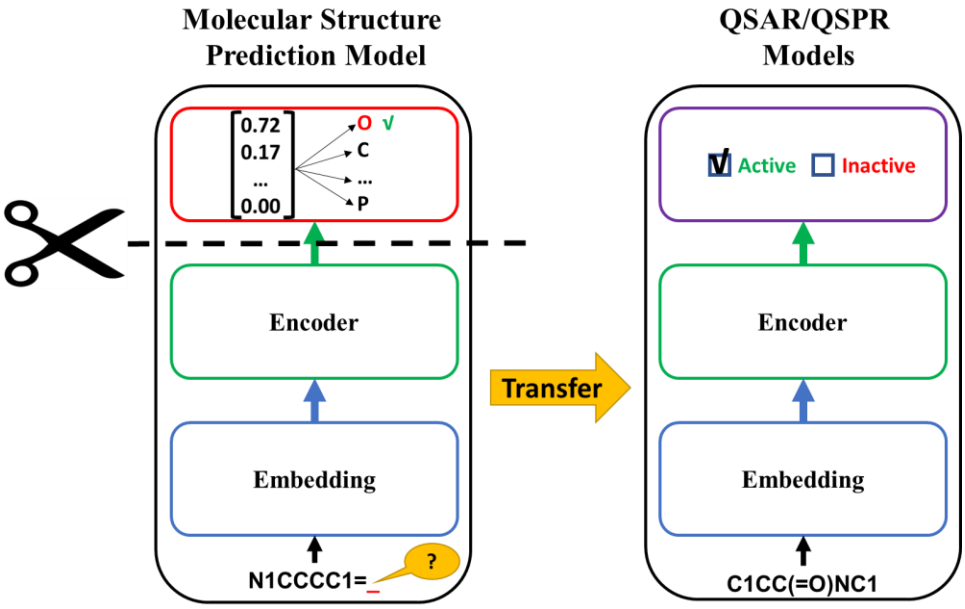
Inductive Transfer Learning for Molecular Activity Prediction: *Next-Gen QSAR Models with MolPMoFiT*

Xinhao Li & Denis Fourches*

Department of Chemistry, Bioinformatics Research Center, North Carolina State University, Raleigh, NC 27695, United States.

** To whom correspondence should be sent. Email: dfourch@ncsu.edu*

Table of Content Graphic



Abstract

Deep neural networks can directly learn from chemical structures without extensive, user-driven selection of descriptors in order to predict molecular properties/activities with high reliability. But these approaches typically require large training sets to learn the endpoint-specific structural features and ensure reasonable prediction accuracy. Even though large datasets are becoming the new normal in drug discovery, especially when it comes to high-throughput screening or metabolomics datasets, one should also consider smaller datasets with challenging endpoints to model and forecast. Thus, it would be highly relevant to better utilize the tremendous compendium of unlabeled compounds from publicly-available datasets for improving the model performances for the user's particular series of compounds. In this study, we propose the **Mol**ecular **P**rediction **Mo**del **Fi**ne-**T**uning (**MolPMoFiT**) approach, an effective transfer learning method based on self-supervised pre-training + task-specific fine-tuning for QSPR/QSAR modeling. A large-scale molecular structure prediction model is pre-trained using one million unlabeled molecules from ChEMBL in a self-supervised learning manner, and can then be fine-tuned on various QSPR/QSAR tasks for smaller chemical datasets with specific endpoints. Herein, the method is evaluated on four benchmark datasets (lipophilicity, FreeSolv, HIV, and blood-brain barrier penetration). The results showed the method can achieve strong performances for all four datasets compared to other *state-of-the-art* machine learning modeling techniques reported in the literature so far.

Keywords: Transfer learning, Neural networks, Self-supervised learning, QSPR/QSAR

1. Introduction

Predicting properties/activities of chemicals from their structures is one of the key objectives in cheminformatics and molecular modeling. Quantitative structure property/activity relationship (QSPR/QSAR) modeling[1–6] relies on machine learning techniques to establish quantified links between molecular structures and their experimental properties/activities. When using a classic machine learning approach, the training process is divided into two main steps: feature extraction/calculation and the actual modeling. The features (also called *descriptors*) characterizing the molecular structures are critical for the model performances. They typically encompass 2D molecular fingerprints, topological indices, or substructural fragments, as well as more complex 3D and 4D descriptors[7, 8] directly computed from the molecular structures[9].

Deep learning methods have demonstrated remarkable performances in several QSPR/QSAR case studies. In addition to use expert-engineered molecular descriptors as input, those techniques can also directly take molecular structures (*e.g.*, molecular graph[10–19][20, 21], SMILES strings[22–24], and molecular 2D/3D grid image[25–30]) and learn the data-driven feature representations for predicting properties/activities. As a result, this type of approach is potentially able to capture and extract underlying, complex structural patterns and feature ⇔ property relationships given sufficient amount of training data. The knowledge derived from these dataset-specific descriptors can then be used to better interpret and understand the structure-property relationships as well as to design new compounds. In a large scale benchmark study, Yang et.al[12] shown that a graph convolutional model that construct a learned representation from molecular graph consistently matches or outperforms models trained with expert-engineered molecular descriptors/fingerprints.

Graph convolutional neural networks (GCNN) directly operate on molecular graphs.[10] A molecular graph is an undirected graph whose nodes correspond to the atoms of the molecule and edges correspond to chemical bonds. GCNNs iteratively update the nodes representation by aggregating the representations of their neighboring nodes and/or edges. After *k* iterations of aggregation, the final nodes representations capture the local structure information within their *k*-hop graph neighborhood (which is somehow similar to augmented substructural fragments[31] but in a more data-driven manner). Moreover, the Simplified Molecular-Input Line-Entry System (SMILES)[32, 33] encodes the molecular structures as strings of text. Widely used in the field of cheminformatics, the SMILES format can be considered as an analogue of natural language. As a result, deep learning model architectures such as RNNs[34, 35], CNNs[36] and transformers[37] can be directly applied to SMILES for QSAR/QSPR tasks. While deep learning models have achieved *state-of-the-art* results on a variety of molecular properties/activities prediction tasks, these *end-to-end* models require very large amount of training data to learn useful feature representations. The learned representations are usually endpoint-specific, which means the models need to be built and retrained from scratch for the new endpoint/dataset of interest. Small chemical datasets with challenging endpoints to model are thus still disadvantaged with these techniques and unlikely to lead to models with reasonable prediction accuracy. As of today, this is considered as a grand challenge for QSAR modelers facing small sets of compounds without a clear path for obtaining reliable models for the endpoint of interest.

Meanwhile, transfer learning is a quickly emerging technique based on the general idea of reusing a pre-trained model built on a large dataset as the starting point for building a new, more optimized model for a target endpoint of interest. It is now widely used in the field of computer vision (CV) and natural language processing (NLP). In CV, a pre-trained deep learning model on

ImageNet[38] can be used as the start point to fine-tune for a new task[39]. Transfer learning in NLP has historically been restricted to the *shallow* word embeddings: NLP models start with embedding layers initialized with pretrained weights from Word2Vec[40], GloVe[41] or fastText[42]. This approach only uses the *prior* knowledge for the first layer of a model, the remaining layers still need to be trained and optimized from scratch. Language model pre-training[43–47] extends this approach by transferring all the learned optimized weights from multiple layers, which providing *contextualized* word embeddings for the downstream tasks. Language scale pre-trained language models have greatly improved the performance on a variety of language tasks. The default task for a language model is to predict the next word given the past sequence. The input and labels of the dataset used to train a language model are provided by the text itself. This is known as *self-supervised learning*. Self-supervised learning opens up a huge opportunity for better utilizing unlabeled data.

Due to the limited amount and sparsity of labeled datasets for certain types of endpoints in chemistry (*e.g.*, inhibitor residence times, allosteric inhibition, renal clearance), several transfer learning methods have been developed for allowing the development of QSPR/QSAR models for those types of endpoints/datasets. Inspired by ImageNet pretraining, Goh et al. proposed ChemNet[26] for transferable chemical property prediction. A deep neural network was pre-trained in a supervised manner on the ChEMBL[48] database using computed molecular descriptors as labels, then fine-tuned on other QSPR/QSAR tasks. Jaeger et al.[49] developed Mol2vec which employed the same idea of Word2Vec in NLP. Mol2vec learns the vector representations of molecular substructures in an unsupervised learning manner. Vectors of closely related molecular substructures are close to each other in the vector space. Molecular representations are computed by summing up the vectors of the individual substructures and be

used as input for QSPR/QSAR models. Hu et. al. pre-trained graph neural networks (GNNs) using both unlabeled data and labeled data from related auxiliary supervised tasks. The pre-trained GNNs were shown to significantly increase the model performances[50]. Multitask learning (MLT) is a related field to transfer learning, aiming at improving the performance of multiple tasks by learning them jointly. Multitask DNNs (deep neural networks) for QSAR were notably introduced by the winning team in the Kaggle QSAR competition and then applied in other QSAR/QSPR studies.[51–58] MTL is particularly useful if the endpoints share a significant relationship. However, MTL requires the tasks to be trained from scratch every time.

Herein, we propose the **Mol**ecular **P**rediction **Mo**del **Fi**ne-**T**uning (**MolPMoFiT –** pronounced *MOLMOFIT*), an effective transfer learning method based on self-supervised pre-training + task-specific fine-tuning for QSPR/QSAR modeling . In the current version, a molecular structure prediction model (MSPM) is pre-trained using one million bioactive molecules from ChEMBL and then fine-tuned for various QSPR/QSAR tasks. This method is "*universal*" in the sense that the pre-trained molecular structure prediction model can be used as a source for any other QSPR/QSAR models dedicated to a specific endpoint and a smaller dataset (*e.g.*, molecular series of congeneric compounds). This approach could constitute a first look at next-gen QSAR models being capable of high prediction reliability even for small series of compounds and highly challenging endpoints.

2. METHODS

2.1. ULMFiT

The MolPMoFiT method we proposed here is adapted from the ULMFiT (**U**niversal **L**anguage **M**odel **Fi**ne-**T**uning)[45], a transfer learning method developed for any NLP classification tasks. The original implementation of ULMFiT breaks the training process into three stages:

1. Train a general domain language model in the self-supervised manner on a large corpus (e.g., Wikitext-103[59]). Language models are a type of model that aim to predict the next word in the sentences given the context precede it. The input and labels of the dataset used to train a language model are provided by the text itself. After training on millions of unlabeled text, the language model captures the extensive and in-depth knowledge[60–62] of a language and can provide useful features for other NLP tasks.
2. Fine-tuning the general language model on the task corpus to create a task specific language model.
3. Fine-tuning the task specific language model for downstream classification/regression model.

As described above, the ULMFiT is a three-stage transfer learning process that includes two types of models: language models and classification/regression models. A language model is a model that takes in a sequence of words and predicts the most likely next word. A language model is trained in a self-supervised manner and no label is required. This means the training data can be generated from a huge amount of unlabeled text data. The classification/regression model is a model that takes a whole sequence and predicts the class/value associated to the sequence, requiring labeled data.

2.2. MolPMoFiT

In this study, we adapted the ULMFiT method to handle molecular property/activity prediction. Specifically, we trained a **molecular structure prediction model (MSPM)** using one million molecules extracted from ChEMBL with self-supervised learning. The pre-trained MSPM was then fine-tuned for the given QSAR/QSPR tasks.

Model Architecture: The architectures for the MSPMs and the QSAR/QSPR models follow similar structures (**Figure 1**): the embedding layer, the encoder and the classifier. The embedding layer converts the numericized tokens into fixed length vector representations (see **Section 2.4** for details); the encoder processes the sequence of embedding vectors into feature representations which contain the *contextualized* token meanings; and the classifier uses the extracted feature representations to make the final prediction. The model architecture used for modeling is AWD-LSTM (ASGD Weight-Dropped LSTM).[63] The main idea of the AWD-LSTM is to use a LSTM (Long Short-Term Memory[64]) model with dropouts in all the possible layers (embedding layer, input layer, weights, and hidden layers). The model hyperparameters are same as the ones initially implemented for ULMFiT. An embedding vector length of 400 was used for the models. The encoder consisted of three LSTM layers: the input size of first LSTM layer is 400, the hidden number of hidden units is 1152, and the output size of the last LSTM layer is 400. The classifiers use the output of the encoder to make predictions. The MSPMs and QSPR/QSAR models use the output of the encoder in different ways for different prediction purposes. The MSPM classifier consists of just a single softmax layer. The MSPMs predict the next token in a SMILES string, using the hidden state at the last time step h_T of the final LSTM layer of the encoder. The QSPR/QSAR model classifier consists of two feedforward neural network layers. The first layer takes the concatenation of output vectors from the last LSTM layer of the encoder

(concatenation of max pooling, mean pooling and last time step h_T[45]), followed by a ReLU activation function. The final output size is determined by the QSPR/QSAR endpoints, *e.g.*, for regression models, a single output node is used; for classification models, the output size equals to the number of classes.

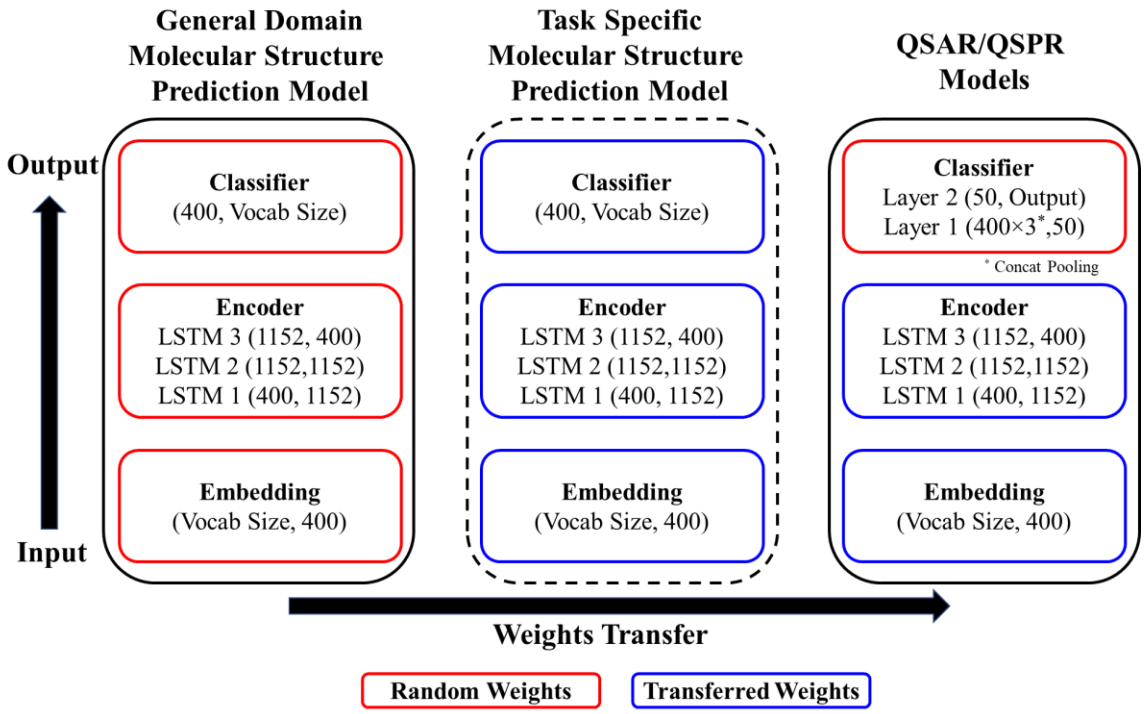


Figure 1. Scheme illustrating the **MolPMoFiT** Architecture: During the fine-tuning, learned weights are transferred between models. Vocab Size corresponds to the number of unique characters tokenized (See **Section 2.4**) from SMILES in a data set. The stage of Task Specific Molecular Structure Prediction Model fine-tuning is optional.

General-Domain MSPM Training: In the first stage of training, a general domain MSPM is trained on one million molecules curated from ChEMBL. The model is trained using the one cycle policy with a constant learning rate for 10 epochs. One cycle policy is a learning rate schedule method proposed by Smith[65]. The MSPM forms the source for all the subsequent QSPR/QSAR models. The training of the general-domain MSPM model requires about one day on a single NVIDIA Quadro P4000 GPU but it only needs to be trained once and can be reused for other QSPR/QSAR tasks.

Task Specific MSPM Model Fine-Tuning (Optional): The stage is optional for MolPMoFiT. The MSPM trained on ChEMBL covers a large and diverse chemical space of bioactive molecules and can be directly fine-tuned to predict physical properties of molecules such as lipophilicity and solubility. For bioactivities such as HIV inhibition and other drug activities, scientists are more interested in compounds with desired activities. The experimental tested data (target task dataset) may have a different distribution from ChEMBL. Fine-tuning the general domain MSPM on target task data to adapt to the idiosyncrasies of the task data would be helpful to the downstream QSAR models. The impact of task specific MSPM fine-tuning will be analyzed in **Section 3.1**.

In this stage, the goal is to fine-tuning the general domain MSPM on the target QSAR datasets to create the task-specific (endpoint-specific) MSPM. The initial weights (embedding, encoder and linear head) of task specific MSPM are transferred from the general domain MSPM. The task specific MSPMs are fine-tuned using the one cycle policy and discriminative fine-tuning[45]. In a neural network, different layers encode different levels of information[66]. Higher layers contain less general knowledge toward the target task and need more fine-tuning compared to lower layers. Instead of using the same learning rate for fine-tuning all the layers, the discriminative fine-tuning trains higher layers with higher learning rates. Learning rates are adjusted based on the same the function $\eta^{layer-1} = \eta^{layer} / 2.6$ used in the original ULMFiT approach, where η is the learning rate.

QSAR/QSPR Models Fine-Tuning: When fine-tuning the QSAR/QSPR model, only the embedding layer and the encoder are transferred from the pre-trained model, as the QSAR/QSPR model required a different classifier. In other word, the weights of classifier are initialized randomly and need to be trained from scratch for each task.[45] The QSPR/QSAR model is fine-

tuned using one cycle policy, discriminative fine-tuning and gradual unfreezing[45]. During the fine-tuning, the model is gradually unfrozen over four layer-groups: (i) classifier; (ii) classifier + final LSTM layer; (iii) classifier + final two LSTM layers, and (iv) full model. Gradual unfreezing first trains the classifier of the model with the embedding and encoder layers frozen (weights are not updated). Then unfreezing the second to last layer-groups and fine-tuning the model. This process continues until all the layer-groups are unfrozen and fine-tuned.

Implementation. We implemented our model using the PyTorch[67] (https://pytorch.org/) deep learning framework and fastai v1 library[68] (https://docs.fast.ai). To ensure the reproducibility of this study, the data and code used in this study are freely available at: https://github.com/XinhaoLi74/MolPMoFiT.

2.3. Dataset preparation

SMILES of all molecules in ChEMBL[48] were downloaded and curated following the procedure: (1) Removing mixtures, molecules with more than 50 heavy atoms (2) Standardizing with MolVS[69] package; (3) Sanitizing and canonizing with RDKit[70] package. After curation, one million SMILES were randomly selected for training and testing the molecular structure perdition model.

We tested our method on four publicly-available, benchmark datasets[15]: (1) molecular lipophilicity; (2) experimental measured solvation energy in kcal/mol (FreeSolv) (3) HIV inhibition, and (4) blood-brain barrier penetration (BBBP). The detailed descriptions are summarized in **Table 1**.

Table 1. Description of QSAR/QSPR datasets.

Data Set	Description	Size	# of Active Compound	Task
Lipophilicity	Octanol/water distribution coefficient	4,200		*Regression*
FreeSolv	Experimental measured solvation energy (kcal/mol)	642		*Regression*
HIV	Inhibition of HIV replication	41,127	1,443	*Classification*
BBBP	Ability to penetrate the blood-brain barrier	2,039	1,560	*Classification*

2.4. Molecular Representation

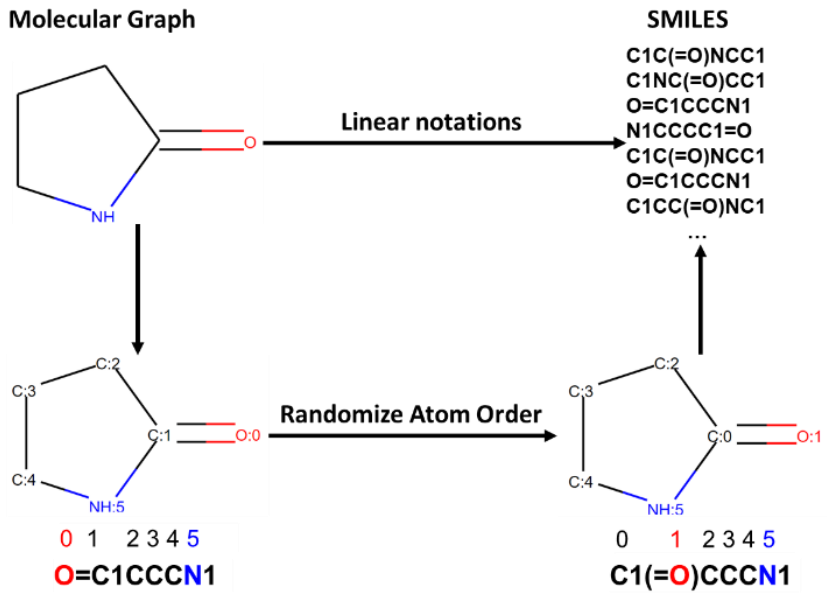
In this study, we use SMILES strings as the textual representation of molecules. SMILES is a linear notation for representing molecular structures. For SMILES to be processed by machine learning models, they need to be transformed into numeric representations. SMILES strings are tokenized at the character level with a few specific treatments: (1) 'Cl', 'Br' are two-character tokens; (2) special characters encoded between brackets are considered as tokens (e.g., '[nH], '[O-]' and '[Te]' et.al). The unique tokens are mapped to integers to be used as input for the deep learning models.

2.5. Data Augmentation

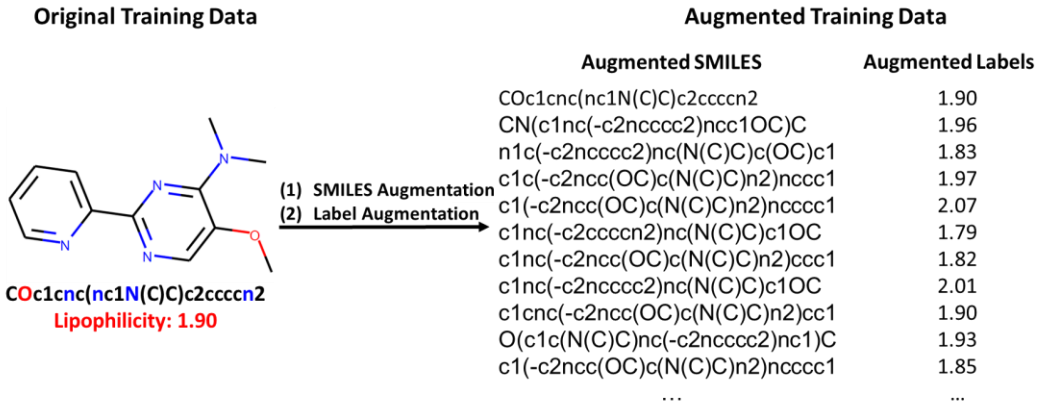
Deep learning models are data-hungry so that various data augmentation techniques have been developed for different types of data and applications[71–74]. Data augmentation usually helps deep learning models to be better generalized for new data. Each SMILES corresponds to one unique molecular structure, whereas several SMILES strings can be derived from the same molecule. In fact, for a single molecular structure, many SMILES can be generated by simply randomizing the atom ordering (**Figure 2a**). Bjerrum shown the SMILES enumeration as a data augmentation technique for QSAR models based on SMILES input can improve the robustness and accuracy[75]. It has been also shown that the generative models trained on both augmented

and canonical SMILES can create a larger chemical space of structures[76, 77]. Herein, we used SMILES enumeration as the basis for data augmentation technique. The SMILES augmentation technique was applied to both the MSPM and QSAR/QSPR models. For MSPM, the SMILES augmentation ensures the trained model can cover a large and diverse chemical space (characterized by SMILES). For unbalanced classification QSAR/QSPR datasets, the SMILES augmentation can be applied to re-balance the class distribution of training data. In addition to SMILES augmentation, for regression QSAR/QSPR models, a Gaussian noise (mean set at 0 and standard deviation $\sigma_{noise)}$ is added to the labels of augmented SMILES which could be considered as a simulation of experimental errors [78]. (**Figure 2b**). The standard deviation σ_{noise} is considered as a hyperparameter for the models and need to be tuned from task to task. The impact of training data augmentation will be analyzed in **Section 3.2**.

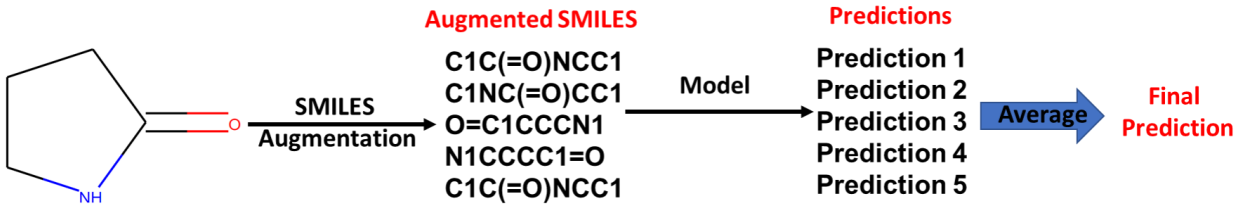
We also applied the test time augmentation (**TTA**): Briefly, the final predictions are generated by averaging predictions of the canonical SMILES and four augmented SMILES (**Figure 2c**). The impact of TTA will be discussed in **Section 3.1**.



(**a**) SMILES Augmentation



(**b**) Training Data Augmentation (Training sets)



(c) Test-Time Augmentation (TTA)

Figure 2. SMILES and Data Augmentation.

2.6. Baselines and Comparison Models

To evaluate the performance of our method, we compared our models to the models reported by Yang et al.[12], including directed message passing neural network (D-MPNN), D-MPNN with RDKit features, random forest (RF) model on binary Morgan fingerprints, feed-forward network (FFN) on binary Morgan fingerprints, FFN on count-based Morgan fingerprints and FFN on RDKit descriptors,. We evaluated all models based on the original random and scaffold splits from Yang et. al. for a fair and reproducible comparison. All the models were evaluated on the test sets on 10 randomly seeded 80:10:10 data splits. For regression model, we use root-mean-square-error (RMSE) as the key metric. For classification model, we use area under the receiver operating characteristic curve (AUROC) as the key metric.

2.7. Hyperparameters and Training Procedure

QSAR/QSPR Model Fine-Tuning: We are interested in obtaining a model that perform robustly across a variety of QSPR/QSAR tasks. Herein, we used the same set of hyperparameters for fine-tuning QSPR/QSAR models across different tasks, which we tuned on the HIV dataset (**Table 2**). The batch size is set to 128 (64 for HIV dataset due to the GPU memory limit). The optimal hyperparameters of the HIV dataset was determined based on the validation set results on 3 randomly 80:10:10 data splits. Specifically, we optimized the dropout rates, the base learning rate and training epochs.

Table 2. Hyperparameters for QSPR/QSAR Model Fine-tuning.

Layer Groups	Base Learning Rate	Epochs
Linear head only	$3e^{-2}$	4
Linear head + final LSTM layer	$5e^{-3}$	4
Linear head + final two LSTM layers	$5e^{-4}$	4
Full Model	$5e^{-5}$	6

Data Augmentation: In order to train a molecular structure prediction model that can be applied to a large chemical space, ChEMBL data is augmented by 4 times in addition to the original canonical SMILES. For the lipophilicity and FreeSolv datasets (regression), the number of augmented SMILES and the label noise σ_{noise} were tuned on the validation set on three 80:10:10 random split. Specifically, the SMILES of lipophilicity training data were augmented 25 times with the label noise $\sigma_{noise} = 0.3$ and the SMILES of FreeSolv training data were augmented 50 times with the label noise $\sigma_{noise} = 0.5$. For classification tasks, we used data augmentation to balance the class distribution. Specifically, for HIV data, the SMILES of active class were augmented 60 times and the SMILES of inactive class were augmented 2 times. For BBBP data, the SMILES of positive class were augmented 10 times and the SMILES of negative class were augmented 30 times.

3. RESULTS AND DISCUSSION

3.1. Benchmark

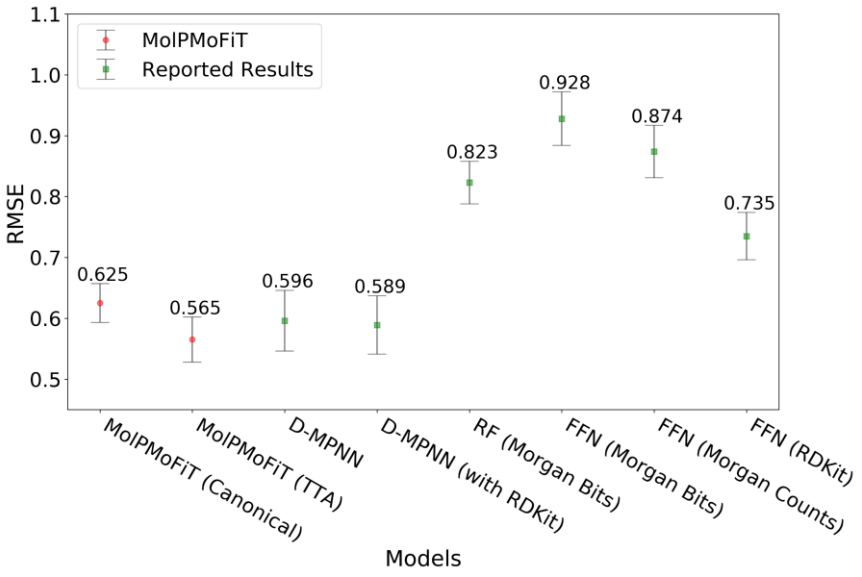
Yang et. al. [12] developed a graph convolutional model based on directed message passing neural network (D-MPNN) and benchmarked it across a wide variety of public and proprietary datasets, achieved consistently strong performance. We benchmarked our MolPMoFiT method to the *state-of-the-art* models from Yang et. al. on four well-studied chemical datasets: lipophilicity, FreeSolv, HIV and BBBP. Both random and scaffold splits were evaluated. Scaffold split enforced all training and test sets shared no common molecular scaffolds, which represent a more challenging and realistic evaluation compared to a random split. All the models were evaluated on test set on the exact same ten 80:10:10 splits from Yang et. al. to ensure a fair and reproducible benchmark. Results for lipophilicity and FreeSolv data were evaluated by root mean square error (RMSE), whereas results for HIV and BBBP were evaluated by area under the receiver operating

characteristic curve (AUROC). For physical properties lipophilicity and FreeSolv data, the regression models were fine-tuned on the general domain MSPM. For bioactivities HIV and BBBP data, the classification models were fine-tuned on both the general and task-specific MSPMs (See **Section 2.2**). Evaluation metrics were computed in two settings: (1) testing on canonical SMILES only and (2) Time-time augmentation (TTA, See **Section 2.5**).

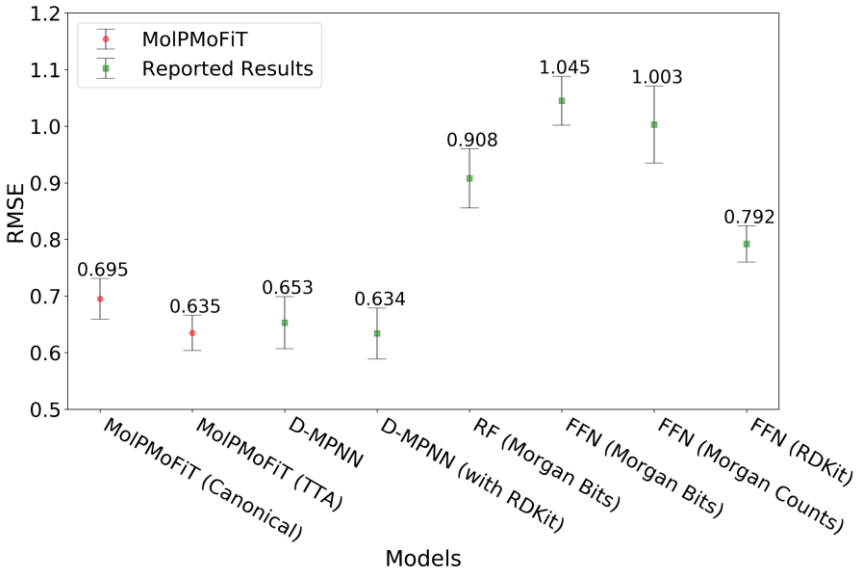
The results for test sets are summarized in **Figures 3-6**. Across all four data sets, MolPMoFiT models achieved comparable or better prediction performances compared to the baselines. Generally, a scaffold split resulted in a worse performance compared to a random split. But a scaffold split can better measure the generalization ability of a model, which is very useful[79] for new molecular series with scaffolds being dissimilar to any other compounds in the modeling set.

For lipophilicity data, MolPMoFiT models tested on TTA outperform those tested on canonical SMILES. On random split, MolPMoFiT achieved a test set RMSE of 0.565±0.037 and 0.625±0.032 with and without TTA, respectively (**Figure 3a**). On scaffold split, MolPMoFiT achieved a test set RMSE of 0.635±0.031 and 0.695±0.036 with and without TTA, respectively (**Figure 3b**).

The FreeSolv dataset only contains 642 compounds, different data splits resulted in a large variance in RMSE (**Figure 4**). MolPMoFiT models tested on TTA outperform those tested on canonical SMILES on random split but have no significant difference on scaffold split. On random split, MolPMoFiT achieved a test set RMSE of 1.197±0.127 and 1.338±0.144 with and without TTA, respectively (**Figure 4a**). On scaffold split, MolPMoFiT achieved a test set RMSE of 2.082±0.460 and 2.185±0.448 with and without TTA, respectively (**Figure 4b**).

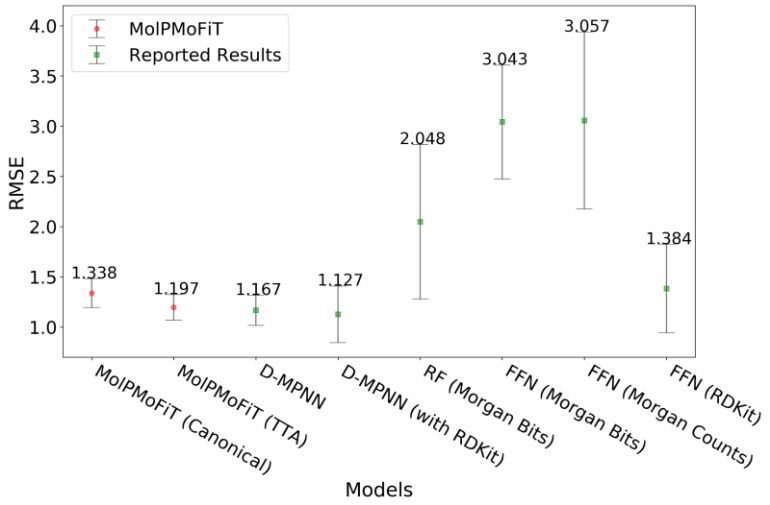


(a) Random Split

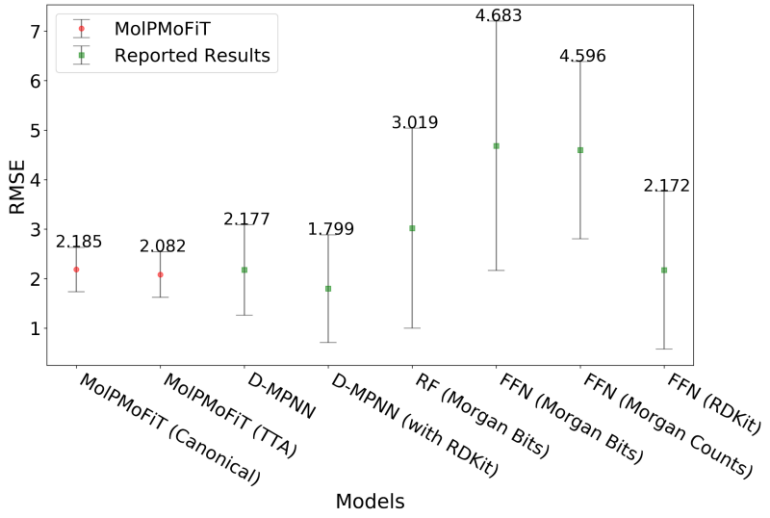


(b) Scaffold Split

Figure 3. Comparison of MolPMoFiT to Reported Results from Yang's[12] on Lipophilicity. (**a**) Random split; (**b**) Scaffold split. MolPMoFiT: Molecular Prediction Model Fine-Tuning; D-MPNN: Directed Message Passing Neural Network; RF: Random Forest; FFN: Feed-Forward Network.



(a) Random Split



(b) Scaffold Split

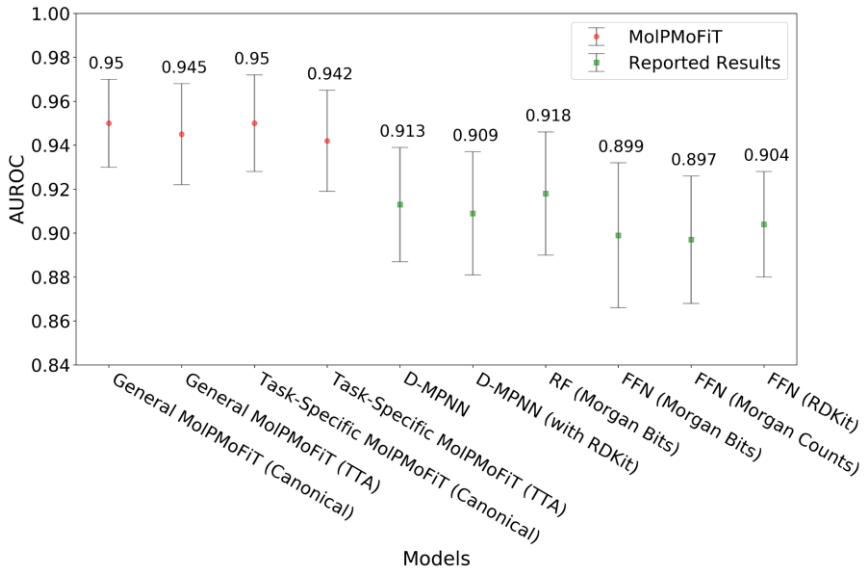
Figure 4. Comparison of MolPMoFiT to Reported Results from Yang's[12] on FreeSolv. (**a**) Random split; (**b**) Scaffold split.

For bioactivities like BBBP and HIV inhibition, the molecules of interest (tested experimentally) may have a different distribution from ChEMBL. Fine-tuning the general domain MSPM on target task data to adapt to the idiosyncrasies of the task data would be helpful to the downstream QSAR models. We evaluated the QSAR models fine-tuned both on general domain MSPM (named as general MolPMoFiT) and task-specific MSPM (named as task-specific MolPMoFiT) on BBBP and HIV datasets. For both BBBP (**Figure 5**) and HIV (**Figure 6**) dataset, the performance of models fine-tuned on general domain MSPM is on-par with the performance

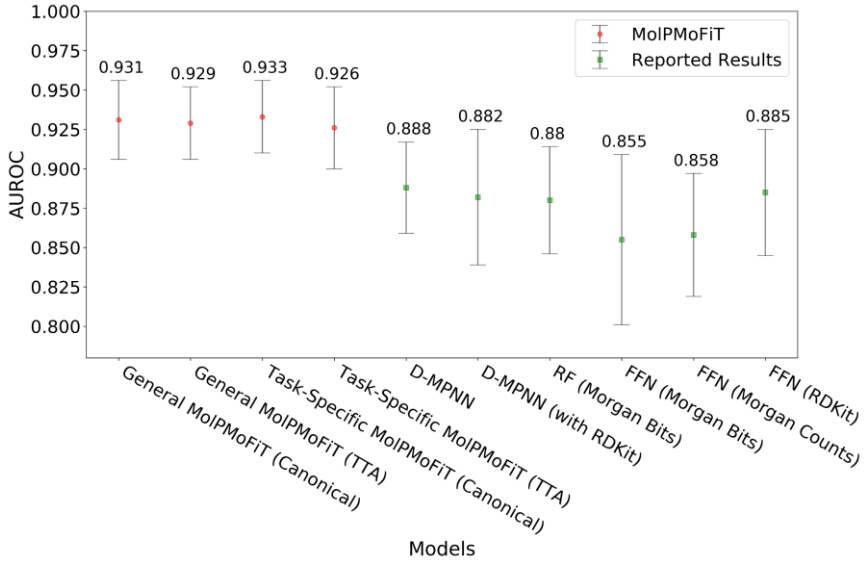
of models fine-tuned on the task-specific MSPMs. It requires more case studies to show whether fine-tuning on task-specific MSPM is beneficial.

On BBBP dataset, MolPMoFiT models outperform other comparison models (**Figure 5**). Specifically, the general MolPMoFiT models achieved a test set AUROC of 0.950±0.020 (Canonical SMILES) and 0.945±0.023 (TTA) on random split and achieved a test set AUROC of 0.931±0.025 (Canonical SMILES) and 0.929±0.023 (TTA) on scaffold split. The task-specific MolPMoFiT models achieved a test set AUROC of 0.950±0.022 (Canonical SMILES) and 0.942±0.023 (TTA) on random split and achieved a test set AUROC of 0.933±0.023 (Canonical SMILES) and 0.926±0.026 (TTA) on scaffold split. It is worth noting that the implementation of TTA shows no improvements of the model accuracy.

For HIV data (**Figure 6**), the general MolPMoFiT models achieved a test set AUROC of 0.801±0.032 (Canonical SMILES) and 0.828±0.029 (TTA) on random split and achieved a test set AUROC of 0.794±0.023 (Canonical SMILES) and 0.816±0.022 (TTA) on scaffold split. The task-specific MolPMoFiT models achieved a test set AUROC of 0.811±0.021 (Canonical SMILES) and 0.834±0.025 (TTA) on random split and achieved a test set AUROC of 0.782±0.018 (Canonical SMILES) and 0.805±0.014 (TTA) on scaffold split.

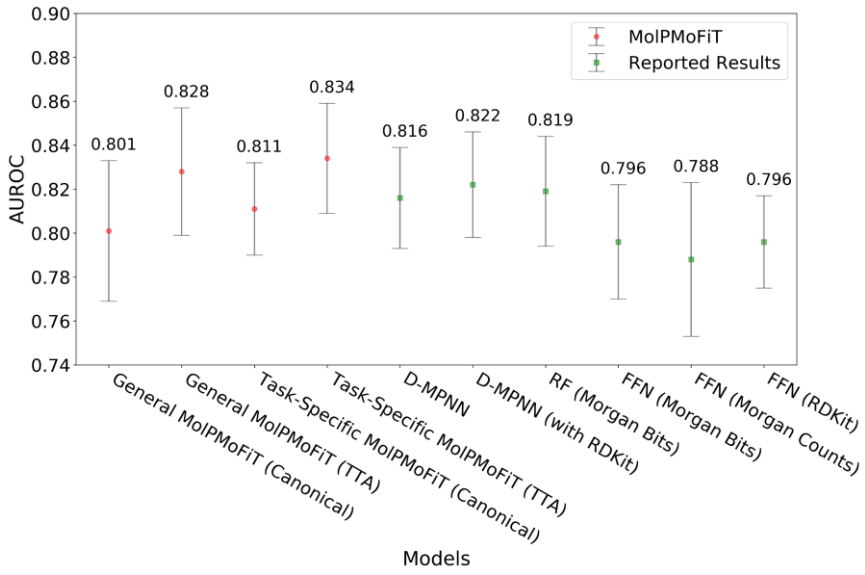


(a) Random Split

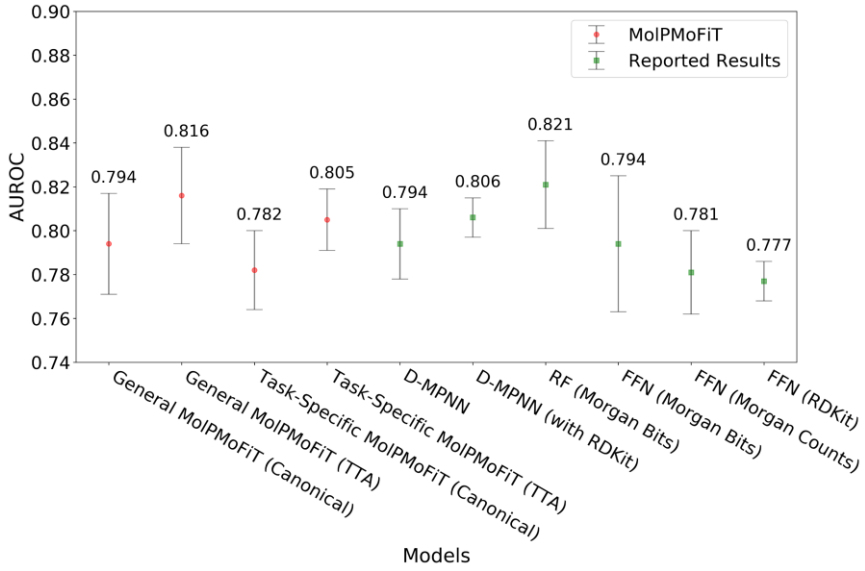


(b) Scaffold Split

Figure 5. Comparison of MolPMoFiT to Reported Results from Yang's[12] on BBBP. (**a**) Random split; (**b**) Scaffold split.



(a) Random Split



(b) Scaffold Split

Figure 6. Comparison of MolPMoFiT to Reported Results from Yang's[12] on HIV. (**a**) Random split; (**b**) Scaffold split.

3.2. Analysis

Impact of Transfer Learning: MolPMoFiT models were compared to the models that were trained from scratch. The models were trained on different number of training data and tested on the test set on a single 80:10:10 random split. The hyperparameters (learning rate, dropout rate and training epochs) were kept fixed: the hyperparameters of MolPMoFiT models were the same

as we used in benchmark and the hyperparameters of models trained from scratch were tuned based on the validation set using the full training set. The results are illustrated in **Figure 7**. Generally, with different numbers of training data, the MolPMoFiT model always outperforms the model trained from scratch. This indicated that the MolPMoFiT transfer learning technique provided a robust improvement for the model performances.

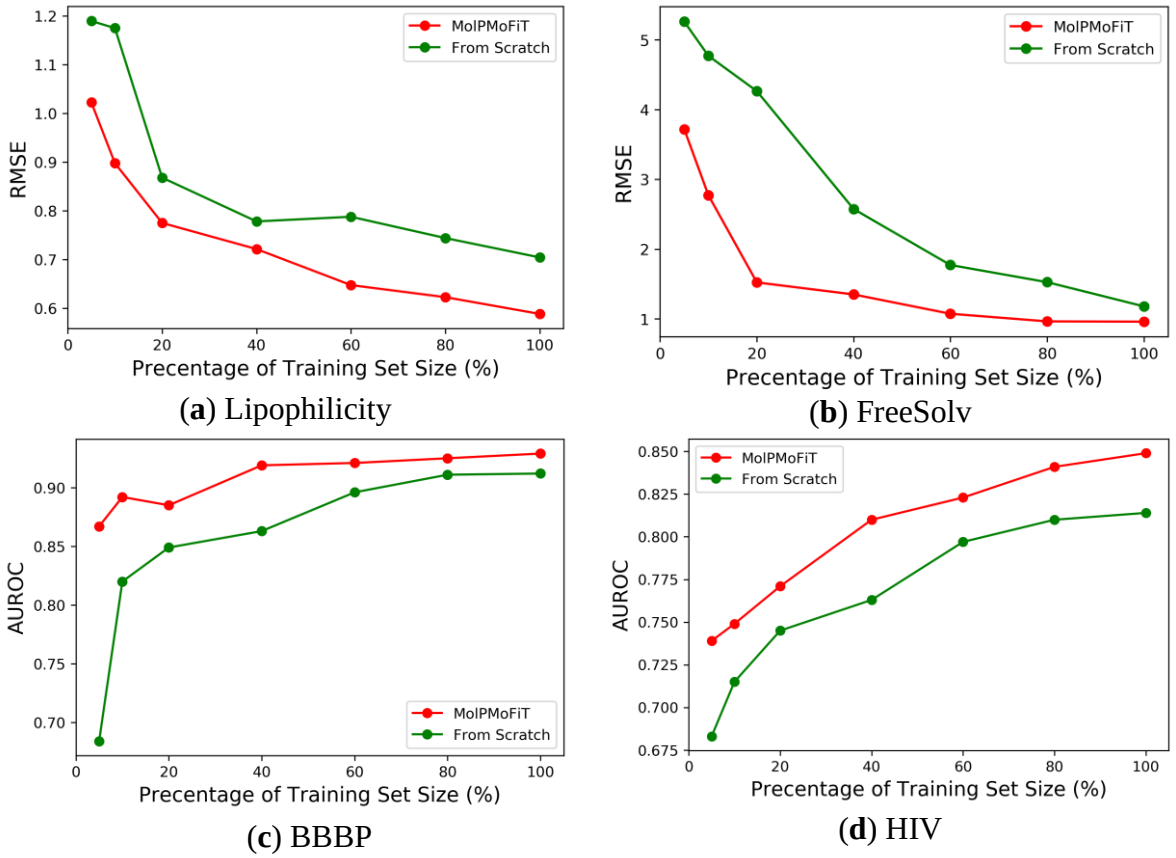


Figure 7. Performances of models on the different size of the training set. (**a**) Lipophilicity; (**b**) FreeSolv; (**c**) BBBP and (**d**) HIV.

Impact of Training Data Augmentation: In **Section 3.1**, we shown that test-time augmentation (TTA) can improve the accuracy of predictions. Herein, we analyze the effect of training data augmentation. All models were evaluated with evaluation metrics computed with

TTA on the test sets on three 80:10:10 random splits. The hyperparameters of models were the same as we used in benchmark.

For classification tasks (BBBP and HIV), models were trained on different sizes of augmented training data. On HIV dataset, when model trained on the original dataset (no augmentation), the AUROC is 0.816±0.005, which is significantly low than those with data augmentation. The models achieved similar performance when data augmentation applied no matter class re-balancing or not (**Table 3**). Similarly, training data augmentation significantly improves the accuracy of the model on BBBP data. The model shows no improvement with the class re-balancing (**Table 4**).

Table 3. Impact of SMILES Augmentation on HIV Dataset.

Iteration of Augmentation		**Class Ratio (Positive: Negative)**	**AUROC**
Positive Class	**Negative Class**		
0	0	0.037	0.816±0.005
4	4	0.037	**0.831±0.003**
30	1	0.52	**0.830±0.007**
60	1	1	**0.835±0.007**

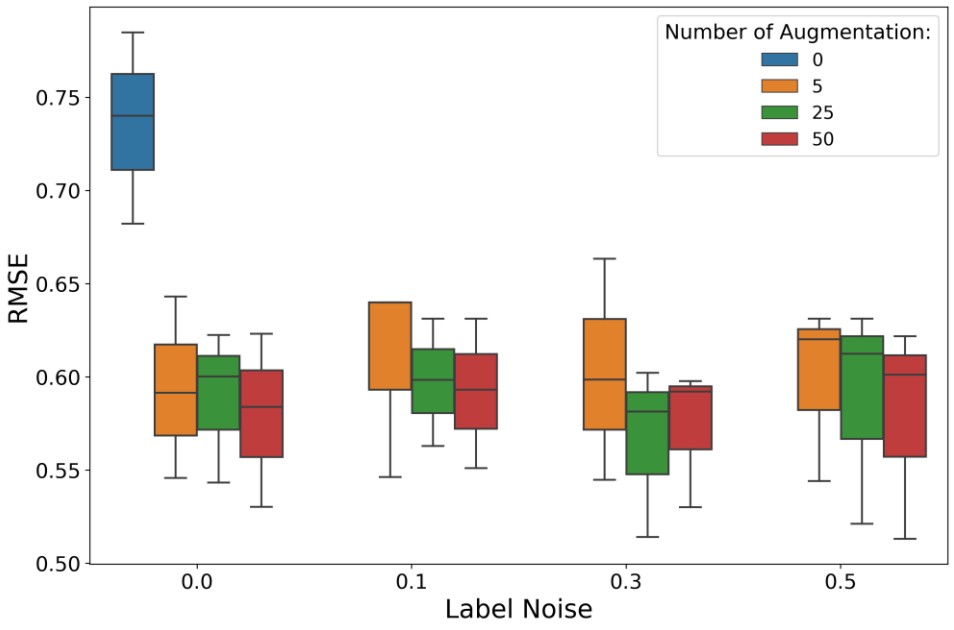
Table 4. Impact of SMILES Augmentation on BBBP Dataset.

Iteration of Augmentation		**Class Ratio (Positive: Negative)**	**AUROC**
Positive Class	**Negative Class**		
0	0	3.38	0.894±0.004
4	4	3.10	0.937±0.005
10	10	3.25	**0.949±0.002**
9	30	1.1	**0.946±0.002**

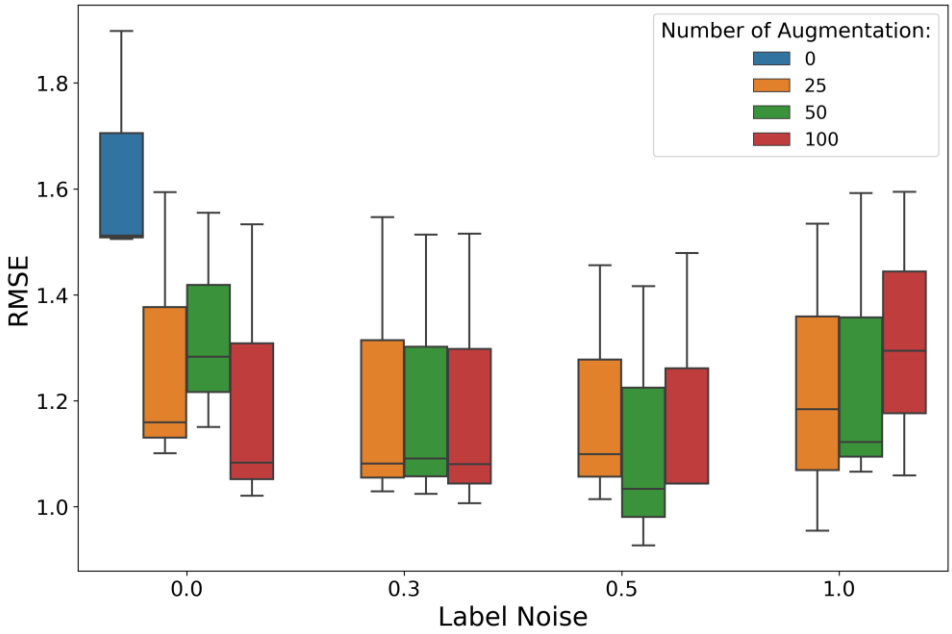
For regression tasks (lipophilicity and FreeSolv), models were trained on different sizes of augmented training data, whose labels were perturbed with different Gaussian noise σ_{noise}. The evaluated numbers of augmented SMILES per compound were {0, 5, 25, 50} and {0, 25, 50, 100} for lipophilicity and FreeSolv, respectively. The evaluated Gaussian noise σ_{noise} values were {0,

0.1, 0.3, 0.5}and {0, 0.3, 0.5, 1} for lipophilicity and FreeSolv, respectively. The results on the test set are shown in **Figure 8**. For both lipophilicity and FreeSolv datasets, when the model was only trained on the original training data (no augmented SMILES and perturbed labels), the performance is significantly worse than those of the models trained on augmented training data.

The results above show one limitation of using SMILES as input for deep learning model: the model actually learns to map individual SMILES to molecular properties/activities instead of linking actual molecular structures to their properties/activities. However, the SMILES augmentation is used as a regularization technique, making the model more robust to various SMILES representation for the same molecule. Appropriately adding random label noise to the augmented SMILES led to improved predictive power of the regression model. For the same data augmentation setting, testing results with TTA were found to be almost always better than the results on only canonical SMILES. While augmentation for training set can help in building models that can generalize well on new data, prediction accuracy can be further improved by TTA.



(a) Lipophilicity



(b) FreeSolv

Figure 8. Performances of Lipophilicity models on different number of augmented SMILES per compound and Gaussian Noise (σ_{noise}) added to the original experimental values. TTA: Test-time augmentation.

4. Conclusions

In this study, we introduced the MolPMoFiT, a novel transfer learning method for QSPR/QSAR tasks. We pre-trained a molecular structure prediction model (MSPM) using one million bioactive molecules from ChEMBL and then fine-tuned it for various QSPR/QSAR tasks. This pre-training + fine-tuning approach enables knowledge learned from large chemical data sets to transfer to smaller data sets, thereby improving the model performance and generalization. Without endpoint-specific hyperparameter tuning, this method showed comparable or better results compared to that of the *state-of-the-art* results reported in the literature for four benchmark datasets. In addition to the strong *out-of-box* performance, this method reuses the pre-trained MSMP across QSPR/QSAR tasks so that reduces the burden of hyperparameters tuning and model training. We posit that transfer learning techniques such as MolPMoFiT could significantly

contribute in boosting the reliability of next-generation QSPR/QSAR models, especially for small/medium size datasets that are extremely challenging for QSAR modeling.

List of Abbreviations

MolPMoFiT: **Mol**ecular **P**rediction **Mo**del **Fi**ne-**T**uning; QSPR/QSAR: Quantitative structure property/activity relationship; SMILES: Simplified Molecular-Input Line-Entry System; GCNN: Graph convolutional neural networks; RNN: Recurrent neural network; CNN: convolutional neural network; CV: computer vision; NLP: natural language processing; MLT: multitask learning; MSPM: molecular structure prediction model; ULMFiT: **U**niversal **L**anguage **M**odel **Fi**ne-**T**uning; AWD-LSTM: ASGD Weight-Dropped LSTM; D-MPNN: Directed Message Passing Neural Network; RF: Random Forest; FFN: Feed-Forward Network; TTA: Test-time augmentation

Availability of data and materials

The curated datasets (.smi and .csv files) and the full updated code used in this study are freely-available at: https://github.com/XinhaoLi74/MolPMoFiT.

Competing interests

The authors declare no competing financial interest

Funding

We gratefully thank the financial support from DARPA and ARO (grant number W911NF-18-1-0315).

Authors' Contributions

XL conceived, developed and implemented the method, performed the analysis, and wrote the manuscript. DF conceived the method and wrote the manuscript.

Acknowledgements

We gratefully thank the financial support from DARPA and ARO (grant number W911NF-18-1-0315).

References

1. Cherkasov A, Muratov EN, Fourches D, et al (2014) QSAR modeling: Where have you been? Where are you going to? J Med Chem 57:4977–5010. https://doi.org/10.1021/jm4004285
2. Mater AC, Coote ML (2019) Deep Learning in Chemistry. J Chem Inf Model 59:2545–2559. https://doi.org/10.1021/acs.jcim.9b00266
3. Tropsha A (2010) Best Practices for QSAR Model Development, Validation, and Exploitation. Mol Inform 29:476–488. https://doi.org/10.1002/minf.201000061
4. Ma J, Sheridan RP, Liaw A, et al (2015) Deep Neural Nets as a Method for Quantitative Structure–Activity Relationships. J Chem Inf Model 55:263–274. https://doi.org/10.1021/ci500747n
5. Fourches D, Williams AJ, Patlewicz G, et al (2018) Computational Tools for ADMET Profiling. In: Computational Toxicology. pp 211–244
6. Li X, Kleinstreuer NC, Fourches D (2020) Hierarchical Quantitative Structure–Activity Relationship Modeling Approach for Integrating Binary, Multiclass, and Regression Models of Acute Oral Systemic Toxicity. Chem Res Toxicol 33:353–366. https://doi.org/10.1021/acs.chemrestox.9b00259
7. Ash J, Fourches D (2017) Characterizing the Chemical Space of ERK2 Kinase Inhibitors Using Descriptors Computed from Molecular Dynamics Trajectories. J Chem Inf Model 57:1286–1299. https://doi.org/10.1021/acs.jcim.7b00048
8. Fourches D, Ash J (2019) 4D- quantitative structure–activity relationship modeling: making a comeback. Expert Opin Drug Discov 1–9. https://doi.org/10.1080/17460441.2019.1664467
9. Xue L, Bajorath J (2012) Molecular Descriptors in Chemoinformatics, Computational Combinatorial Chemistry, and Virtual Screening. Comb Chem High Throughput Screen

3:363–372. https://doi.org/10.2174/1386207003331454

10. Gilmer J, Schoenholz SS, Riley PF, et al (2017) Neural Message Passing for Quantum Chemistry. http://arxiv.org/abs/1704.01212
11. Chen C, Ye W, Zuo Y, et al (2019) Graph Networks as a Universal Machine Learning Framework for Molecules and Crystals. Chem Mater 31:3564–3572. https://doi.org/10.1021/acs.chemmater.9b01294
12. Yang K, Swanson K, Jin W, et al (2019) Analyzing Learned Molecular Representations for Property Prediction. J Chem Inf Model 59:3370–3388. https://doi.org/10.1021/acs.jcim.9b00237
13. Duvenaud D, Maclaurin D, Aguilera-Iparraguirre J, et al (2015) Convolutional Networks on Graphs for Learning Molecular Fingerprints. Adv Neural Inf Process Syst 2015-Janua:2224–2232
14. Coley CW, Barzilay R, Green WH, et al (2017) Convolutional Embedding of Attributed Molecular Graphs for Physical Property Prediction. J Chem Inf Model 57:1757–1772. https://doi.org/10.1021/acs.jcim.6b00601
15. Wu Z, Ramsundar B, Feinberg EN, et al (2018) MoleculeNet: a benchmark for molecular machine learning. Chem Sci 9:513–530. https://doi.org/10.1039/C7SC02664A
16. Pham T, Tran T, Venkatesh S (2018) Graph Memory Networks for Molecular Activity Prediction. In: Proceedings - International Conference on Pattern Recognition. pp 639–644
17. Wang X, Li Z, Jiang M, et al (2019) Molecule Property Prediction Based on Spatial Graph Embedding. J Chem Inf Model acs.jcim.9b00410. https://doi.org/10.1021/acs.jcim.9b00410
18. Feinberg EN, Sur D, Wu Z, et al (2018) PotentialNet for Molecular Property Prediction. ACS Cent Sci 4:1520–1530. https://doi.org/10.1021/acscentsci.8b00507
19. Stokes JM, Yang K, Swanson K, et al (2020) A Deep Learning Approach to Antibiotic Discovery. Cell 180:688-702.e13. https://doi.org/10.1016/j.cell.2020.01.021
20. Tang B, Kramer ST, Fang M, et al (2020) A self-attention based message passing neural network for predicting molecular lipophilicity and aqueous solubility. J Cheminform 12:15. https://doi.org/10.1186/s13321-020-0414-z
21. Withnall M, Lindelöf E, Engkvist O, Chen H (2020) Building attention and edge message

passing neural networks for bioactivity and physical-chemical property prediction. J Cheminform 12:1–18. https://doi.org/10.1186/s13321-019-0407-y

22. Goh GB, Hodas NO, Siegel C, Vishnu A (2017) SMILES2Vec: An Interpretable General-Purpose Deep Neural Network for Predicting Chemical Properties. http://arxiv.org/abs/1712.02034
23. Zheng S, Yan X, Yang Y, Xu J (2019) Identifying Structure–Property Relationships through SMILES Syntax Analysis with Self-Attention Mechanism. J Chem Inf Model 59:914–923. https://doi.org/10.1021/acs.jcim.8b00803
24. Kimber TB, Engelke S, Tetko I V, et al (2018) Synergy Effect between Convolutional Neural Networks and the Multiplicity of SMILES for Improvement of Molecular Prediction. http://arxiv.org/abs/1812.04439
25. Goh GB, Siegel C, Vishnu A, et al (2017) Chemception: A Deep Neural Network with Minimal Chemistry Knowledge Matches the Performance of Expert-developed QSAR/QSPR Models. https://arxiv.org/pdf/1706.06689.pdf
26. Goh GB, Siegel C, Vishnu A, Hodas NO (2017) Using Rule-Based Labels for Weak Supervised Learning: A ChemNet for Transferable Chemical Property Prediction. 9:. https://doi.org/10.475/123
27. Paul A, Jha D, Al-Bahrani R, et al (2018) CheMixNet: Mixed DNN Architectures for Predicting Chemical Properties using Multiple Molecular Representations. http://arxiv.org/abs/1811.08283
28. Goh GB, Siegel C, Vishnu A, et al (2018) How Much Chemistry Does a Deep Neural Network Need to Know to Make Accurate Predictions? In: Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018. pp 1340–1349
29. Fernandez M, Ban F, Woo G, et al (2018) Toxic Colors: The Use of Deep Learning for Predicting Toxicity of Compounds Merely from Their Graphic Images. J Chem Inf Model 58:1533–1543. https://doi.org/10.1021/acs.jcim.8b00338
30. Asilar E, Hemmerich J, Ecker GF (2020) Image Based Liver Toxicity Prediction. J Chem Inf Model acs.jcim.9b00713. https://doi.org/10.1021/acs.jcim.9b00713
31. Varnek A, Fourches D, Hoonakker F, Solov'ev VP (2005) Substructural fragments: an universal language to encode reactions, molecular and supramolecular structures. J Comput Aided Mol Des 19:693–703. https://doi.org/10.1007/s10822-005-9008-0

32. Weininger D (1988) SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. J Chem Inf Model 28:31–36. https://doi.org/10.1021/ci00057a005
33. Weininger D, Weininger A, Weininger JL (1989) SMILES. 2. Algorithm for generation of unique SMILES notation. J Chem Inf Model 29:97–101. https://doi.org/10.1021/ci00062a008
34. Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. Proc Natl Acad Sci 79:2554–2558. https://doi.org/10.1073/pnas.79.8.2554
35. Lipton ZC, Berkowitz J, Elkan C (2015) A Critical Review of Recurrent Neural Networks for Sequence Learning. http://arxiv.org/abs/1506.00019
36. Kim Y Convolutional neural networks for sentence classification. http://arxiv.org/abs/1408.5882
37. Vaswani A, Shazeer N, Parmar N, et al (2017) Attention Is All You Need. http://arxiv.org/abs/1706.03762
38. Deng J, Dong W, Socher R, et al (2009) ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp 248–255
39. Canziani A, Paszke A, Culurciello E (2016) An Analysis of Deep Neural Network Models for Practical Applications. http://arxiv.org/abs/1605.07678
40. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient Estimation of Word Representations in Vector Space. http://arxiv.org/abs/1301.3781
41. Pennington J, Socher R, Manning CD (2014) GloVe: Global Vectors for Word Representation. In: Empirical Methods in Natural Language Processing (EMNLP). pp 1532–1543
42. Joulin A, Grave E, Bojanowski P, et al (2016) FastText.zip: Compressing text classification models. http://arxiv.org/abs/1612.03651
43. Peters ME, Neumann M, Iyyer M, et al (2018) Deep contextualized word representations. http://allennlp.org/elmo
44. Devlin J, Chang M-W, Lee K, Toutanova K (2018) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. http://arxiv.org/abs/1810.04805

45. Howard J, Ruder S (2018) Universal Language Model Fine-tuning for Text Classification. http://arxiv.org/abs/1801.06146
46. Yang Z, Dai Z, Yang Y, et al (2019) XLNet: Generalized Autoregressive Pretraining for Language Understanding. http://arxiv.org/abs/1906.08237
47. Liu Y, Ott M, Goyal N, et al (2019) RoBERTa: A Robustly Optimized BERT Pretraining Approach. http://arxiv.org/abs/1907.11692
48. Gaulton A, Bellis LJ, Bento AP, et al (2012) ChEMBL: A large-scale bioactivity database for drug discovery. Nucleic Acids Res 40:1100–1107. https://doi.org/10.1093/nar/gkr777
49. Jaeger S, Fulle S, Turk S (2018) Mol2vec: Unsupervised Machine Learning Approach with Chemical Intuition. J Chem Inf Model 58:27–35. https://doi.org/10.1021/acs.jcim.7b00616
50. Hu W, Liu B, Gomes J, et al (2019) Pre-training Graph Neural Networks. https://arxiv.org/pdf/1905.12265.pdf
51. Xu Y, Ma J, Liaw A, et al (2017) Demystifying Multitask Deep Neural Networks for Quantitative Structure-Activity Relationships. J Chem Inf Model 57:2490–2504. https://doi.org/10.1021/acs.jcim.7b00087
52. Sosnin S, Karlov D, Tetko I V, Fedorov M V (2019) Comparative Study of Multitask Toxicity Modeling on a Broad Chemical Space. J Chem Inf Model 59:1062–1072. https://doi.org/10.1021/acs.jcim.8b00685
53. de la Vega de León A, Chen B, Gillet VJ (2018) Effect of missing data on multitask prediction methods. J Cheminform 10:26. https://doi.org/10.1186/s13321-018-0281-z
54. Wu K, Wei G-W (2018) Quantitative Toxicity Prediction Using Topology Based Multitask Deep Neural Networks. J Chem Inf Model 58:520–531. https://doi.org/10.1021/acs.jcim.7b00558
55. Varnek A, Gaudin C, Marcou G, et al (2009) Inductive Transfer of Knowledge: Application of Multi-Task Learning and Feature Net Approaches to Model Tissue-Air Partition Coefficients. J Chem Inf Model 49:133–144. https://doi.org/10.1021/ci8002914
56. Ramsundar B, Liu B, Wu Z, et al (2017) Is Multitask Deep Learning Practical for Pharma? J Chem Inf Model 57:2068–2076. https://doi.org/10.1021/acs.jcim.7b00146
57. Wu K, Wei G-W (2018) Quantitative Toxicity Prediction Using Topology Based Multitask Deep Neural Networks. J Chem Inf Model 58:520–531.

https://doi.org/10.1021/acs.jcim.7b00558

58. Varnek A, Gaudin C, Marcou G, et al (2009) Inductive Transfer of Knowledge: Application of Multi-Task Learning and Feature Net Approaches to Model Tissue-Air Partition Coefficients. J Chem Inf Model 49:133–144. https://doi.org/10.1021/ci8002914
59. Merity S, Xiong C, Bradbury J, Socher R (2016) Pointer Sentinel Mixture Models. http://arxiv.org/abs/1609.07843
60. Linzen T, Dupoux E, Goldberg Y (2016) Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. http://arxiv.org/abs/1611.01368
61. Gulordava K, Bojanowski P, Grave E, et al (2018) Colorless green recurrent networks dream hierarchically. http://arxiv.org/abs/1803.11138
62. Radford A, Jozefowicz R, Sutskever I (2017) Learning to Generate Reviews and Discovering Sentiment. http://arxiv.org/abs/1704.01444
63. Merity S, Keskar NS, Socher R (2017) Regularizing and Optimizing LSTM Language Models. http://arxiv.org/abs/1708.02182
64. Hochreiter S, Schmidhuber J (1997) Long Short-Term Memory. Neural Comput 9:1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
65. Smith LN (2018) A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay. http://arxiv.org/abs/1803.09820
66. Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems. pp 3320–3328
67. Adam Paszke; Sam Gross; et al (2017) Automatic differentiation in PyTorch. 31st Conf Neural Inf Process Syst (NIPS 2017)
68. Howard J, Gugger S (2020) Fastai: A Layered API for Deep Learning. Information 11:108. https://doi.org/10.3390/info11020108
69. Swain M MolVS: Molecule Validation and Standardization. https://github.com/mcs07/MolVS
70. Landrum G RDKit: Open-source cheminformatics. http://www.rdkit.org
71. Fadaee M, Bisazza A, Monz C (2017) Data Augmentation for Low-Resource Neural Machine Translation. http://arxiv.org/abs/1705.00440
72. Kobayashi S (2018) Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. In: Proceedings of the 2018 Conference of the North American

Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). Association for Computational Linguistics, Stroudsburg, PA, USA, pp 452–457

73. Kafle K, Yousefhussien M, Kanan C (2017) Data Augmentation for Visual Question Answering. In: Proceedings of the 10th International Conference on Natural Language Generation. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 198–202
74. Lei C, Hu B, Wang D, et al (2019) A preliminary study on data augmentation of deep learning for image classification. In: ACM International Conference Proceeding Series
75. Bjerrum EJ (2017) SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules. http://arxiv.org/abs/1703.07076
76. Arús-Pous J, Blaschke T, Ulander S, et al (2019) Exploring the GDB-13 chemical space using deep generative models. J Cheminform 11:20. https://doi.org/10.1186/s13321-019-0341-z
77. Arús-Pous J, Johansson SV, Prykhodko O, et al (2019) Randomized SMILES strings improve the quality of molecular generative models. J Cheminform 11:71. https://doi.org/10.1186/s13321-019-0393-0
78. Cortes-Ciriano I, Bender A (2015) Improved Chemical Structure–Activity Modeling Through Data Augmentation. J Chem Inf Model 55:2682–2692. https://doi.org/10.1021/acs.jcim.5b00570
79. Sheridan RP (2013) Time-Split Cross-Validation as a Method for Estimating the Goodness of Prospective Prediction. J Chem Inf Model 53:783–790. https://doi.org/10.1021/ci400084k