atom-oriented GNN and bond-oriented GNN. These two vectors are fed into two prediction networks to make the predictions. Since the two vectors generated via atom-oriented GNN and bond-oriented GNN come from the same input SMILES, the predictions should be the same. Thus, we employ mean squared error loss to restrain the training, called disagreement loss. Formally, we formulate this molecular property prediction loss as follows:

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{pred}} + \lambda \mathcal{L}_{\text{dis}} \quad (3)$$

where $\mathcal{L}_{\text{pred}}$ is the supervised loss for each prediction, and $\mathcal{L}_{\text{dis}}$ is the disagreement loss between two classifiers.

Property Augmentation Learning. The DILI data set only contains a few hundred drugs, which is extremely small for deep learning. In order to take advantage of the expressive power of deep graph learning models such as MV-GNN^{cross}, we demand more information to boost the training. Since DILI is a property of human toxicity, we compare it with four other available human toxicity data sets: herg,[33,34] PLD,[29] ames,[35,36] and mmp.[37,38] We notice there are overlapping molecules between DILI and these four toxicity data sets. We assume that such correlation may help the training of DILI. Hence, we propose to utilize this additional toxicity information to promote the prediction performance of DILI.

Multilabel Training. As shown in Figure 5, the original DILI data set contains only 479 SMILES. We take it with four other toxicity properties (herg, PLD, ames, and mmp) which are provided by the National Institutes of Health (NIH), to form a larger data set. Specifically, we combine these five data sets based on the SMILES representation of the drugs. Thus, a large matrix containing 15,669 data samples is generated, where each row stands for one SMILES, and the five columns are the corresponding property labels. Each SMILES could have one or more property labels, and those properties which are not observed for each SMILES are marked as missing values and are represented as NaN. The constructed Tox-DILI then goes through the MV-GNN^{cross} model to classify the labels. We employ a multilabel training approach to establish the property augmentation learning process. During the training process, all property predictions share the same encoder network and make predictions for each property label individually. Then, the average of all the prediction loss is used to update the neural network parameters. We treat each property as equally important and ignore the prediction for those NaN properties to avoid deviation.

Missing Labels Handling. In order to eliminate the effects of the missing labels during the training period, we need to identify such labels for each SMILES and ignore them during the back-propagation. In our experiments, a mask scheme is implemented as the filter. The mask is a matrix with the exact same size of the input, which is applied in the prediction network. While the prediction is made by the prediction network and the loss is calculated for each data sample, the mask is then multiplied with the loss values. The mask matrix is filled by 0s and 1s, as the corresponding positions with missing labels are recorded as 0 and others as 1. Thus, any weights associated with those missing labels would have no influence on further computation.

Since each SMILES may have multiple binary property labels at the same time, such a task could be regarded as a multiple binary classification problem. Hence, we employ the binary cross entropy (BCE) loss as the prediction loss function and compute the average loss across each property. Suppose the data set contains molecules $\mathcal{M} = \{M_i\}_{i=1}^{K}$, we formulate the final loss processed by the mask as follows:

$$\mathcal{L}_{\text{pred}} = \frac{1}{N \times K} \sum_{n=1}^{N} \sum_{M_i \in \mathcal{M}} (\mathcal{L}_{\text{a}}(y_i, \gamma_{a,M_i}) \times \text{mask} + \mathcal{L}_{\text{b}}(y_i, \gamma_{b,M_i}{}^{*}) \times \text{mask}) \qquad (4)$$

where γ_{a,M_i} and γ_{b,M_i} are the output predictions produced by the two prediction networks, $\mathcal{L}_{\text{a}}$ and $\mathcal{L}_{\text{b}}$ are the corresponding computed loss, y_i is the ground truth label, and N is the total number of properties, which is 5 in our experiments here.

Evaluation Criteria. Since our task is to predict the binary label of DILI by considering Most-DILI-Concern as the positive label and No-DILI-Concern as the negative label, we thoroughly evaluate the performance of each method by calculating the accuracy, sensitivity, specificity, F1-score, Matthews correlation coefficient (MCC), and receiver operating characteristic-area under the curve (ROC-AUC). The accuracy score is the total percentage of the correct predictions of DILI label. Sensitivity is also called a true positive rate, which measures the percentage that drugs with positive DILI labels are truly predicted as positive. Specificity is the true negative rate, which represents the rate that drugs without DILI risks are correctly predicted as negative labels. The F1-score is the weighted average of precision and recall, where precision is the ratio of the correct positive predictions to all positive predictions, and recall is the ratio of the correct positive predictions to all ground truth positive labels. MCC leverages the performance of all four confusion matrix categories (true positives, false negatives, true negatives, and false positives). ROC-AUC measures the separability of the model to correctly predict positive labels as positive and negative labels as negative. In addition, we evaluate statistical significance using a one-sided Wilcoxon signed-rank test.

■ EXPERIMENTS

We have conducted extensive experiments using circular fingerprints (Circular-fp),[20] neural fingerprints (Neural-fp),[21] message passing neural network (MPNN),[25] directed message passing neural network (DMPNN),[26] and MV-GNN$^{\text{cross}}$[27] on DILI to validate the performance. We took MV-GNN$^{\text{cross}}$ as the backbone and employed our proposed property augmentation approach to involve more data, in order to further boost the prediction performance of DILI. Moreover, we conducted additional experiments using MPNN and DMPNN on augmented Tox-DILI data set to prove the effectiveness of our method.

Data Set Description. Two data sets are used during the experiments, DILI and Tox-DILI (see Supporting Information). DILI is the DILI data set provided by NIH, which contains 479 molecules with DILI label. The original DILI data set comes from the DILIrank[12] data set, which contains 197 molecules with Most-DILI-Concern, 282 molecules with No-DILI-Concern, and 464 molecules with Less-DILI-Concern. We consider Most-DILI-Concern as label 1 and No-DILI-concern as label 0 to solve the classification problem. Thus, 479 molecules in total are selected to constitute the DILI data set. The Tox-DILI is formed by DILI and four other data sets with toxicity relevant properties: herg,[33,34] PLD,[29] ames,[35,36] and mmp.[37,38] The description of each property is stated in Table 3, and the label distribution is shown in Table 4.

Table 3. Description of Four Toxicity Properties Used for Augmentation

category	property	description
toxicity	herg[33,34]	measures cardiotoxic effects of compounds
	PLD[29]	stands for phospholipidosis, which measures organism-level toxicity of compounds
	ames[35,36]	measures mutagenicity, one of the most important end points of toxicity
	mmp[37,38]	the mitochondrial membrane potential (MMP) is a key parameter for evaluating mitochondrial function

Table 4. Datasets Statstics

data set	data set size	property	no. of molecules	no. of label 0	no. of label 1
DILI	479	DILI	479	282	197
Tox-DILI	15,675	herg	3024	2541	483
		PLD	4159	3777	382
		ames	7940	3406	4534
		mmp	5970	5070	900
		DILI	479	282	197

Comparison Experiments. *Circular-fp.* Circular fingerprints (Circular-fp) is one of the traditional ways to generate so-called fingerprints to represent the molecule. It is a vector representation generated by a hand-crafted hash-based algorithm to define the local features. Circular-fp employs a fixed hash function to extract each layer's features of a molecule and concatenate them together. The generated vector representations usually go through machine learning models to perform further predictions, and we applied the GradientBoost[39] model here in the experiments.

Neural-fp. Neural fingerprints (Neural-fp) is constructed on a supervised deep graph convolutional neural network.[21] It applies convolutional neural networks on graphs directly. The difference between Neural-fp and Circular-fp is the replacement of the hash function. Neural-fp applies a nonlinear activated densely connected layer to generate the fingerprints.

MPNN. Another promising graph-based deep learning technique is the MPNN.[25] It recursively updates the atom features by aggregating the feature information from its neighbors and adjacent bonds and then pools all of the updated features of the atoms to deliver the global representation

Table 5. Performance of DILI Models Using Cross-Validation with Random Splitting[a]

	circular-fp	neural-fp	MPNN	DMPNN	MV-GNN^{cross}	property augmentation with Tox-DILI
accuracy	0.688 ± 0.051	0.704 ± 0.091	0.738 ± 0.094	0.750 ± 0.098	0.788 ± 0.077	**0.814** ± 0.047
sensitivity	0.364 ± 0.125	0.647 ± 0.091	0.727 ± 0.133	0.728 ± 0.135	0.762 ± 0.105	**0.768** ± 0.100
specificity	**0.879** ± 0.086	0.740 ± 0.087	0.752 ± 0.129	0.764 ± 0.172	0.809 ± 0.092	0.849 ± 0.097
F1-score	0.485 ± 0.091	0.615 ± 0.106	0.666 ± 0.124	0.681 ± 0.095	0.721 ± 0.105	**0.753** ± 0.063
MCC	0.289 ± 0.130	0.381 ± 0.191	0.473 ± 0.202	0.499 ± 0.179	0.562 ± 0.178	**0.621** ± 0.114
ROC-AUC	0.738 ± 0.056	0.753 ± 0.093	0.833 ± 0.075	0.832 ± 0.068	0.866 ± 0.055	**0.882** ± 0.031

[a] Higher is better. Best scores are marked as bold.

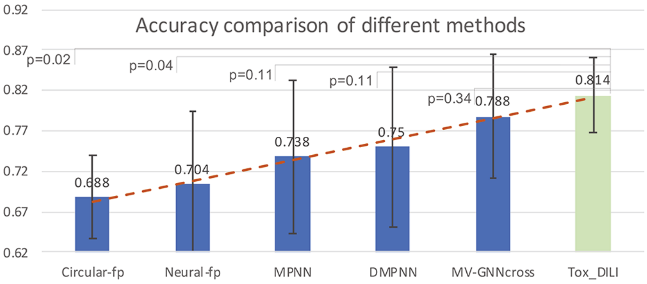


Figure 6. Performance comparison on the accuracy of different methods using cross-validation with random splitting (higher is better). Light green color indicates our proposed method. P indicates the *p*-value calculated from the Wilcoxon test between our proposed method and other baselines.

of each molecule via a readout function. The generated representation is then fed into the downstream molecular property prediction network.

DMPNN. Inspired by MPNN,[25] DMPNN[26] converts the passing process to bond-wise instead of atom-wise. Instead of aggregating the neighbor atoms' messages, DMPNN proposes a directed message passing scheme to avoid unnecessary loops. It aggregates the information on neighbor bonds with the same direction and takes the starter atom features as attached features to implement message passing. The following network is used to predict the property label as well.

MV-GNN^{cross}. MV-GNN^{cross} model extracts the atom messages and bond messages simultaneously. It considers atom message passing and bond message passing as two parallel streams and allows the atom/bond messages to communicate during the passing phase. A self-attention readout mechanism and a disagreement loss are employed to restrain the model training.

MV-GNN^{cross} with Property Augmentation. The results of different models on the DILI data set empirically demonstrate that MV-GNN^{cross} has achieved the highest prediction

Table 6. Performance Comparison between without Property Augmentation (DILI) and with Property Augmentation (Tox-DILI) Using Cross-Validation with Random Splitting[a]

	MPNN (DILI)	MPNN (Tox-DILI)	DMPNN (DILI)	DMPNN (Tox-DILI)	MV-GNN^{cross} (DILI)	MV-GNN^{cross} (Tox-DILI)
accuracy	0.738 ± 0.094	**0.788** ± 0.044	0.750 ± 0.098	**0.785** ± 0.024	0.788 ± 0.077	**0.814** ± 0.047
sensitivity	0.727 ± 0.133	**0.761** ± 0.072	0.728 ± 0.135	**0.748** ± 0.091	0.762 ± 0.105	**0.768** ± 0.100
specificity	0.752 ± 0.129	**0.807** ± 0.070	0.764 ± 0.172	**0.812** ± 0.045	0.809 ± 0.092	**0.849** ± 0.097
F1-score	0.666 ± 0.124	**0.728** ± 0.045	**0.764** ± 0.172	0.718 ± 0.045	0.721 ± 0.105	**0.753** ± 0.063
MCC	0.473 ± 0.202	**0.562** ± 0.082	0.499 ± 0.179	**0.553** ± 0.060	0.562 ± 0.178	**0.621** ± 0.114

[a] Higher scores within each pair-wise comparison are marked as **bold**.

Table 7. Performance of DILI Models Using Leave-one-out Cross-Validation[a]

	circular-fp	neural-fp	MPNN	DMPNN	MV-GNN^{cross}	property augmentation with Tox-DILI
accuracy	0.668 ± 0.085	0.683 ± 0.063	0.706 ± 0.057	0.715 ± 0.059	0.728 ± 0.047	**0.787** ± 0.070
sensitivity	0.351 ± 0.171	0.595 ± 0.089	0.590 ± 0.141	0.617 ± 0.140	0.651 ± 0.121	**0.721** ± 0.106
specificity	**0.899** ± 0.063	0.757 ± 0.081	0.798 ± 0.115	0.803 ± 0.107	0.791 ± 0.087	0.837 ± 0.062
F1-score	0.447 ± 0.175	0.604 ± 0.064	0.614 ± 0.078	0.631 ± 0.086	0.655 ± 0.076	**0.731** ± 0.076
MCC	0.294 ± 0.120	0.353 ± 0.118	0.406 ± 0.114	0.432 ± 0.113	0.448 ± 0.099	**0.558** ± 0.131
ROC-AUC	0.775 ± 0.069	0.734 ± 0.035	0.789 ± 0.072	0.792 ± 0.051	0.797 ± 0.039	**0.840** ± 0.064

[a] Higher is better. Best scores are marked as bold.

Table 8. Performance Comparison between without Property Augmentation (DILI) and with Property Augmentation (Tox-DILI) Using Leave-one-out Cross-Validation[a]

	MPNN (DILI)	MPNN (Tox-DILI)	DMPNN (DILI)	DMPNN (Tox-DILI)	MV-GNN^{cross} (DILI)	MV-GNN^{cross} (Tox-DILI)
accuracy	0.706 ± 0.057	**0.736** ± 0.074	0.715 ± 0.059	**0.748** ± 0.064	0.728 ± 0.047	**0.787** ± 0.070
sensitivity	0.590 ± 0.141	**0.625** ± 0.104	0.617 ± 0.140	**0.632** ± 0.099	0.651 ± 0.121	**0.721** ± 0.106
specificity	0.798 ± 0.115	**0.820** ± 0.117	0.803 ± 0.107	**0.817** ± 0.079	0.791 ± 0.087	**0.837** ± 0.062
F1-score	0.614 ± 0.078	**0.655** ± 0.090	0.631 ± 0.086	**0.657** ± 0.095	0.655 ± 0.076	**0.731** ± 0.076
MCC	0.406 ± 0.114	**0.456** ± 0.148	0.432 ± 0.113	**0.454** ± 0.135	0.448 ± 0.099	**0.558** ± 0.131
ROC-AUC	0.789 ± 0.072	**0.813** ± 0.070	0.792 ± 0.051	**0.806** ± 0.067	0.797 ± 0.039	**0.840** ± 0.064

[a] Higher scores within each pair-wise comparison are marked as bold.

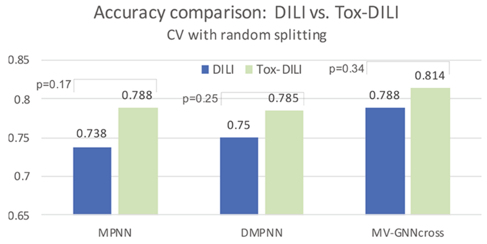


Figure 7. Cross-validation with random splitting. Visualization from Table 6. DILI indicates baseline, and Tox-DILI demonstrates the performance of utilizing property augmentation. The *p*-value is calculated between the two prediction results for each model.

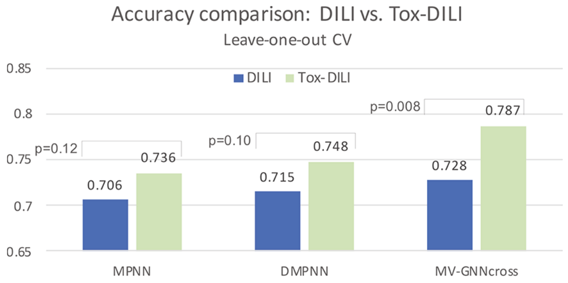


Figure 8. Leave-one-out cross-validation. Visualization from Table 8. DILI indicates baseline, and Tox-DILI demonstrates the performance of utilizing property augmentation. The *p*-value is calculated between the two prediction results for each model.

accuracy. Considering the extremely limited availability of DILI data, we propose to involve more data in a property augmentation fashion to facilitate training the molecular representation. In this regard, we combine DILI with four more data sets with other toxicity labels to form Tox-DILI data set and apply MV-GNN^{cross} model on it.

Additional Experiments with Property Augmentation. In order to further prove the effectiveness of proposed method, we conducted additional experiments on the Tox-DILI data set to compare the performance improvement from using DILI only. Since MPNN and DMPNN outperform circular-fp and neural-fp on the DILI data set and both of them are graph-based message passing models, we then utilize them to assess the prediction performance of the proposed property augmentation strategy.

Experimental Procedure. In order to thoroughly verify the superiority of the proposed method and eliminate the randomness, we conducted extensive experiments using three evaluation methods: 5-fold cross-validation with random splitting, 10-fold leave-one-out cross-validation, and 5-fold cross-validation with scaffold splitting. To make a fair comparison, we used the same data set splits over DILI and Tox-DILI for all the models, respectively. For each cross-validation (CV) method, we first ran all the models on the DILI data set and then applied property augmentation using MV-GNN^{cross} on the Tox-DILI data set to further boost the performance. Moreover, we took MPNN and DMPNN as backbones to implement property augmentation to confirm the effectiveness of our method. The pairwise comparison between experiments without and with property augmentation is visualized with a *p*-value calculated through the Wilcoxon test.

Cross-Validation with Random Splitting. We first applied the 5-fold cross-validation with random seeds to evaluate the performance of each model. In each fold, the input data set was randomly split into 8:1:1, while 80% was used for training, 10% is used for validation, and the last 10% was used for testing. For Tox-DILI, we ensured each data split contained balanced data for each property. We calculated the mean and standard deviation of the results from all folds as the final results.

Leave-one-out Cross-Validation. Considering the randomness of data set splits in the first evaluation method, we then applied the 10-fold leave-one-out cross-validation to evaluate the performance again. The input data set was split into 10 folds equally, and each fold has been used as the testing data set in sequence. Within the remaining nine folds, one fold is used as the validation data set, and the rest are used for training. We took the average of the results from all folds as the final results.

Cross-Validation with Scaffold Splitting. Other than the two commonly used evaluation methods, we also conducted experiments with scaffold splitting, which is more practical and challenging than random splitting. Scaffold splitting splits the molecules with distinct two-dimensional structural frameworks

Table 9. Performance of DILI Models Using Cross-Validation with Scaffold Splitting[a]

	circular-fp	neural-fp	MPNN	DMPNN	MV-GNN^{cross}	property augmentation with Tox-DILI
accuracy	0.657 ± 0.037	0.665 ± 0.048	0.706 ± 0.010	0.714 ± 0.043	0.735 ± 0.045	**0.765** ± 0.047
sensitivity	0.485 ± 0.074	0.642 ± 0.066	0.695 ± 0.098	0.693 ± 0.082	0.684 ± 0.094	**0.765** ± 0.090
specificity	**0.784** ± 0.073	0.688 ± 0.082	0.708 ± 0.066	0.724 ± 0.062	0.765 ± 0.099	0.774 ± 0.046
F1-score	0.533 ± 0.049	0.609 ± 0.062	0.653 ± 0.052	0.660 ± 0.070	0.674 ± 0.060	**0.740** ± 0.036
MCC	0.284 ± 0.086	0.328 ± 0.103	0.402 ± 0.027	0.415 ± 0.090	0.458 ± 0.087	**0.534** ± 0.089
ROC-AUC	0.719 ± 0.028	0.744 ± 0.051	0.758 ± 0.025	0.782 ± 0.040	0.774 ± 0.042	**0.834** ± 0.022

[a] Higher is better. Best scores are marked as bold.

Table 10. Performance Comparison between without Property Augmentation (DILI) and with Property Augmentation (Tox-DILI) Using Cross-Validation with Scaffold Splitting[a]

	MPNN (DILI)	MPNN (Tox-DILI)	DMPNN (DILI)	DMPNN (Tox-DILI)	MV-GNN^{cross} (DILI)	MV-GNN^{cross} (Tox-DILI)
accuracy	0.706 ± 0.010	**0.727** ± 0.030	0.714 ± 0.043	**0.741** ± 0.040	0.735 ± 0.045	**0.765** ± 0.047
sensitivity	0.695 ± 0.098	**0.727** ± 0.102	0.693 ± 0.082	**0.801** ± 0.073	0.684 ± 0.094	**0.765** ± 0.090
specificity	0.708 ± 0.066	**0.716** ± 0.108	**0.724** ± 0.062	0.669 ± 0.110	0.765 ± 0.099	**0.774** ± 0.046
F1-score	0.653 ± 0.052	**0.717** ± 0.049	0.660 ± 0.070	**0.748** ± 0.052	0.674 ± 0.060	**0.740** ± 0.036
MCC	0.402 ± 0.027	**0.452** ± 0.064	0.415 ± 0.090	**0.482** ± 0.082	0.458 ± 0.087	**0.534** ± 0.089
ROC-AUC	0.758 ± 0.025	**0.796** ± 0.052	0.782 ± 0.040	**0.814** ± 0.072	0.774 ± 0.042	**0.834** ± 0.022

[a] Higher scores within each pair-wise comparison are marked as bold.

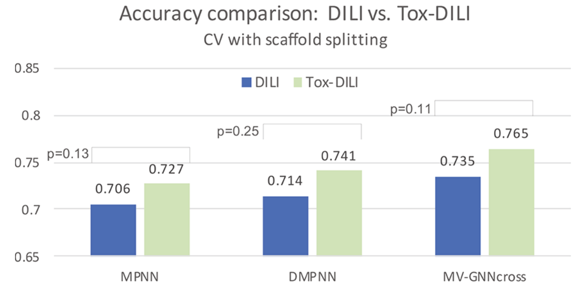
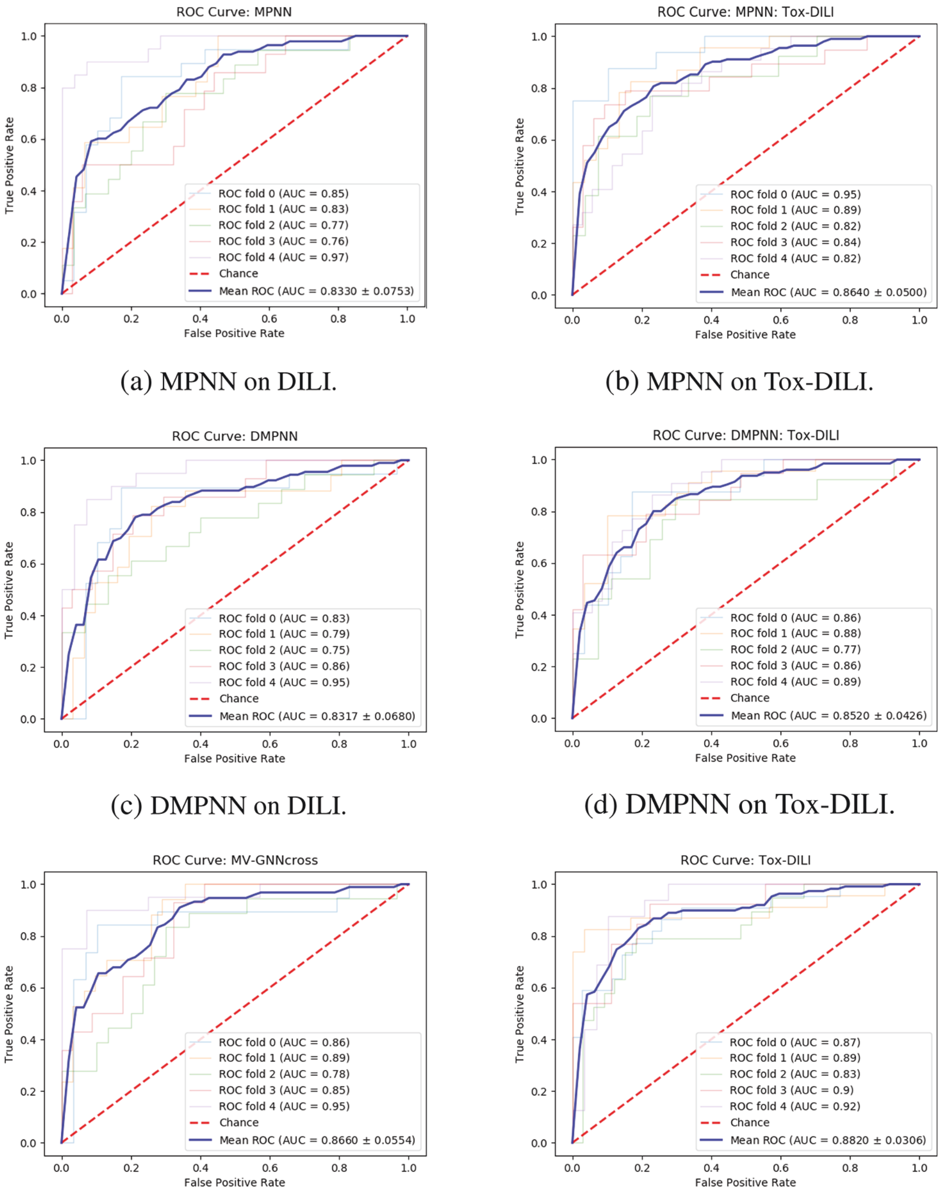


Figure 9. Cross-validation with scaffold splitting. Visualization from Table 10. DILI indicates baseline, and Tox-DILI demonstrates the performance of utilizing property augmentation. The *p*-value is calculated between the two prediction results for each model.

into different subsets,[40] which can be considered as a clustering process based on the molecular structure prior to the training process. We followed the process introduced in Yang et al.[26] The molecules in the data set are categorized into bins based on their Murcko scaffold, which are calculated by RDKit.[28] The bins are then randomly put into train, validation, and test data sets. We applied a five-fold cross-validation here with 8:1:1 train/validation/test splits too and calculated the mean and standard deviation as the final results.

RESULTS AND DISCUSSION

Other than the prediction accuracy, we also analyze the predicted labels with the ground truth labels in detail by computing the sensitivity, specificity, F-1 score, MCC, and ROC-AUC. All of these evaluation criteria are important since we expect to find a model that can filter the drugs with potential DILI concern as well as pick out the drugs without DILI risks, thus



(a) MPNN on DILI.

(b) MPNN on Tox-DILI.

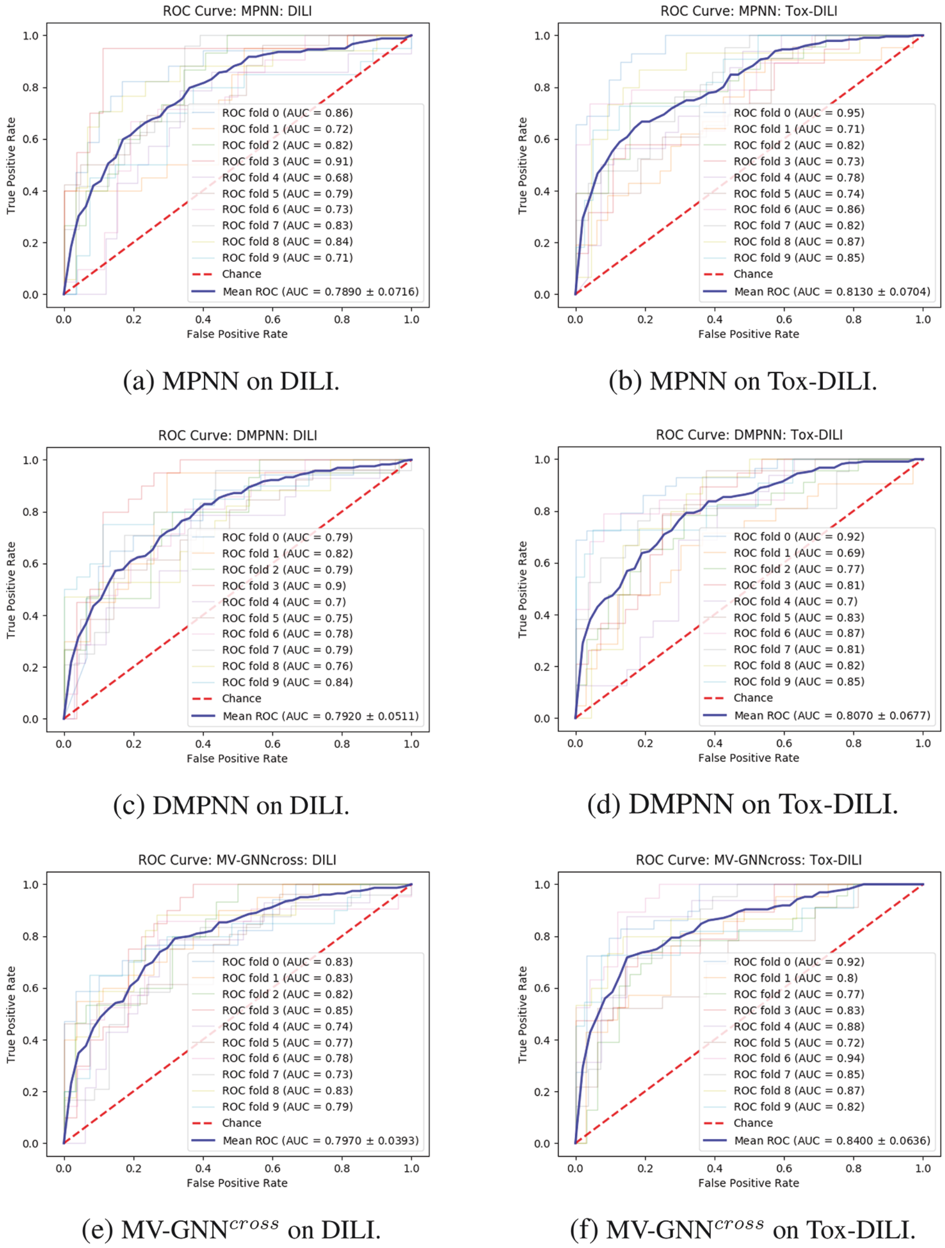
(c) DMPNN on DILI.

(d) DMPNN on Tox-DILI.

(e) MV-GNN^{cross} on DILI.

(f) MV-GNN^{cross} on Tox-DILI.

Figure 10. Cross-validation with random splitting. ROC curve comparison (larger AUC is better) between without property augmentation (DILI) and with property augmentation (Tox-DILI). The lighter lines demonstrate the performance of each fold, and the blue line represents the mean AUC for each method.



(a) MPNN on DILI.

(b) MPNN on Tox-DILI.

(c) DMPNN on DILI.

(d) DMPNN on Tox-DILI.

(e) MV-GNN^{cross} on DILI.

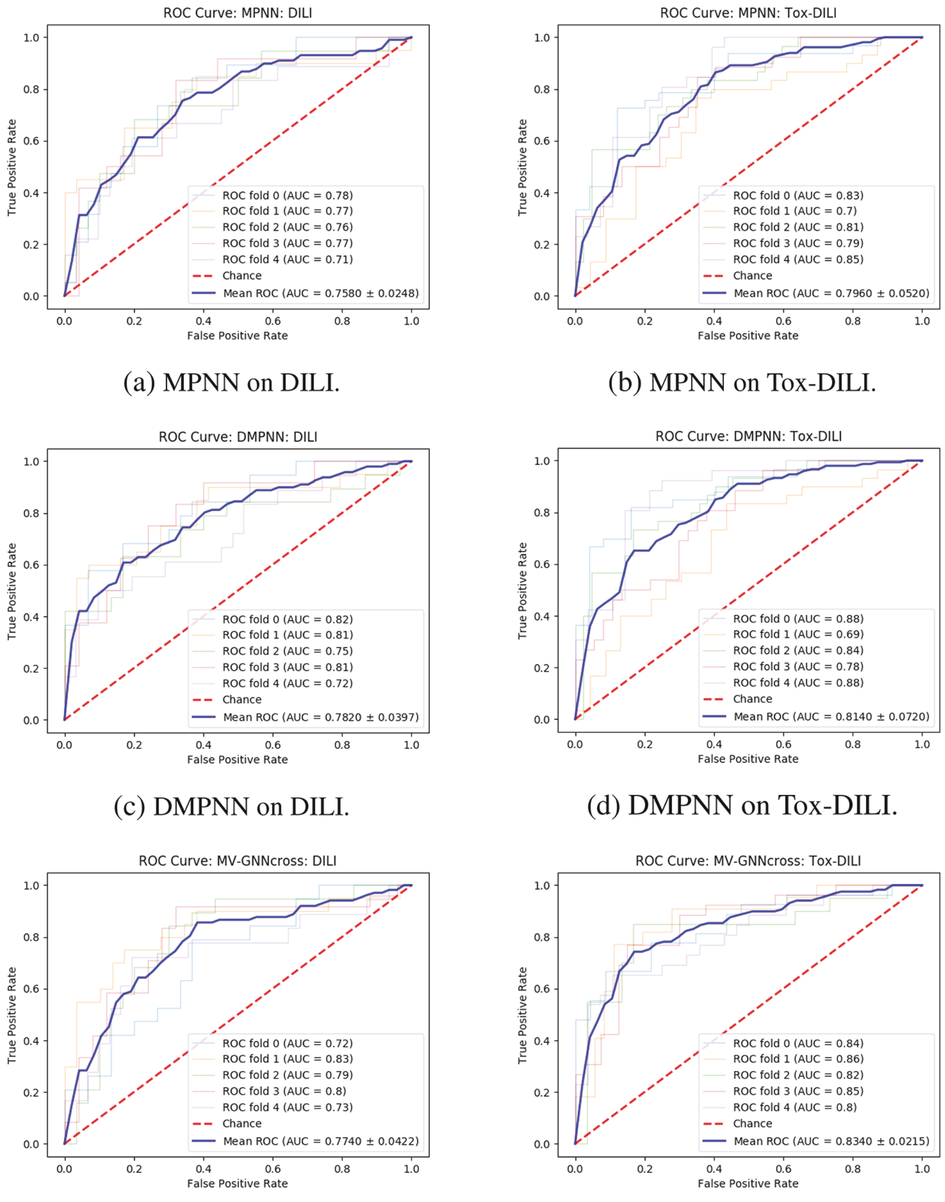
(f) MV-GNN^{cross} on Tox-DILI.

Figure 11. Leave-one-out cross-validation. ROC curve comparison (larger AUC is better) between without property augmentation (DILI) and with property augmentation (Tox-DILI). The lighter lines demonstrate the performance of each fold, and the blue line represents the mean AUC for each method.

further experiments can be conducted on these approved drug candidates.

Cross-Validation with Random Splitting. The prediction performance of cross-validation with random splitting are shown in Table 5 and visualized in Figure 6. As observed, graph-based message passing models generally perform better than other baselines on the DILI data set. Meanwhile, the MV-GNN^{cross} model outperforms other message passing methods as well as is equiped with smaller variance. The augmentation strategy that combines more data with other properties precisely improves the performance of DILI to 81.4%, which empirically proves that involving more property data to co-train the model indeed brings more information. In this fashion, the MV-GNN^{cross} model gains an accuracy boost by 2.6% compared with the vanilla MV-GNN^{cross}. The *p*-values obtained from the Wilcoxon test may not be sufficiently small for some baselines considering the difficulty and challenge for the DILI prediction problem, yet we believe our proposed method has accomplished remarkable improvement.

As our goal is to identify drugs that might cause DILI and sort out drugs without DILI, a model with high scores of all the evaluation metrics as well as a balanced sensitivity/specificity would be more helpful. As shown in Table 5, circular-fp has a very high specificity but extremely low sensitivity, so it is more likely to identify drugs without DILI as positive. The lowest MCC verifies that it cannot achieve a balanced prediction over positive and negative labels. All the criteria values of neural-fp are not significant. MPNN and DMPNN have almost equal sensitivity and specificity scores, but the in terms of the overall accuracy, F1-score and MCC are not notably high. The accuracy, sensitivity, F1-score, and MCC of MV-GNN^{cross} are higher than other baselines on the DILI data set. The



(a) MPNN on DILI.

(b) MPNN on Tox-DILI.

(c) DMPNN on DILI.

(d) DMPNN on Tox-DILI.

(e) MV-GNN^{cross} on DILI.

(f) MV-GNN^{cross} on Tox-DILI.

Figure 12. Cross-validation with scaffold splitting. ROC curve comparison (larger AUC is better) between without property augmentation (DILI) and with property augmentation (Tox-DILI). The lighter lines demonstrate the performance of each fold, and the blue line represents the mean AUC for each method.

specificity score is slightly lower than circular-fp, but is still competitive. MV-GNN^{cross} utilizing a property augmentation strategy obtained the highest accuracy score, which is 81.4%. The specificity score is fairly high at 0.849, and a sensitivity score of 0.768 is also the highest compared with other baselines. The comparisons of F1-score and MCC confirm that our MV-GNN^{cross} model with property augmentation significantly performs better than other models on the DILI prediction task.

We also conducted additional experiments with our method utilizing MPNN and DMPNN, where the performance is compared in Table 6 in a pairwise manner (DILI vs Tox-DILI). The accuracy improvement is visualized in Figure 7, and the ROC-AUC is plot in Figure 10. We can observe that models with a proposed property augmentation almost outperform the other one over all evaluation criteria.

We can observe the performance comparison between each model based on Figure 10. Figure 10 visualizes the ROC-AUC for each model. As we know, the larger AUC represents better model performance. When the inflection point is close to the left top corner, the AUC is approximate to 1. Figure 10f illustrates that MV-GNN^{cross} on Tox-DILI outperforms other models.

Leave-one-out Cross-Validation. To eliminate the randomness of splitting method, we use a 10-fold leave-one-out cross-validation to rerun all the experiments. The performance is shown in Tables 7 and 8. The results follow the similar trend as obtained using cross-validation with random splitting. MV-GNN^{cross} with property augmentation learning performs best over all evaluation criteria except for specificity, where circular-fp obtains highest value. However, the other performance results such as sensitivity, MCC, and F1-score indicate that the prediction results of circular-fp are extremely unbalanced. The accuracy and ROC-AUC visualization between without and with property augmentation on MPNN, DMPNN and MV-GNN^{cross}, which are shown in Figures 8 and 11, further prove the superiority of proposed method. As shown in Figure 8, the *p*-value calculated from MV-GNN^{cross} without and with property augmentation is <0.01, which can be considered as statistically significant. The prediction results with leave-one-out cross-validation confirm that our method is capable of improving the prediction performance of DILI.

Cross-Validation with Scaffold Splitting. Last, we challenge the most difficult but practical scenario by conducting experiments using scaffold splitting. The results are recorded in Tables 9 and 10, while the accuracy and ROC-AUC are visualized in Figures 9 and 12. The accuracy scores dropped compared with random splitting, which is reasonable considering the strict splitting. However, other criteria such as F1-score and MCC do not vary much, and the general trending is still similar to the performance obtained from the other two evaluation methods. MV-GNN^{cross} with property augmentation learning outperforms all other methods, including MPNN and DMPNN with property augmentation, which effectively illustrates the superiority of proposed method.

In addition to extensive experiments, several studies have investigated different methods to tackle the DILI prediction problem in years. Two recent works from Ancuceanu et al. and Minerali et al. also seek appropriate approaches to enhance the prediction performance of DILIrank. Minerali et al. utilizes a Bayesian model to obtain an ROC-AUC of 0.814, a sensitivity of 0.741, a specificity of 0.755, and an accuracy of 0.746. The sensitivity/specificity is nearly perfectly balanced which denotes the model holds stabilized expressive power, but the ROC-AUC and accuracy are not remarkable compared with deep graph-based models. Ancuceanu et al. explores different features of selection and various machine learning algorithms to build meta-models. Some models have achieved up to 95% sensitivity but have low specificity around 50%, and some models have reletively balanced sensitivity/specificity (e.g., 76%/73.2%), yet the accuracy is <0.75%. Ergo, the superiority of our deep graph-based model is empirically demonstrated along with a property augmentation strategy.

CONCLUSIONS

Enhancing the prediction performance of DILI is crucial for drug development. Current studies generally focus on either bringing in more features, stacking multiple models, or enlarging the data set. These attempts have attained impressive achievements. In spite of this, we notice that certain properties of the drugs might contain hidden correlations between each other. Hence, we propose to establish a property augmentation approach to include more information to boost the training. Extensive experiments on Tox-DILI confirm the superiority of our method by improving the accuracy to 81.4% using cross-validation with random splitting, 78.7% using leave-one-out cross-validation, and 76.5% with cross-validation with scaffold splitting. The proposed method not only brings in more input data for the encoder network to learn better molecular vector representation but also utilizes the correlations between different property labels during the prediction network. We believe it is a promising perspective to improve the prediction performance of DILI as well as other properties with limited available data.

ASSOCIATED CONTENT

Supporting Information

Data format, the SMILES representation of the molecule along with the corresponding property label; labels not observed are displayed as missing values. The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.chemrestox.0c00322.

DILI.xlsx: Data set used for experiments without property augmentation learning (XLSX)

Tox-DILI.xlsx: Data set used for experiments with property augmentation learning (XLSX)

AUTHOR INFORMATION

Corresponding Author

Junzhou Huang − *Department of Computer Science, University of Texas at Arlington, Arlington, Texas 76013, United States*; Email: jzhuang@uta.edu

Authors

Hehuan Ma − *Department of Computer Science, University of Texas at Arlington, Arlington, Texas 76013, United States*; orcid.org/0000-0002-5971-0053

Weizhi An − *Department of Computer Science, University of Texas at Arlington, Arlington, Texas 76013, United States*

Yuhong Wang − *National Center for Advancing Translating Sciences, National Institutes of Health, Rockville, Maryland 20850, United States*

Hongmao Sun − *National Center for Advancing Translating Sciences, National Institutes of Health, Rockville, Maryland 20850, United States*

Ruili Huang − *National Center for Advancing Translating Sciences, National Institutes of Health, Rockville, Maryland 20850, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.chemrestox.0c00322

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

This work was partially supported by U.S. National Science Foundation IIS-1718853, the CAREER grant IIS-1553687, and Cancer Prevention and Research Institute of Texas (CPRIT) award (RP190107).

REFERENCES

(1) Paul, S. M., Mytelka, D. S., Dunwiddie, C. T., Persinger, C. C., Munos, B. H., Lindborg, S. R., and Schacht, A. L. (2010) How to improve R&D productivity: the pharmaceutical industry's grand challenge. *Nat. Rev. Drug Discovery 9*, 203−214.

(2) DiMasi, J. A., Grabowski, H. G., and Hansen, R. W. (2015) The cost of drug development. *N. Engl. J. Med. 372*, 1972.

(3) Berggren, R., Møller, M., Moss, R., Poda, P., and Smietana, K. (2012) Outlook for the next 5 years in drug innovation. *Nat. Rev. Drug Discovery 11*, 435−436.

(4) Thakkar, S., Li, T., Liu, Z., Wu, L., Roberts, R., and Tong, W. (2020) Drug-induced liver injury severity and toxicity (DILIst): Binary classification of 1279 drugs by human hepatotoxicity. *Drug Discovery Today 25*, 201−208.

(5) Parasrampuria, D. A., Benet, L. Z., and Sharma, A. (2018) Why drugs fail in late stages of development: case study analyses from the last decade and recommendations. *AAPS J. 20*, 46.

(6) Kullak-Ublick, G. A., Andrade, R. J., Merz, M., End, P., Benesic, A., Gerbes, A. L., and Aithal, G. P. (2017) Drug-induced liver injury: recent advances in diagnosis and risk assessment. *Gut 66*, 1154−1164.

(7) Olson, H., Betton, G., Robinson, D., Thomas, K., Monro, A., Kolaja, G., Lilly, P., Sanders, J., Sipes, G., Bracken, W., et al. (2000) Concordance of the toxicity of pharmaceuticals in humans and in animals. *Regul. Toxicol. Pharmacol. 32*, 56−67.

(8) Greene, N., Fisk, L., Naven, R. T., Note, R. R., Patel, M. L., and Pelletier, D. J. (2010) Developing Structure−Activity Relationships for the Prediction of Hepatotoxicity. *Chem. Res. Toxicol. 23*, 1215−1222.

(9) Zhu, X., and Kruhlak, N. L. (2014) Construction and analysis of a human hepatotoxicity database suitable for QSAR modeling using post-market safety data. *Toxicology 321*, 62−72.

(10) Xu, J. J., Henstock, P. V., Dunn, M. C., Smith, A. R., Chabot, J. R., and de Graaf, D. (2008) Cellular Imaging Predictions of Clinical Drug-Induced Liver Injury. *Toxicol. Sci. 105*, 97−105.

(11) Sakatis, M. Z., Reese, M. J., Harrell, A. W., Taylor, M. A., Baines, I. A., Chen, L., Bloomer, J. C., Yang, E. Y., Ellens, H. M., Ambroso, J. L., et al. (2012) Preclinical Strategy to Reduce Clinical Hepatotoxicity Using in Vitro Bioactivation Data for > 200 Compounds. *Chem. Res. Toxicol. 25*, 2067−2082.

(12) Chen, M., Suzuki, A., Thakkar, S., Yu, K., Hu, C., and Tong, W. (2016) DILIrank: the largest reference drug list ranked by the risk for developing drug-induced liver injury in humans. *Drug Discovery Today 21*, 648−653.

(13) Chen, M., Vijay, V., Shi, Q., Liu, Z., Fang, H., and Tong, W. (2011) FDA-approved drug labeling for the study of drug-induced liver injury. *Drug Discovery Today 16*, 697−703.

(14) Minerali, E., Foil, D. H., Zorn, K. M., Lane, T. R., and Ekins, S. (2020) Comparing Machine Learning Algorithms for Predicting Drug-Induced Liver Injury (DILI). *Mol. Pharmaceutics 17*, 2628−2637.

(15) Aleo, M. D., Shah, F., Allen, S., Barton, H. A., Costales, C., Lazzaro, S., Leung, L., Nilson, A., Obach, R. S., Rodrigues, A. D., et al. (2020) Moving beyond Binary Predictions of Human Drug-Induced Liver Injury (DILI) toward Contrasting Relative Risk Potential. *Chem. Res. Toxicol. 33*, 223−238.

(16) Mora, J. R., Marrero-Ponce, Y., García-Jacas, C. R., and Suarez Causado, A. (2020) Ensemble Models Based on QuBiLS-MAS Features and Shallow Learning for the Prediction of Drug-Induced Liver Toxicity: Improving Deep Learning and Traditional Approaches. *Chem. Res. Toxicol. 33*, 1855−1873.

(17) Ancuceanu, R., Hovanet, M. V., Anghel, A. I., Furtunescu, F., Neagu, M., Constantin, C., and Dinu, M. (2020) Computational Models Using Multiple Machine Learning Algorithms for Predicting Drug Hepatotoxicity with the DILIrank Dataset. *Int. J. Mol. Sci. 21*, 2114.

(18) Morgan, H. L. (1965) The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *J. Chem. Doc. 5*, 107−113.

(19) O'Boyle, N. M., Campbell, C. M., and Hutchison, G. R. (2011) Computational Design and Selection of Optimal Organic Photovoltaic Materials. *J. Phys. Chem. C 115*, 16200−16210.

(20) Rogers, D., and Hahn, M. (2010) Extended-Connectivity Fingerprints. *J. Chem. Inf. Model. 50*, 742−754.

(21) Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. (2015) Convolutional Networks on Graphs for Learning Molecular Fingerprints. *Advances in Neural Information Processing Systems*, pp 2224−2232, Neural Information Processing Systems, San Diego, CA.

(22) Le, T., Winter, R., Noé, F., and Clevert, D.-A. (2020) Neuraldecipher - Reverse-Engineering ECFP Fingerprints to Their Molecular Structures. *Chem. Sci. 11*, 10378.

(23) Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. (2018) MoleculeNet: a benchmark for molecular machine learning. *Chemical Science 9*, 513−530.

(24) Li, R., Wang, S., Zhu, F., and Huang, J. (2018) Adaptive Graph Convolutional Neural Networks. Proceedings from the *Thirty-second AAAI conference on artificial intelligence*, February 2−7, 2018, New Orleans, LA, pp 3546−3553, Association for the Advancement of Artificial Intelligence, Menlo Park, CA.

(25) Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017) Neural Message Passing for Quantum Chemistry. Proceedings of the *34th International Conference on Machine Learning*, August 6−11, 2017, Sydney, Australia, pp 1263−1272, International Machine Learning Society, San Diego, CA.

(26) Yang, K., Swanson, K., Jin, W., Coley, C., Eiden, P., Gao, H., Guzman-Perez, A., Hopper, T., Kelley, B., Mathea, M., et al. (2019) Analyzing Learned Molecular Representations for Property Prediction. *J. Chem. Inf. Model. 59*, 3370−3388.

(27) Ma, H., Rong, Y., Huang, W., Xu, T., Xie, W., Ye, G., and Huang, J. (2020) *Multi-View Graph Neural Networks for Molecular Property Prediction. arXiv (Quantitative Methods)*, June 12, 2020, 2005.13607, https://arxiv.org/abs/2005.13607 (accessed 2020-09-25).

(28) Landrum, G. et al. (2006) *RDKit: Open-source cheminformatics Software*, http://www.rdkit.org/

(29) Shahane, S. A., Huang, R., Gerhold, D., Baxa, U., Austin, C. P., and Xia, M. (2014) Detection of Phospholipidosis Induction: A Cell-Based Assay in High-Throughput and High-Content Format. *J. Biomol. Screening 19*, 66−76.

(30) Sun, S. (2013) A survey of multi-view machine learning. *Neural Computing and Applications 23*, 2031−2038.

(31) Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018) Graph Attention Networks. Proceedings from the *International Conference on Learning Representations*, April 30−May 3, 2018, Vancouver, Canada, International Conference on Representation Learning, La Jolla, CA.

(32) Li, J., Rong, Y., Cheng, H., Meng, H., Huang, W., and Huang, J. (2019) Semi-Supervised Graph Classification: A Hierarchical Graph Perspective. *World Wide Web Conference*, 972−982.

(33) Sun, H., Xia, M., Shahane, S. A., Jadhav, A., Austin, C. P., Huang, R., et al. (2013) Are hERG channel blockers also phospholipidosis inducers? *Bioorg. Med. Chem. Lett. 23*, 4587−4590.

(34) Xia, M., Shahane, S. A., Huang, R., Titus, S. A., Shum, E., Zhao, Y., Southall, N., Zheng, W., Witt, K. L., Tice, R. R., et al. (2011) Identification of quaternary ammonium compounds as potent inhibitors of hERG potassium channels. *Toxicol. Appl. Pharmacol. 252*, 250−258.

(35) Kazius, J., McGuire, R., and Bursi, R. (2005) Derivation and Validation of Toxicophores for Mutagenicity Prediction. *J. Med. Chem. 48*, 312−320.

(36) Hansen, K., Mika, S., Schroeter, T., Sutter, A., Ter Laak, A., Steger-Hartmann, T., Heinrich, N., and Müller, K.-R. (2009) Benchmark Data Set for in Silico Prediction of Ames Mutagenicity. *J. Chem. Inf. Model. 49*, 2077−2081.

(37) Attene-Ramos, M. S., Huang, R., Michael, S., Witt, K. L., Richard, A., Tice, R. R., Simeonov, A., Austin, C. P., and Xia, M. (2015) Profiling of the Tox21 Chemical Collection for Mitochondrial Function to Identify Compounds that Acutely Decrease Mitochondrial Membrane Potential. *Environ. Health Perspect. 123*, 49−56.

(38) Attene-Ramos, M. S., Huang, R., Sakamuru, S., Witt, K. L., Beeson, G. C., Shou, L., Schnellmann, R. G., Beeson, C. C., Tice, R. R., Austin, C. P., et al. (2013) Systematic Study of Mitochondrial Toxicity of Environmental Chemicals Using Quantitative High Throughput Screening. *Chem. Res. Toxicol. 26*, 1323−1332.

(39) Friedman, J. H. (2001) Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics 29*, 1189−1232.

(40) Bemis, G. W., and Murcko, M. A. (1996) The Properties of Known Drugs. 1. Molecular Frameworks. *J. Med. Chem. 39*, 2887−2893.