

Supplementary Information for:
Systematic Benchmarking of 13 AI Methods for
Predicting Cyclic Peptide Membrane Permeability

Wei Liu, Jianguo Li, Chandra S. Verma, and Hwee Kuan Lee

1 Brief description of benchmarked models

- RF (Random Forest) [1]: An ensemble learning method that constructs multiple decision trees during training and aggregates their predictions for improved generalization. By leveraging bagging and random feature subsets, it reduces overfitting and handles high-dimensional data effectively.
- SVM (Support Vector Machine) [2]: A supervised max-margin models with associated learning algorithms designed for classification and regression tasks. It can handle non-linear decision boundaries using kernel tricks and is particularly effective for small to medium-sized datasets.
- MPNN (Message Passing Neural Network) [3]: A fundamental framework for graph neural networks where nodes update their features by exchanging messages with neighboring nodes. This iterative process allows the network to capture local structural patterns, making it particularly effective for molecular property prediction tasks.
- AttentiveFP [4]: A graph neural network that employs an attention mechanism for extracting nonlocal effects at the intramolecular level. This mechanism enables the model to focus on the most relevant atomic features, improving prediction accuracy.
- DMPNN (Directed Message Passing Neural Network) [5]: A graph-based model that updates edge features during the message-passing phase, whereas other GNN-based benchmarks primarily update node features. This design aims to avoid "totters", a phenomenon where a node redundantly sends back a message it previously received from a neighboring node in earlier layers.
- GAT (Graph Attention Network) [6]: A graph neural network that uses attention mechanisms to weigh the importance of neighboring nodes when aggregating information. Unlike traditional GNNs that treat all neighbors equally or rely on fixed weights, GAT dynamically learns attention scores, allowing the model to focus more on informative neighbors.
- GCN (Graph Convolutional Network) [7]: A neural architecture that extends convolutional operations to graph-structured data by aggregating neighboring node features with degree-normalized weights. It efficiently captures local graph structure and node attributes, enabling applications like node classification, link prediction, and graph classification.
- PAGTN (Path-Augmented Graph Transformer Network) [8]: A graph neural architecture that enhances standard graph transformers by explicitly encoding path information between distant nodes, rather than relying solely on local neighborhood attention. This mechanism allows the model to better capture long-range dependencies and global graph structure, improving performance in molecular property prediction and other graph-based tasks.
- RNN (Recurrent Neural Network) [9]: A foundational neural architecture designed to process sequential data by maintaining hidden states that propagate information across time steps. While effective for short sequences, standard RNNs suffer from vanishing gradients, making them prone to forgetting long-range dependencies during training.
- LSTM (Long Short-Term Memory) [10]: An advanced RNN architecture that addresses vanishing gradients through gated mechanisms (input, output, and forget gates). It excels at modeling long-range dependencies in sequences, such as text or time-series data.
- GRU (Gated Recurrent Unit) [11]: A simplified LSTM variant using just two gates (update and reset) for efficient sequential modeling. It matches LSTM's performance on many tasks while being faster to train and requiring fewer parameters.
- ChemCepcion [12]: A convolutional neural network model based on the Inception-ResNet architecture [13], designed to learn molecular features directly from 2D images generated from SMILES. It is a purely data-driven approach that does not rely on prior chemical knowledge.
- ImageMol [14]: A convolutional neural network that treats molecular structures as 2D images for molecular property prediction. The model was pretrained in an unsupervised manner on 10 million unlabeled drug-like and bioactive molecules.

2 Evaluation metrics

In the regression task, we selected a few metrics to evaluate the model performance, mean absolute error (MAE), root mean square error (RMSE), coefficient of determination (R^2), and Pearson correlation coefficient (r):

$$\begin{aligned} MAE &= \frac{1}{N} \sum_{i=1}^N |o_i - y_i| \\ RMSE &= \sqrt{\frac{1}{N} \sum_{i=1}^N (o_i - y_i)^2} \\ R^2 &= 1 - \frac{\sum_{i=1}^N (o_i - y_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \\ r &= \frac{\sum_{i=1}^N (o_i - \bar{o})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (o_i - \bar{o})^2 \sum_{i=1}^N (y_i - \bar{y})^2}} \end{aligned}$$

where N is the number of samples, o_i is the predicted value for the i th sample, y_i is the actual value of the i th sample, $\bar{o}$ and $\bar{y}$ are the mean of the predicted and actual value, respectively.

In the classification task, we evaluated the performance with accuracy (Acc), precision(Pre), recall score (Recall), and F1 score (F1):

$$\begin{aligned} Acc &= \frac{TP + TN}{TP + TN + FP + FN} \\ Pre &= \frac{TP}{TP + EP} \\ Recall &= \frac{TP}{TP + EN} \\ F1 &= \frac{2TP}{2TP + FP + FN} \end{aligned}$$

In addition, the area under the receiver operating characteristic curve (AUC-ROC) was used as a general metric across all methods to make a fair performance comparison between the regression task and the classification task. AUC-ROC was computed using the scikit-learn library [15]. For classification tasks, the computation is straightforward. For regression tasks, we linearly mapped the model output range from $[-1, 1]$ to $[0, 1]$ before calculating the AUC-ROC. Although the output of a regression model corresponds to the predicted permeability value, we interpreted it as the probability of being a positive sample when computing AUC-ROC. This assumption is reasonable to some extent, as higher predicted permeability values generally indicate a greater likelihood of cell permeability.

3 Code modification in DeepChem for soft labels

Since the default DeepChem configuration does not support the classification task with soft labels, we modified function `SparseSoftmaxCrossEntropy` in `deepchem.models.losses` from:

```
if len(labels.shape) == len(output.shape):  
    labels = labels.squeeze(-1)  
return ce_loss(output, labels.long())
```

to:

```
if (len(labels.shape) == len(output.shape) and  
    labels.size(-1) == output.size(-1)):  
    return ce_loss(output, labels)  
else:  
    return ce_loss(output, labels.long())
```

This modification allows compatibility with soft labels by checking both the shape and the size of the last dimension, enabling flexible label handling when soft labels are required.

4 Model training details

Table S1: Model training details.

Method	Backend	Optimizer	Learning Rate	Pretraining Method
RF	scikit-learn	N.A.	N.A.	N.A.
SVM	scikit-learn	N.A.	N.A.	N.A.
DMPNN	DeepChem	Adam	0.001	Pretrained from Scratch
GAT	DeepChem	Adam	0.001	Pretrained from Scratch
GCN	DeepChem	Adam	0.001	Pretrained from Scratch
MPNN	DeepChem	Adam	0.001	Pretrained from Scratch
PAGTN	DeepChem	Adam	0.001	Pretrained from Scratch
RNN	PyTorch	Adam	0.0001	Pretrained from Scratch
LSTM	PyTorch	Adam	0.001	Pretrained from Scratch
GRU	PyTorch	Adam	0.001	Pretrained from Scratch
ChemCeption	DeepChem	Adam	0.001	Pretrained from Scratch
ImageMol	PyTorch	SGD	0.01	Loaded Shared Weights

N.A.: not applicable. For regression tasks, models were pretrained on the Delaney dataset [16]. For classification tasks, models were pretrained on the Blood-Brain Barrier Penetration dataset [17].

5 Model performance on random-split classification tasks using binary and soft labels

Table S2: Performance of models in binary-label classification on random-split test data for peptides of lengths 6, 7, and 10.

Represent molecule as	Method	Acc $\uparrow$	Pre $\uparrow$	Recall $\uparrow$	F1 $\uparrow$	ROC-AUC $\uparrow$
Fingerprint	RF	0.790 $\pm$ 0.012	0.830 $\pm$ 0.015	0.866$\pm$0.018	0.847 $\pm$ 0.011	0.850 $\pm$ 0.014
	SVM	0.780 $\pm$ 0.015	0.813 $\pm$ 0.016	0.873$\pm$0.012	0.842 $\pm$ 0.012	0.838 $\pm$ 0.014
Graph	AttentiveFP	0.791 $\pm$ 0.010	0.822 $\pm$ 0.024	0.881$\pm$0.030	0.850 $\pm$ 0.009	0.856 $\pm$ 0.012
	DMPNN	0.816*$\pm$0.010	0.850$\pm$0.014	0.881$\pm$0.017	0.865*$\pm$0.008	0.872*$\pm$0.012
	GAT	0.791 $\pm$ 0.012	0.820 $\pm$ 0.025	0.884$\pm$0.027	0.850 $\pm$ 0.009	0.848 $\pm$ 0.015
	GCN	0.774 $\pm$ 0.015	0.801 $\pm$ 0.024	0.886$\pm$0.032	0.840 $\pm$ 0.011	0.825 $\pm$ 0.022
	MPNN	0.766 $\pm$ 0.070	0.813 $\pm$ 0.052	0.842 $\pm$ 0.063	0.827 $\pm$ 0.056	0.809 $\pm$ 0.101
	PAGTN	0.769 $\pm$ 0.017	0.800 $\pm$ 0.018	0.874$\pm$0.037	0.835 $\pm$ 0.017	0.816 $\pm$ 0.014
String	RNN	0.710 $\pm$ 0.022	0.747 $\pm$ 0.019	0.861$\pm$0.032	0.799 $\pm$ 0.017	0.720 $\pm$ 0.035
	LSTM	0.767 $\pm$ 0.018	0.812 $\pm$ 0.025	0.851 $\pm$ 0.030	0.830 $\pm$ 0.013	0.823 $\pm$ 0.020
	GRU	0.766 $\pm$ 0.023	0.815 $\pm$ 0.016	0.843 $\pm$ 0.057	0.828 $\pm$ 0.025	0.823 $\pm$ 0.020
Image	ChemCeption	0.742 $\pm$ 0.019	0.793 $\pm$ 0.042	0.840 $\pm$ 0.056	0.813 $\pm$ 0.013	0.772 $\pm$ 0.027
	ImageMol	0.797 $\pm$ 0.009	0.845$\pm$0.011	0.849 $\pm$ 0.019	0.847 $\pm$ 0.011	0.860 $\pm$ 0.010
Mix	Multi_CycGT ¹	0.821 $\pm$ 0.019	0.860 $\pm$ 0.017	0.882 $\pm$ 0.017	0.871 $\pm$ 0.015	0.861 $\pm$ 0.022
	MuCoCP ¹	0.870	0.771	0.910	0.909	—

If the best metric value is statistically superior to the second-best, it is marked with *; otherwise, values not statistically inferior to the best are presented in bold. Metric values from Multi_CycG and MuCoCP were taken from the original reports, where their datasets differ from this study but are all subsets of the CycPeptMPDB dataset.

Table S3: Performance of models in soft-label classification on random-split test data for peptides of lengths 6, 7, and 10.

Represent molecule as	Method	Acc $\uparrow$	Pre $\uparrow$	Recall $\uparrow$	F1 $\uparrow$	ROC-AUC $\uparrow$
Graph	AttentiveFP	0.795 $\pm$ 0.008	0.819 $\pm$ 0.020	0.891$\pm$0.026	0.853 $\pm$ 0.007	0.862 $\pm$ 0.008
	DMPNN	0.815*$\pm$0.017	0.850$\pm$0.021	0.880$\pm$0.034	0.864*$\pm$0.015	0.884*$\pm$0.008
	GAT	0.790 $\pm$ 0.011	0.814 $\pm$ 0.035	0.896$\pm$0.043	0.851 $\pm$ 0.005	0.851 $\pm$ 0.014
	GCN	0.773 $\pm$ 0.016	0.797 $\pm$ 0.036	0.893$\pm$0.050	0.840 $\pm$ 0.012	0.833 $\pm$ 0.017
	MPNN	0.792 $\pm$ 0.015	0.833$\pm$0.024	0.865$\pm$0.037	0.848 $\pm$ 0.014	0.860 $\pm$ 0.008
	PAGTN	0.765 $\pm$ 0.079	0.793 $\pm$ 0.056	0.879$\pm$0.076	0.833 $\pm$ 0.058	0.811 $\pm$ 0.119
String	RNN	0.719 $\pm$ 0.019	0.773 $\pm$ 0.024	0.822 $\pm$ 0.032	0.797 $\pm$ 0.016	0.745 $\pm$ 0.026
	LSTM	0.766 $\pm$ 0.024	0.828 $\pm$ 0.020	0.821 $\pm$ 0.029	0.825 $\pm$ 0.021	0.833 $\pm$ 0.019
	GRU	0.766 $\pm$ 0.015	0.845$\pm$0.023	0.799 $\pm$ 0.032	0.821 $\pm$ 0.015	0.848 $\pm$ 0.015

If the best metric value is statistically superior to the second-best, it is marked with *; otherwise, values not statistically inferior to the best are presented in bold. RF, SVM, ChemCeption, and ImageMol were not tested with soft labels due to their incompatibility or the complexity of code adaptation.

6 Chemical similarity comparison

The chemical similarity between two peptides was measured using the Tanimoto similarity based on Morgan fingerprints, as implemented in RDKit via the functions `BulkTanimotoSimilarity` and `GetMorganFingerprintAsBitVect`. Similarity values range from 0 to 1, with 1 indicating identical fingerprints.

We computed the chemical similarity between each sample in the validation/test sets and all samples in the training set. For each validation/test sample, we selected its maximum similarity to the training set and plotted the resulting values as histograms in Fig. S1. This analysis indicates whether the model encountered similar compounds during training, when making predictions on validation or test samples. As shown in Fig.S1, in the random split, most validation and test samples have a highly similar counterpart in the training set. In contrast, under the scaffold-based split, validation and especially test samples exhibit substantially lower similarity to the training set.

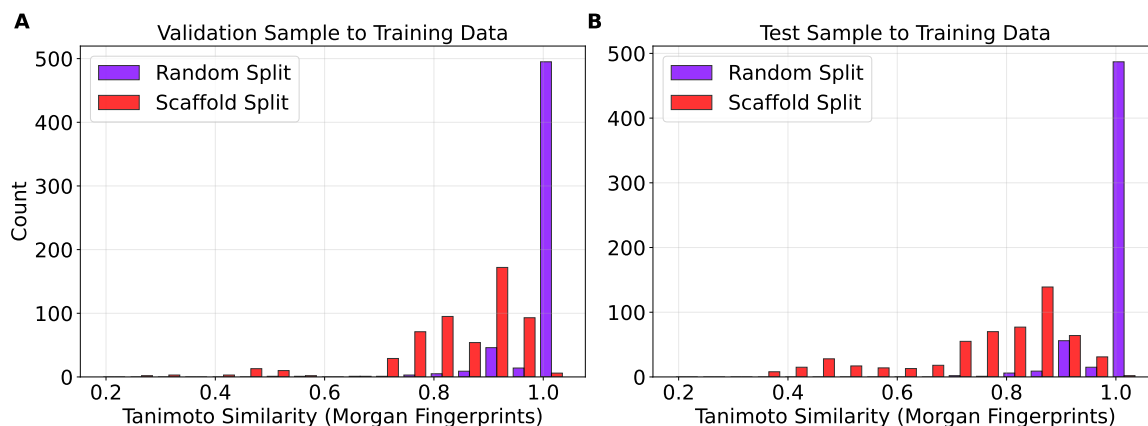


Figure S1: Histograms of maximum chemical similarity between each validation sample (a)/test sample (b) and the training set.

In order to show the chemical diversity of the subsets (training and test sets), for each peptide, we computed the average similarity to its top-k nearest neighbors within the same subset. The mean and standard deviation of these average similarities are shown in Fig. S2. The training set from the scaffold-based split exhibits the highest neighbor similarity, indicating the lowest chemical diversity. In contrast, the test set from the scaffold-based split shows the lowest neighbor similarity, reflecting the highest diversity. The training and test sets from the random split lie between, with the training set having higher similarity, which is expected. That is because the training set is eight times larger; larger sample sizes have a higher probability of including structurally similar neighbors when drawn from the same distribution.

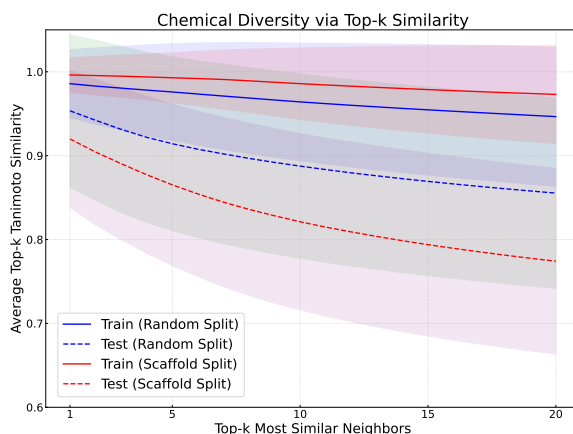


Figure S2: Chemical similarity plot of different subset split by scaffold-based or random. The pairwise similarity is computed using the Tanimoto similarity based on Morgan fingerprints. For each peptide, we computed the average similarity to its top-k nearest neighbors within the same subset. This plot shows the mean and standard deviation of these average similarities for each subset.

7 Model performance on scaffold-split test data

Table S4: Performance of models in regression task on scaffold-split test data for peptides of lengths 6, 7, and 10.

Method	MAE ↓	RMSE ↓	R^2 ↑	r ↑	ROC-AUC ↑
RF	0.709±0.004	0.930±0.007	0.186±0.012	0.438±0.018	0.756±0.007
SVM	0.782	1.013	0.034	0.205	0.601
AttentiveFP	0.758±0.044	0.997±0.061	0.059±0.115	0.368±0.154	0.649±0.067
DMPNN	0.629±0.047	0.827±0.052	0.353±0.082	0.605±0.066	0.765±0.033
GAT	0.808±0.024	1.096±0.053	-0.137±0.106	0.308±0.068	0.638±0.033
GCN	0.820±0.060	1.089±0.087	-0.127±0.182	0.285±0.142	0.634±0.062
MPNN	0.653±0.041	0.861±0.058	0.298±0.096	0.558±0.081	0.757±0.038
PAGTN	0.803±0.032	1.068±0.049	-0.078±0.101	0.202±0.111	0.590±0.047
RNN	0.837±0.007	1.103±0.010	-0.149±0.020	-0.014±0.075	0.522±0.035
LSTM	0.837±0.004	1.103±0.004	-0.147±0.009	0.037±0.173	0.528±0.055
GRU	0.811±0.030	1.083±0.031	-0.107±0.064	0.306±0.125	0.630±0.036
ChemCeption	0.821±0.011	1.077±0.016	-0.094±0.033	-0.243±0.084	0.416±0.021
ImageMol	0.801±0.023	1.030±0.017	-0.001±0.032	0.231±0.050	0.604±0.015

Table S5: Performance of models in binary-label classification on scaffold-split test data for peptides of lengths 6, 7, and 10.

Method	Acc ↑	Pre ↑	Recall ↑	F1 ↑	ROC-AUC ↑
RF	0.560±0.012	0.529±0.007	0.937±0.018	0.676±0.010	0.690±0.021
SVM	0.575±0.003	0.608±0.002	0.376±0.010	0.465±0.008	0.641±7.0e-05
AttentiveFP	0.556±0.042	0.540±0.037	0.886±0.111	0.665±0.028	0.594±0.052
DMPNN	0.593±0.078	0.569±0.059	0.890±0.070	0.689±0.029	0.626±0.084
GAT	0.526±0.021	0.514±0.012	0.951±0.062	0.667±0.023	0.546±0.050
GCN	0.525±0.024	0.515±0.014	0.880±0.069	0.649±0.025	0.597±0.052
MPNN	0.612±0.068	0.583±0.058	0.863±0.059	0.692±0.031	0.661±0.073
PAGTN	0.518±0.008	0.510±0.004	0.989±0.013	0.673±0.004	0.524±0.031
RNN	0.499±0.004	0.500±0.002	0.974±0.046	0.661±0.011	0.524±0.024
LSTM	0.501±1.2e-16	0.501±1.2e-16	1.00±0	0.667±1.2e-16	0.558±0.065
GRU	0.509±0.025	0.506±0.016	0.988±0.039	0.668±0.003	0.626±0.055
ChemCeption	0.483±0.029	0.490±0.020	0.749±0.113	0.590±0.042	0.483±0.047
ImageMol	0.622±0.032	0.632±0.040	0.589±0.065	0.607±0.028	0.661±0.047

Table S6: Performance of models in soft-label classification on scaffold-split test data for peptides of lengths 6, 7, and 10.

Method	Acc $\uparrow$	Pre $\uparrow$	Recall $\uparrow$	F1 $\uparrow$	ROC-AUC $\uparrow$
AttentiveFP	0.594$\pm$0.074	0.578$\pm$0.072	0.843 $\pm$ 0.110	0.676$\pm$0.031	0.618$\pm$0.109
DMPNN	0.577$\pm$0.089	0.560$\pm$0.071	0.919 $\pm$ 0.092	0.689$\pm$0.032	0.635$\pm$0.109
GAT	0.542 $\pm$ 0.023	0.525 $\pm$ 0.014	0.903 $\pm$ 0.091	0.662$\pm$0.030	0.582$\pm$0.058
GCN	0.544 $\pm$ 0.053	0.533$\pm$0.042	0.779 $\pm$ 0.085	0.630 $\pm$ 0.042	0.595$\pm$0.060
MPNN	0.601$\pm$0.070	0.574$\pm$0.055	0.878 $\pm$ 0.072	0.690$\pm$0.030	0.650$\pm$0.118
PAGTN	0.515 $\pm$ 0.009	0.508 $\pm$ 0.005	0.968$\pm$0.054	0.666 $\pm$ 0.013	0.522 $\pm$ 0.040
RNN	0.501 $\pm$ 1.2e-16	0.501 $\pm$ 1.2e-16	1.00$\pm$0	0.667 $\pm$ 1.2e-16	0.510 $\pm$ 0.024
LSTM	0.501 $\pm$ 1.2e-16	0.501 $\pm$ 1.2e-16	1.00$\pm$0	0.667 $\pm$ 1.2e-16	0.519 $\pm$ 0.035
RGU	0.549$\pm$0.052	0.534$\pm$0.036	0.917 $\pm$ 0.089	0.671 $\pm$ 0.007	0.636$\pm$0.052

8 Model performance with auxiliary tasks

Table S7: Performance of models, with auxiliary tasks, in regression task on random-split test data for peptides of lengths 6, 7, and 10.

Method	MolLogP	TPSA	MAE $\downarrow$	RMSE $\downarrow$	R^2 $\uparrow$	r $\uparrow$	ROC-AUC $\uparrow$
DMPNN	$\times$	$\times$	0.347 $\pm$ 0.014	0.512 $\pm$ 0.026	0.555 $\pm$ 0.034	0.750 $\pm$ 0.022	0.887 $\pm$ 0.011
	$\times$	$\checkmark$	0.346 $\pm$ 0.010	0.515 $\pm$ 0.020	0.549 $\pm$ 0.035	0.747 $\pm$ 0.024	0.885 $\pm$ 0.013
	$\checkmark$	$\times$	0.345 $\pm$ 0.009	0.516 $\pm$ 0.019	0.547 $\pm$ 0.038	0.746 $\pm$ 0.021	0.886 $\pm$ 0.009
	$\checkmark$	$\checkmark$	0.349 $\pm$ 0.014	0.529 $\pm$ 0.033	0.525 $\pm$ 0.050	0.738 $\pm$ 0.027	0.884 $\pm$ 0.014
GAT	$\times$	$\times$	0.385 $\pm$ 0.021	0.548 $\pm$ 0.028	0.489 $\pm$ 0.056	0.701 $\pm$ 0.041	0.854 $\pm$ 0.028
	$\times$	$\checkmark$	0.390 $\pm$ 0.022	0.580 $\pm$ 0.095	0.423 $\pm$ 0.171	0.670 $\pm$ 0.079	0.854 $\pm$ 0.020
	$\checkmark$	$\times$	0.386 $\pm$ 0.014	0.548 $\pm$ 0.022	0.490 $\pm$ 0.041	0.704 $\pm$ 0.028	0.854 $\pm$ 0.015
	$\checkmark$	$\checkmark$	0.387 $\pm$ 0.014	0.549 $\pm$ 0.023	0.488 $\pm$ 0.046	0.701 $\pm$ 0.033	0.857 $\pm$ 0.014
GCN	$\times$	$\times$	0.403 $\pm$ 0.022	0.562 $\pm$ 0.033	0.463 $\pm$ 0.055	0.686 $\pm$ 0.037	0.851 $\pm$ 0.016
	$\times$	$\checkmark$	0.407 $\pm$ 0.021	0.565 $\pm$ 0.030	0.458 $\pm$ 0.046	0.682 $\pm$ 0.032	0.848 $\pm$ 0.014
	$\checkmark$	$\times$	0.407 $\pm$ 0.028	0.573 $\pm$ 0.035	0.442 $\pm$ 0.058	0.671 $\pm$ 0.041	0.841 $\pm$ 0.021
	$\times$	$\times$	0.429 $\pm$ 0.021	0.583 $\pm$ 0.030	0.422 $\pm$ 0.058	0.660 $\pm$ 0.040	0.839 $\pm$ 0.023

$\times$: property not used as an auxiliary task; $\checkmark$: property used as an auxiliary task.

References

- [1] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [2] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
- [3] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. *CoRR*, abs/1704.01212, 2017.
- [4] Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang Jiang, and Mingyue Zheng. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of Medicinal Chemistry*, 63(16):8749–8760, 2020.
- [5] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, 2019.
- [6] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.
- [7] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.
- [8] Benson Chen, Regina Barzilay, and Tommi Jaakkola. Path-augmented graph transformer network, 2019.
- [9] Jeffrey L Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.
- [10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [11] Mirco Ravanelli, Philemon Brakel, Maurizio Omologo, and Yoshua Bengio. Light gated recurrent units for speech recognition. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(2):92–102, April 2018.
- [12] Garrett B. Goh, Charles Siegel, Abhinav Vishnu, Nathan O. Hodas, and Nathan Baker. Chemception: A deep neural network with minimal chemistry knowledge matches the performance of expert-developed qsar/qspr models, 2017.
- [13] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016.
- [14] Xiangxiang Zeng, Hongxin Xiang, Linhui Yu, Jianmin Wang, Kenli Li, Ruth Nussinov, and Feixiong Cheng. Accurate prediction of molecular properties and drug targets using a self-supervised image representation learning framework. *Nature Machine Intelligence*, 4(11):1004–1016, 2022.
- [15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [16] John S Delaney. Esol: Estimating aqueous solubility directly from molecular structure. *Journal of Chemical Information and Computer Sciences*, 44(3):1000–1005, 2004.
- [17] Ines Filipa Martins, Ana L. Teixeira, Luis Pinheiro, and Andre O. Falcao. A bayesian approach to in silico blood-brain barrier penetration modeling. *Journal of Chemical Information and Modeling*, 52(6):1686–1697, 2012.