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

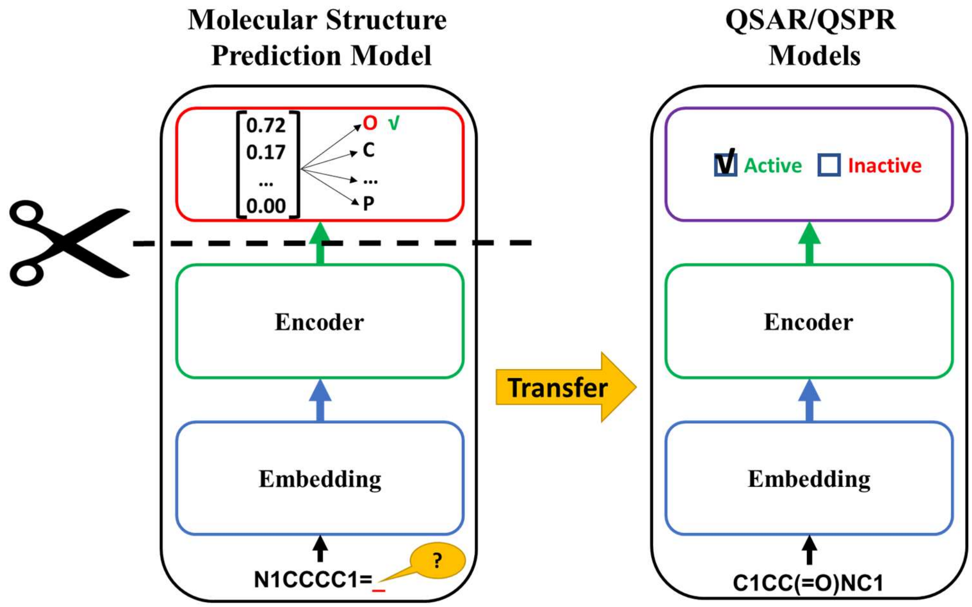
Xinhao Li & Denis Fourches*

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*

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 very large training sets to truly learn the best 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 very 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 that can be applied to any QSPR/QSAR problems. A large-scale molecular structure prediction model is pre-trained using one million unlabeled molecules from ChEMBL in a self-supervised learning manor, and can then be fine-tuned on various QSPR/QSAR tasks for smaller chemical datasets with a specific endpoints. Herein, the method is evaluated on three benchmark datasets (lipophilicity, HIV, and blood-brain barrier penetration). The results showed the method can achieve comparable or better prediction performances on all three datasets compared to *state-of-the-art* prediction techniques reported in the literature so far.



1. Introduction

Predicting the 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[1–5] 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[6,7] directly computed from the molecular structures[8].

On the other hand, deep learning methods have demonstrated remarkable performances in several QSPR/QSAR case studies.[9,10] Those techniques can directly take molecular structures (*e.g.,* molecular graph[9–17] , SMILES strings[18–20], and molecular 2D/3D grid image[21–25]) and learn their own, self-defined 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.

Graph convolutional neural networks (GCNN) directly operate on molecular graphs.[11] 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 but in a more dataset-specific manner). Moreover, the Simplified Molecular-Input Line-Entry System (SMILES)[26,27] encodes the molecular structures as strings of text. Extremely popular 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[28,29], CNNs[30] and transformers[31] 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 learnt 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.

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 the 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[32] can be used as the start point to fine-tune for a new task[33]. 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[34], GloVe[35] or fastText[36]. 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[37–41] extends this approach by transferring all the learnt optimized weights from multiple layers, which providing *contextualized* word embeddings for the downstream tasks.

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[22] for transferable chemical property prediction. A deep neural network was pre-trained in a supervised manor on the ChEMBL[42] database using computed molecular descriptors as labels, then fine-tuned on other QSPR/QSAR tasks. Jaeger et al.[43] developed Mol2vec which employed the same idea of Word2Vec in NLP. Mol2vec learns the vector representations of molecular substructures in an unsupervised learning approach. 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[44]. 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.[45–52] 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**), an effective transfer learning method that can be applied to any QSPR/QSAR problems. 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 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)[39], a transfer learning method developed for any NLP classification tasks. The original implementation of UMLFiT breaks the training process into three stages:

1. Train a general domain language model in the self-supervised manor on a large corpus (e.g., Wikitext-103[53]). Language models are a type of model that aim to predict the next word in the sentences given the context precede it. Tremendous unlabeled text data can be considered as the 'ImageNet' for language modeling. After training on millions of unlabeled text, the language model captures the extensive and in-depth knowledge[54–56] 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 UMLFiT 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 manor 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. UMSPMFiT

In this study, we adapted the UMLFiT method to handle molecular property/activity prediction. Specifically, we trained a **molecular structure prediction model (MSPM)** using one million bioactive molecules extracted from ChEMBL. 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 linear head. The embedding layer converts the numericized tokens into fixed length vector representations (see next section for details); the encoder processes the sequence of embedding vectors into feature representations which contain the *contextualized* token meanings; and the linear head uses the extracted feature representations to make the final prediction. The model architecture used for modeling is AWD-LSTM (ASGD Weight-Dropped LSTM).[57] The main idea of the AWD-LSTM is to use a LSTM (Long Short-Term Memory[58]) model with dropouts in all the possible layers (embedding layer, input layer, weights, and hidden layers). The model parameters are same as the ones initially developed for UMLFiT. 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 linear heads 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 linear head 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 encoder. The QSPR/QSAR model linear head consists of two linear layers. The first linear layer takes the concatenation of ouput vectors from the last LSTM layer of the encoder (concatenation pooling[39]). 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.

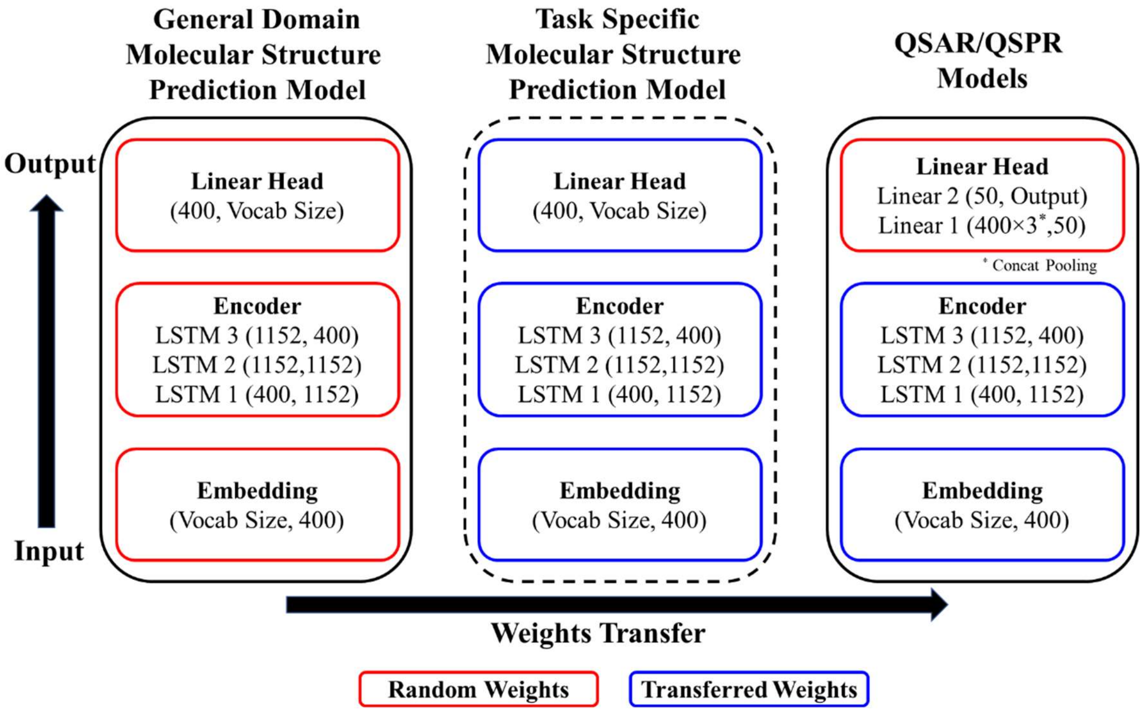


Figure 1. Scheme illustrating the **MolPMoFiT** Architecture: During the fine-tuning, learnt weights are transferred between models. Vocab Size corresponds to the number of unique characters tokenized (See **Section 2.4**) from SMILES in a data set.

General-Domain MSPM Training: In the first stage of training, a general domain MSPM is trained on one million bioactive molecules 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[59]. 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. For QSAR tasks, the target datasets may have a distribution different from ChEMBL dataset (*e.g.,* toxicity data, drug activity data). 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. However, for QSPR tasks like solubility or lipophilicity, there is no actual need to create the task 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[39]. In a neural network, different layers encode different levels of information[60]. 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 UMLFiT approach, where η is the learning rate. The impact of task specific MSPM fine-tuning will be discussed in **Section 3.2**.

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 linear head. In other word, the classification linear head is initialized randomly and needs to be trained from scratch for each task.[39] The QSPR/QSAR model is fine-tuned using one cycle policy, discriminative fine-tuning and gradual unfreezing[39]. During the fine-tuning, the model is gradually unfrozen over four layer-groups: (i) linear head; (ii) linear head + final LSTM layer; (iii) linear head + final two LSTM layers, and (iv) full model. Gradual unfreezing first trains the linear head 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[61] (https://pytorch.org/) deep learning framework and fastai library[62] (https://docs.fast.ai). All the code used in this study will be available at: https://github.com/XinhaoLi74/MolPMoFiT.

2.3. Dataset preparation

SMILES of all bioactive molecules were downloaded from ChEMBL[42] and curated following the procedure: (1) Removing mixtures, molecules with more than 50 heavy atoms (2) Standardizing with MolVS[63] package (https://github.com/mcs07/MolVS); (3) Sanitizing and canonizing with RDKit[64] package (https://github.com/rdkit/rdkit). After curation, one million SMILES were randomly selected for training and testing the molecular structure perdition model.

We tested our method on three publicly-available, benchmark datasets[9]: (1) molecular lipophilicity; (2) HIV inhibition, and (3) blood-brain barrier penetration (BBBP). The detailed descriptions are summarized in **Table 1**.

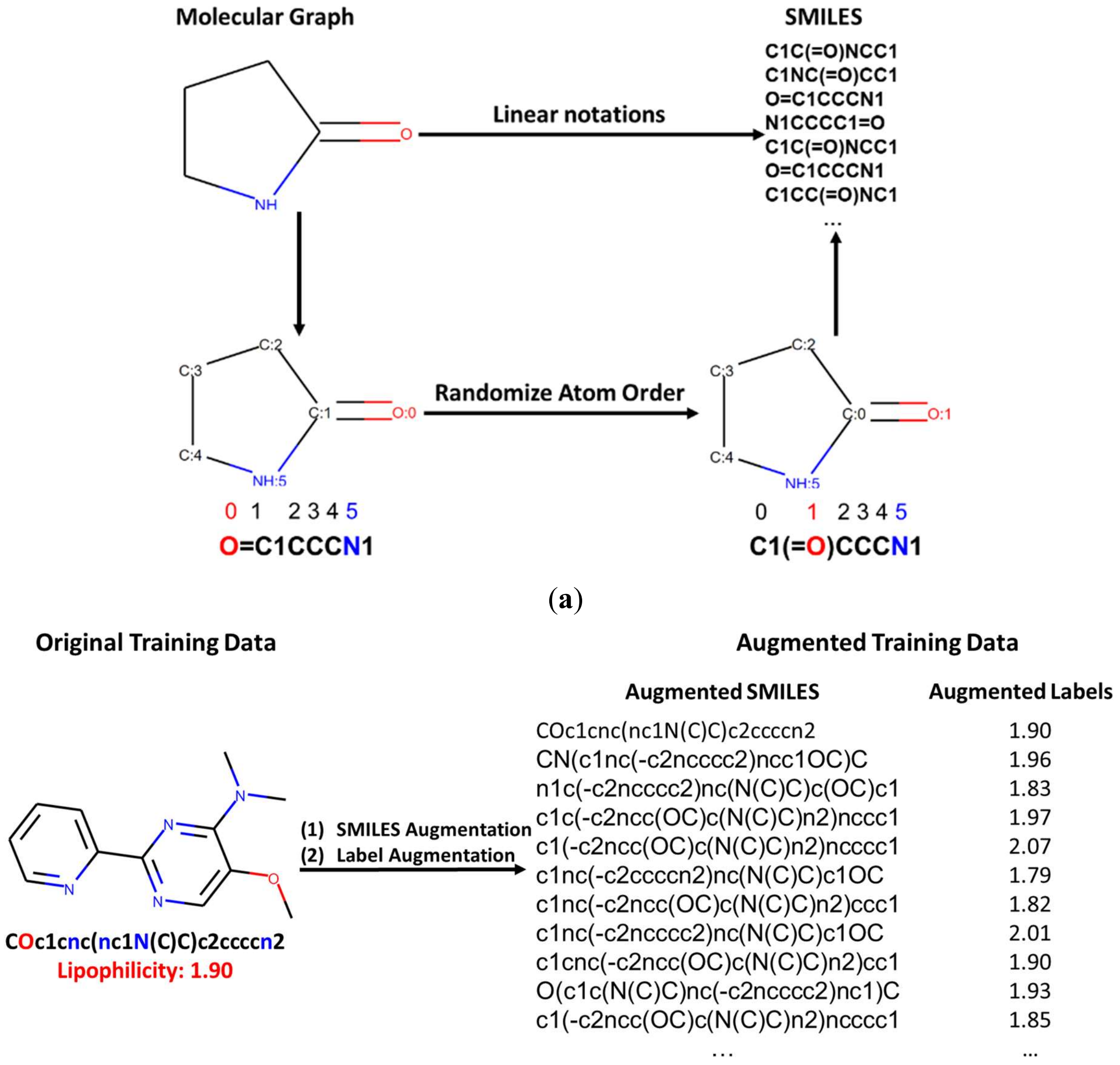
Table 1. Description of QSAR/QSPR datasets.

Data Set	Category	Description	Size	# of Active Compound	Task
Lipophilicity	Physical Chemistry	Octanol/water distribution coefficient	4,200		*Regression*
HIV	Biophysics	Inhibition of HIV replication	41,127	1,443	*Classification*
BBBP	Physiology	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. 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**).

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 some 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.



(**a**)

Augmented SMILES	Augmented Labels
COc1cnc(nc1N(C)C)c2ccccn2	1.90
CN(c1nc(-c2ncccc2)ncc1OC)C	1.96
n1c(-c2ncccc2)nc(N(C)C)c(OC)c1	1.83
c1c(-c2ncc(OC)c(N(C)C)n2)nccc1	1.97
c1(-c2ncc(OC)c(N(C)C)n2)ncccc1	2.07
c1nc(-c2ccccn2)nc(N(C)C)c1OC	1.79
c1nc(-c2ncc(OC)c(N(C)C)n2)ccc1	1.82
c1nc(-c2ncccc2)nc(N(C)C)c1OC	2.01
c1cnc(-c2ncc(OC)c(N(C)C)n2)cc1	1.90
O(c1c(N(C)C)nc(-c2ncccc2)nc1)C	1.93
c1(-c2ncc(OC)c(N(C)C)n2)ncccc1	1.85
…	…

(**b**)

Figure 2. SMILES and Data Augmentation.

2.5. Data Augmentation

Deep learning models are data-hungry so that various data augmentation techniques have been developed for different types of datasets and applications[65–68]. Data augmentation usually helps deep learning models to be generalized better for new data. A single molecular structure can be represented by multiple SMILES. Herein, we used this characteristics (often sought as a detrimental one) as the basis for data augmentation technique[69]. 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[70] (**Figure 2b**). The standard deviation σ_{noise} is considered as a hyperparameter for the models and need to be tuned from task to task. 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. The impact of SMILES augmentation and TTA will be discussed in **Section 3.2**.

2.6. Other Technical Procedures

Data Augmentation: It has been shown that the generative models trained on both augmented and canonical SMILES can create a larger chemical space of structures[69,71]. 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 dataset (regression), the number of augmented SMILES and the label noise σ_{noise} were tuned on the validation set on a single 80:10:10 random split (See discussion in Section 3.2). 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.

QSAR/QSPR Model Fine-Tuning: We are interested in obtaining a model that perform accurately for 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).

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

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

3. RESULTS AND DISCUSSION

3.1. Model Comparison

We benchmarked our MolPMoFiT method to other published models from Yang et al[10] on three well-studied datasets: lipophilicity, HIV and BBBP. All the models were evaluated on the same ten 80:10:10 splits from Yang et al[10] to ensure a fair and reproducible benchmark. Results for lipophilicity 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). Time-time augmentation (TTA, See Section 2.5) was applied for computing the results of MolPMoFiT models. Both random and scaffold splits were evaluated. Scaffold split enforced all training and test sets shared no common molecular scaffolds, which is a more challenging and realistic evaluation compared to a random split. For lipophilicity data, the regression model was fine-tuned on the general domain MSPM. For HIV and BBBP data, the classification models were fine-tuned on the task-specific MSPMs.

The results for test sets are summarized in **Figure 3**. Across all three data sets, MolPMoFiT models achieved comparable or better prediction performances compared to the published models. For all MolPMoFiT models, we used the same hyperparameter settings optimized on the HIV dataset. The experiments showed that our method can achieve great reliability without specific optimization. The performance of models on each individual task could in fact be further improved with proper hyperparameter tuning.

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[72]. Specifically, for lipophilicity data, MolPMoFiT achieved a test set RMSE of 0.565 and 0.635 on random split and scaffold split, respectively. For BBBP data, MolPMoFiT achieved a test set AUROC of 0.942 and 0.926 on random split and scaffold split, respectively. For HIV data, MolPMoFiT achieved a test set AUROC of 0.834 and 0.805 on random split and scaffold split, respectively.

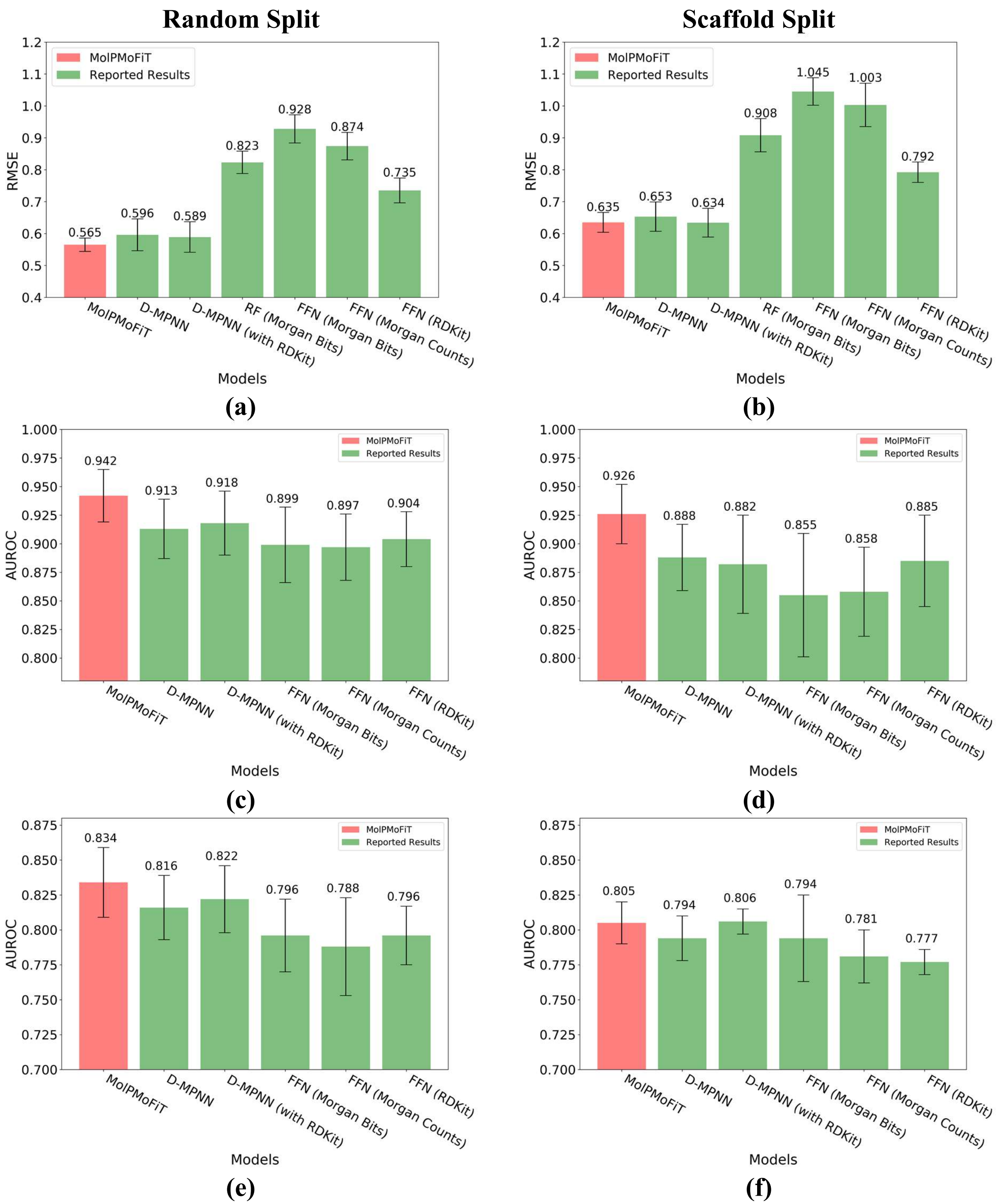
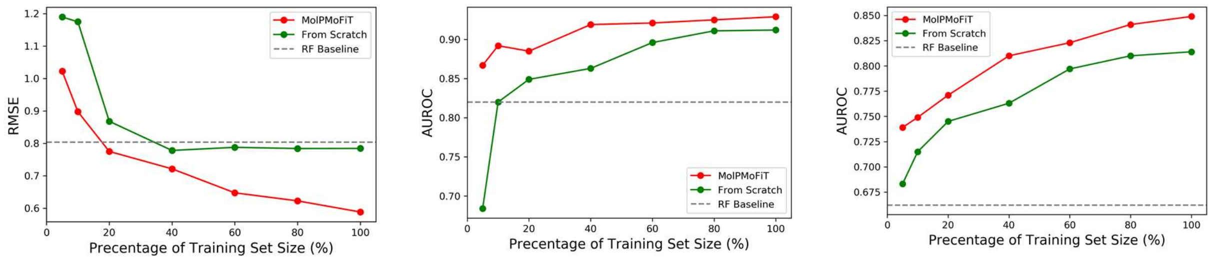


Figure 3. Comparison of MolPMoFiT to Reported Results from Young's[10]. The results of lipophilicity data are shown in (**a**) and (**b**); The results of BBBP data are shown in (**c**) and (**d**); The results of BBBP data are shown in (**e**) and (**f**); (**a**), (**c**) and (**e**) show the results of random splits. (**b**), (**d**) and (**f**) show the results of scaffold splits. MolPMoFiT: Molecular Prediction Model Fine-Tuning; D-MPNN: Directed Message Passing Neural Network; RF: Random Forest; FFN: Feed-Forward Network.

3.2. Analysis

Impact of Transfer Learning: We evaluated MolPMoFiT on different training datasets and kept the test sets fixed on a single 80:10:10 random split. The MolPMoFiT models were compared to the models that were trained from scratch. The hyperparameters and training epochs were kept fixed. The results are illustrated in **Figure 4**. Generally, with different numbers of training data, the MolPMoFiT model was always outperforming the model trained from scratch. This indicated that the MolPMoFiT transfer learning technique provided a robust improvement for the model performances.

The baseline random forest models were trained on the whole training data with ECFP6[73] (grey lines in **Figure 4**). Regarding the lipophilicity dataset, with only 20% of the training data, the MolPMoFiT model matched the performance of the baseline random forest model built with the full set, which is highly encouraging. On BBBP and HIV data, the MolPMoFiT models trained with only 5% of the training data outperform the baseline random forest model built on the full set.



(a) Lipophilicity **(b)** BBBP (**c**) HIV

Figure 4. Performances of models on the different size of the training set. Random forest baseline model is trained on the full training set with ECFP6. (**a**) Lipophilicity; (**b**) BBBP and (**c**) HIV.

Impact of Task Specific MSPM: We compared the QSAR model performances with the fine-tuned task-specific MSPM versus no fine-tuning on HIV and BBBP data sets (**Table 3**). For BBBP, the fine-tuning was found not beneficial for the model performances. For HIV, the fine-tuning actually led to worse results for the scaffold split.

Table 3. Test AUROC for MolPMoFiT with and without task specific MSPM fine-tuning.

Dataset	Random Split		Scaffold Split	
	No Fine-tuning	Fine-tuned	No Fine-tuning	Fine-tuned
BBBP	0.945 ± 0.023	0.942 ± 0.023	0.929 ± 0.023	0.926 ± 0.026
HIV	0.828 ± 0.029	0.834 ± 0.025	0.816 ± 0.022	0.805 ± 0.015

Impact of Data Augmentation for Regression Task: We assessed the effect of data augmentation on lipophilicity data in a single 80:10:10 random split. Models were trained on different sizes of augmented training data, whose labels were perturbed with different Gaussian noise. The evaluated numbers of augmented SMILES per compound were {0, 5, 15, 25, 50} and evaluated σ_{noise} values were {0, 0.1, 0.3, 0.5}. The models were tested according to two scenarios: (1) only test on canonical SMILES and (2) Test-time augmentation (TTA) (averaging predictions of the canonical SMILES and four augmented SMILES). The results on the test set are shown in **Figure 5**. When the model is only trained on the original training data (no augmented SMILES and perturbed labels), the RMSE of only tested on canonical SMILES is lower than that with TTA, indicating only training the model on canonical SMILES will limit the predictive ability of the model on non-canonical SMILES.

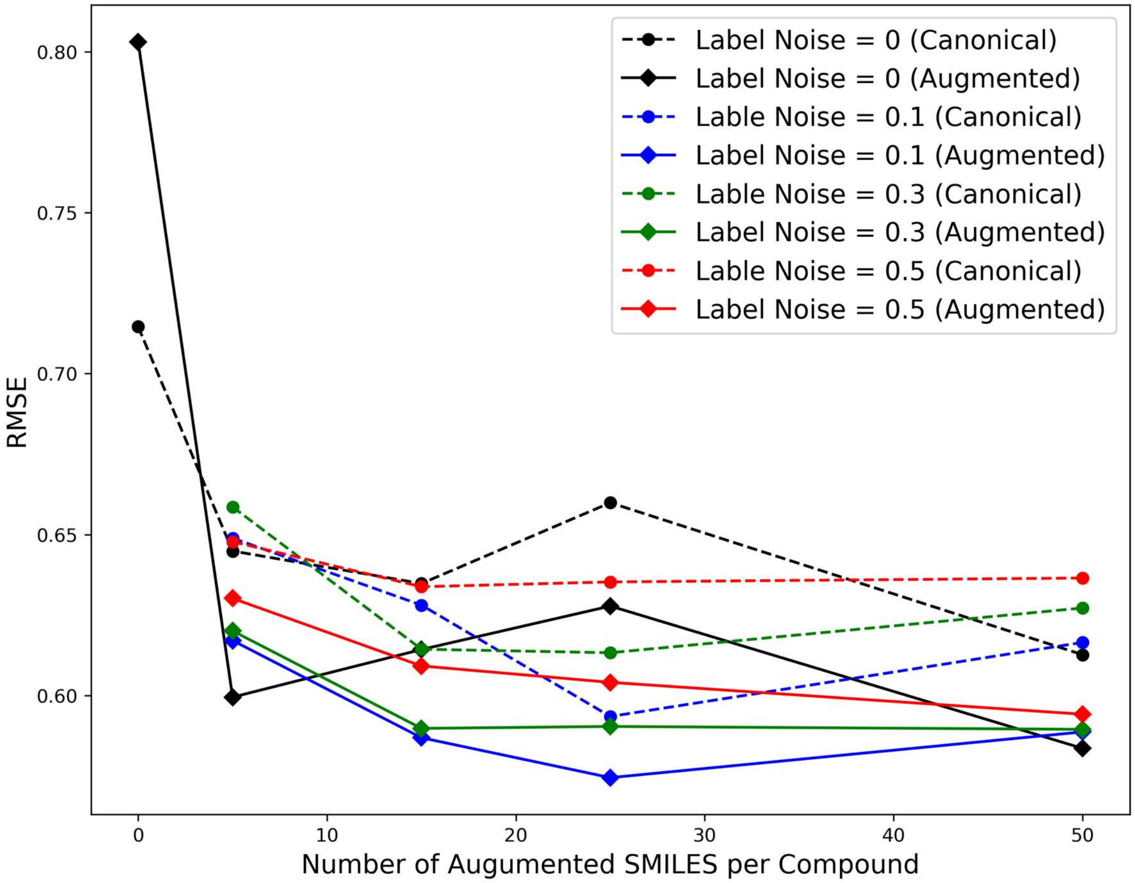
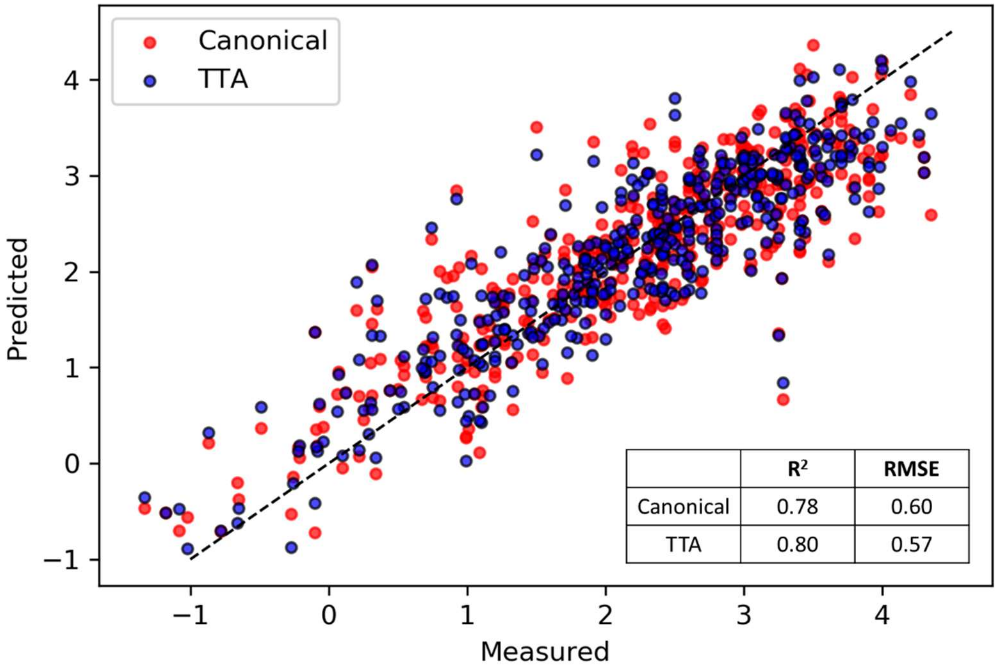


Figure 5. Performances of Lipophilicity models on different number of augmented SMILES per compound and Gaussian Noise (σ_{noise}) added to the original experimental values.

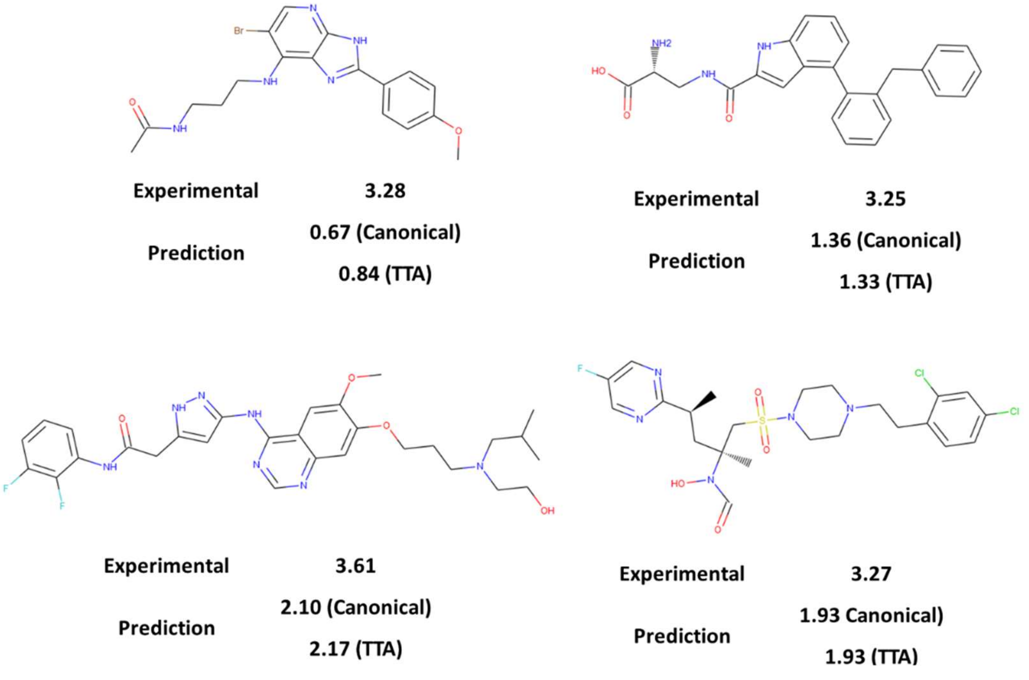
The results in **Figure 5** show one limitation of using SMILES as input for deep learning model: the model actually learns to map individual SMILES to molecular properties/activities instead of linking molecular structures to their properties/activities. The augmented SMILES are used as a regularization 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 model. For the same data augmentation setting, testing results with TTA are 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.

Figure 6 shows the correlation between experimental and predicted lipophilicity values for the test set compounds with the best data augmentation parameters: augmented SMILES per compound 25 and σ_{noise} 0.1. Most of the prediction errors are fallen into one log unit. The top mis-predicted molecules are listed in **Figure 7**.



	R²	RMSE
Canonical	0.78	0.60
TTA	0.80	0.57

Figure 6. Measured values vs. Predicted values of a Lipophilicity Model.



Experimental 3.28
Prediction 0.67 (Canonical)
0.84 (TTA)

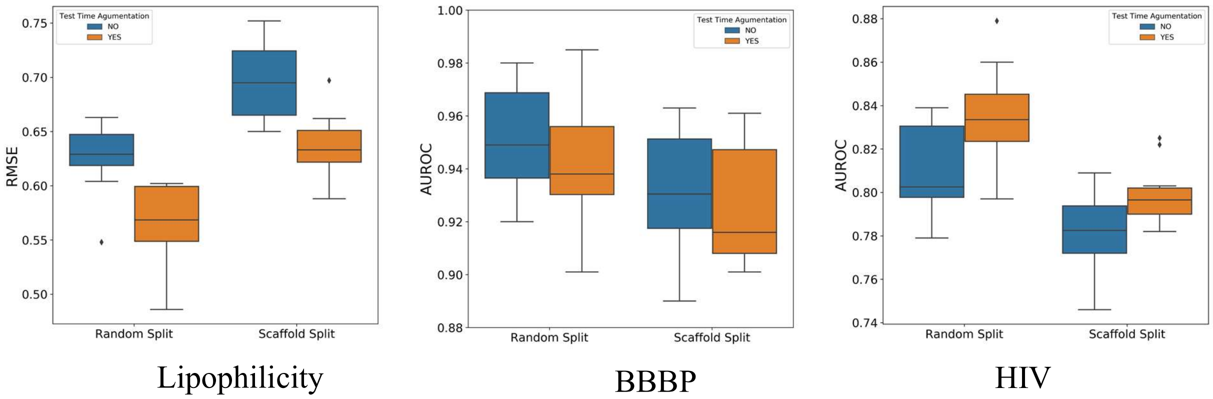
Experimental 3.25
Prediction 1.36 (Canonical)
1.33 (TTA)

Experimental 3.61
Prediction 2.10 (Canonical)
2.17 (TTA)

Experimental 3.27
Prediction 1.93 Canonical)
1.93 (TTA)

Figure 7. Top Mis-Predicted Molecules.

Impact of Test Time Augmentation (TTA): We compared the results from predictions on canonical SMILES only and TTA (averaging predictions from the canonical and four augmented SMILES). All models were evaluated on ten 80:10:10 splits. The test set results are illustrated in **Figure 8.** For lipophilicity and HIV data, TTA significantly improved the model performance, whereas TTA was found not to be beneficial for BBBP.



Lipophilicity BBBP HIV

Figure 8. Comparison of model performances on predictions from Canonical SMILES and Test time augmentation (TTA).

Conclusions

In this study, we introduced the MolPMoFiT, a novel transfer learning method for QSPR/QSAR tasks. We pre-trained a universal molecular structure prediction model using one million bioactive molecules from ChEMBL and then fine-tuned it for various QSPR/QSAR tasks. The method is universal in the sense of using a single architecture and training process across QSPR/QSAR tasks. 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 three benchmark datasets. 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.

References

(1) Cherkasov, A., Muratov, E. N., Fourches, D., Varnek, A., Baskin, I. I., Cronin, M., Dearden, J., Gramatica, P., Martin, Y. C., Todeschini, R., et al. (2014) QSAR Modeling: Where Have You Been? Where Are You Going To? *J. Med. Chem. 57*, 4977–5010.

(2) Mater, A. C., Coote, M. L. (2019) Deep Learning in Chemistry. *J. Chem. Inf. Model. 59*, 2545–2559.

(3) Tropsha, A. (2010) Best Practices for QSAR Model Development, Validation, and Exploitation. *Molecular Informatics*. John Wiley & Sons, Ltd July 6, 2010, pp 476–488.

(4) Ma, J., Sheridan, R. P., Liaw, A., Dahl, G. E., Svetnik, V. (2015) Deep Neural Nets as a Method for Quantitative Structure–Activity Relationships. *J. Chem. Inf. Model. 55*, 263–274.

(5) Fourches, D., Williams, A. J., Patlewicz, G., Shah, I., Grulke, C., Wambaugh, J., Richard, A., Tropsha, A. (2018) Computational Tools for ADMET Profiling. In *Computational Toxicology*; pp 211–244.

(6) 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.

(7) Fourches, D., Ash, J. (2019) 4D- Quantitative Structure–Activity Relationship Modeling: Making a Comeback. *Expert Opin. Drug Discov.* 1–9.

(8) Xue, L., Bajorath, J. (2000) Molecular Descriptors in Chemoinformatics, Computational Combinatorial Chemistry, and Virtual Screening. *Comb. Chem. High Throughput Screen. 3*, 363–372.

(9) Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., Pande, V. (2018) MoleculeNet: A Benchmark for Molecular Machine Learning. *Chem. Sci. 9*, 513–530.

(10) 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.

(11) Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., Dahl, G. E. (2017) Neural Message Passing for Quantum Chemistry.

(12) Chen, C., Ye, W., Zuo, Y., Zheng, C., Ong, S. P. (2019) Graph Networks as a Universal Machine Learning Framework for Molecules and Crystals. *Chem. Mater. 31*, 3564–3572.

(13) Duvenaud, D., Maclaurin, D., Aguilera-Iparraguirre, J., Gómez-Bombarelli, R., Hirzel, T., Aspuru-Guzik, A., Adams, R. P. (2015) Convolutional Networks on Graphs for Learning Molecular Fingerprints. *Adv. Neural Inf. Process. Syst. 2015-Janua*, 2224–2232.

(14) Coley, C. W., Barzilay, R., Green, W. H., Jaakkola, T. S., Jensen, K. F. (2017) Convolutional Embedding of Attributed Molecular Graphs for Physical Property Prediction. *J. Chem. Inf. Model. 57*, 1757–1772.

(15) Pham, T., Tran, T., Venkatesh, S. (2018) Graph Memory Networks for Molecular Activity Prediction.

(16) Wang, X., Li, Z., Jiang, M., Wang, S., Zhang, S., Wei, Z. (2019) Molecule Property Prediction Based on Spatial Graph Embedding. *J. Chem. Inf. Model.* acs.jcim.9b00410.

(17) Feinberg, E. N., Sur, D., Wu, Z., Husic, B. E., Mai, H., Li, Y., Sun, S., Yang, J., Ramsundar, B., Pande, V. S. (2018) PotentialNet for Molecular Property Prediction. *ACS Cent. Sci. 4*, 1520–1530.

(18) Goh, G. B., Hodas, N. O., Siegel, C., Vishnu, A. (2017) SMILES2Vec: An Interpretable General-Purpose Deep Neural Network for Predicting Chemical Properties.

(19) 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.

(20) Kimber, T. B., Engelke, S., Tetko, I. V, Bruno, E., Godin, G. (2018) Synergy Effect between Convolutional Neural Networks and the Multiplicity of SMILES for Improvement of Molecular Prediction.

(21) Goh, G. B., Siegel, C., Vishnu, A., Hodas, N. O., Baker, N. (2017) Chemception: A Deep

Neural Network with Minimal Chemistry Knowledge Matches the Performance of Expert-Developed QSAR/QSPR Models.

(22) Goh, G. B., Siegel, C., Vishnu, A., Hodas, N. O. (2017) Using Rule-Based Labels for Weak Supervised Learning: A ChemNet for Transferable Chemical Property Prediction. *9*.

(23) Paul, A., Jha, D., Al-Bahrani, R., Liao, W., Choudhary, A., Agrawal, A. (2018) CheMixNet: Mixed DNN Architectures for Predicting Chemical Properties Using Multiple Molecular Representations.

(24) Goh, G. B., Siegel, C., Vishnu, A., Hodas, N., Baker, N. *How Much Chemistry Does a Deep Neural Network Need to Know to Make Accurate Predictions?*

(25) Fernandez, M., Ban, F., Woo, G., Hsing, M., Yamazaki, T., LeBlanc, E., Rennie, P. S., Welch, W. J., Cherkasov, A. (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.

(26) Weininger, D. (1988) SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Model. 28*, 31–36.

(27) Weininger, D., Weininger, A., Weininger, J. L. (1989) SMILES. 2. Algorithm for Generation of Unique SMILES Notation. *J. Chem. Inf. Model. 29*, 97–101.

(28) Hopfield, J. J. (1982) Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proc. Natl. Acad. Sci. 79*, 2554–2558.

(29) Lipton, Z. C., Berkowitz, J., Elkan, C. (2015) A Critical Review of Recurrent Neural Networks for Sequence Learning.

(30) Kim, Y. (2014) Convolutional Neural Networks for Sentence Classification.

(31) Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. (2017) Attention Is All You Need.

(32) Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, Li Fei-Fei. (2009) ImageNet: A Large-Scale Hierarchical Image Database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*; IEEE; pp 248–255.

(33) Canziani, A., Paszke, A., Culurciello, E. (2016) An Analysis of Deep Neural Network Models for Practical Applications.

(34) Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013) Efficient Estimation of Word Representations in Vector Space. 1–12.

(35) Jeffrey Pennington, Richard Socher, C. D. M. (2014) GloVe: Global Vectors for Word Representation. 2014.

(36) Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., Mikolov, T. (2016) FastText.Zip: Compressing Text Classification Models.

(37) Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L. (2018) Deep Contextualized Word Representations.

(38) Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. (2018) BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding.

(39) Howard, J., Ruder, S. (2018) Universal Language Model Fine-Tuning for Text Classification.

(40) Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q. V. (2019) XLNet: Generalized Autoregressive Pretraining for Language Understanding.

(41) Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V. (2019) RoBERTa: A Robustly Optimized BERT Pretraining Approach.

(42) Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., Light, Y.,

McGlinchey, S., Michalovich, D., Al-Lazikani, B., et al. (2012) ChEMBL: A Large-Scale Bioactivity Database for Drug Discovery. *Nucleic Acids Res. 40*, 1100–1107.

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

(44) Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., Leskovec, J. (2019) Pre-Training Graph Neural Networks.

(45) Xu, Y., Ma, J., Liaw, A., Sheridan, R. P., Svetnik, V. (2017) Demystifying Multitask Deep Neural Networks for Quantitative Structure-Activity Relationships. *J. Chem. Inf. Model. 57*, 2490–2504.

(46) 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.

(47) de la Vega de León, A., Chen, B., Gillet, V. J. (2018) Effect of Missing Data on Multitask Prediction Methods. *J. Cheminform. 10*, 26.

(48) Wu, K., Wei, G.-W. (2018) Quantitative Toxicity Prediction Using Topology Based Multitask Deep Neural Networks. *J. Chem. Inf. Model. 58*, 520–531.

(49) Varnek, A., Gaudin, C., Marcou, G., Baskin, I., Pandey, A. K., Tetko, I. V. (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.

(50) Ramsundar, B., Liu, B., Wu, Z., Verras, A., Tudor, M., Sheridan, R. P., Pande, V. (2017) Is Multitask Deep Learning Practical for Pharma? *J. Chem. Inf. Model. 57*, 2068–2076.

(51) Wu, K., Wei, G.-W. (2018) Quantitative Toxicity Prediction Using Topology Based Multitask Deep Neural Networks. *J. Chem. Inf. Model. 58*, 520–531.

(52) Varnek, A., Gaudin, C., Marcou, G., Baskin, I., Pandey, A. K., Tetko, I. V. (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.

(53) Merity, S., Xiong, C., Bradbury, J., Socher, R. (2016) Pointer Sentinel Mixture Models.

(54) Linzen, T., Dupoux, E., Goldberg, Y. (2016) Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies.

(55) Gulordava, K., Bojanowski, P., Grave, E., Linzen, T., Baroni, M. (2018) Colorless Green Recurrent Networks Dream Hierarchically.

(56) Radford, A., Jozefowicz, R., Sutskever, I. (2017) Learning to Generate Reviews and Discovering Sentiment.

(57) Merity, S., Keskar, N. S., Socher, R. (2017) Regularizing and Optimizing LSTM Language Models.

(58) Hochreiter, S., Schmidhuber, J. (1997) Long Short-Term Memory. *Neural Comput. 9*, 1735–1780.

(59) Smith, L. N. (2018) A Disciplined Approach to Neural Network Hyper-Parameters: Part 1 -- Learning Rate, Batch Size, Momentum, and Weight Decay.

(60) Yosinski, J., Clune, J., Bengio, Y., Lipson, H. (2014) How Transferable Are Features in Deep Neural Networks?

(61) Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A. (2017) Automatic Differentiation in PyTorch.

(62) Howard, J., others. (2018) Fastai. GitHub 2018.

(63) Swain, M. MolVS: Molecule Validation and Standardization.

(64) Landrum, G. RDKit: Open-Source Cheminformatics.

(65) Fadaee, M., Bisazza, A., Monz, C. (2017) Data Augmentation for Low-Resource Neural Machine Translation.

(66) 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.

(67) 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.

(68) Hu, B., Lei, C., Wang, D., Zhang, S., Chen, Z. (2019) A Preliminary Study on Data Augmentation of Deep Learning for Image Classification.

(69) Arús-Pous, J., Johansson, S., Prykhodko, O., Bjerrum, E. J. (2019) Randomized SMILES Strings Improve the Quality of Molecular Generative Models. *ChemRxiv*.

(70) Cortes-Ciriano, I., Bender, A. (2015) Improved Chemical Structure–Activity Modeling Through Data Augmentation. *J. Chem. Inf. Model. 55*, 2682–2692.

(71) Arús-Pous, J., Blaschke, T., Ulander, S., Reymond, J.-L., Chen, H., Engkvist, O. (2019) Exploring the GDB-13 Chemical Space Using Deep Generative Models. *J. Cheminform. 11*, 20.

(72) Sheridan, R. P. (2013) Time-Split Cross-Validation as a Method for Estimating the Goodness of Prospective Prediction. *J. Chem. Inf. Model. 53*, 783–790.

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