JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

Open Access

pubs.acs.org/jcim

Article

ChemXTree: A Feature-Enhanced Graph Neural Network-Neural Decision Tree Framework for ADMET Prediction

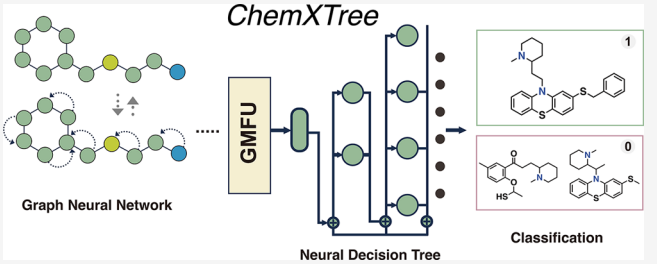
Yuzhi Xu, Xinxin Liu, Wei Xia, Jiankai Ge, Cheng-Wei Ju, Haiping Zhang, and John Z.H. Zhang*

Cite This: https://doi.org/10.1021/acs.jcim.4c01186

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: The rapid progression of machine learning, especially deep learning (DL), has catalyzed a new era in drug discovery, introducing innovative approaches for predicting molecular properties. Despite the many methods available for feature representation, efficiently utilizing rich, high-dimensional information remains a significant challenge. Our work introduces ChemXTree, a novel graph-based model that integrates a Gate Modulation Feature Unit (GMFU) and neural decision tree (NDT) in the output layer to address this challenge. Extensive evaluations on benchmark data sets, including MoleculeNet and eight additional drug databases, have demonstrated ChemXTree's superior performance, surpassing or matching the current state-of-the-art models. Visualization techniques clearly demonstrate that ChemXTree significantly improves the separation between substrates and nonsubstrates in the latent space. In summary, ChemXTree demonstrates a promising approach for integrating advanced feature extraction with neural decision trees, offering significant improvements in predictive accuracy for drug discovery tasks and opening new avenues for optimizing molecular properties.



INTRODUCTION

In recent decades, the development of machine learning has remarkably accelerated the pace of drug discovery.[1,2] This acceleration is primarily because machine learning, unlike traditional drug discovery methods, can quickly process large data sets to identify crucial ADMET (absorption, distribution, metabolism, excretion, and toxicity) properties of drug compounds.[3−5] By efficiently predicting factors like distribution coefficients, free energy, solubility, stability, etc., machine learning enhances the efficacy and safety assessment of potential drugs.[6,7] As one of the most rapidly evolving subsets of machine learning, deep learning (DL) has had a huge impact on molecular property prediction. In molecular property prediction, DL methods avoids the need for predefined features, a limitation in traditional machine learning approaches.[8−10] DL provides a direct mapping from input to output and categorizes its key methodologies for extracting molecular features into different types based on computational frameworks. Models that use visual grid encoding typically employ Convolutional Neural Networks (CNNs) to identify spatial features in molecules.[11−14] Sequence-based models utilize architectures like Recurrent Neural Networks (RNNs) and Transformers to interpret molecular notations.[15,16] Lastly, graph-based models apply Graph Neural Networks (GNNs) to understand the complex relationships between atoms and bonds in molecular structures.[17,18]

Recent advancements in deep learning (DL) for molecular property prediction have primarily focused on enhancing molecular representation capabilities.[11,19−22] While these approaches have shown superior feature extraction compared to traditional molecular fingerprints,[23,24] they have not consistently outperformed decision tree methods with simple molecular fingerprints on small data sets.[25−28] Consequently, researchers have explored combining tree-based models with DL-based extractors to improve performance in various fields, including molecular prediction.[29,30] Notable examples include XGraphBoost,[31,32] which enhanced D-MPNN's performance by replacing its FFN component with XGBoost, and similar approaches using Random Forest or XGBoost as output layers.[33−35] However, these combination models often lack end-to-end training capabilities due to the absence of gradient back-propagation in traditional decision trees.[36] To address this limitation, recent innovations have introduced Neural Decision Trees (NDTs).[36−38] NDTs use gradient descent to optimize split points in continuous feature spaces, potentially enhancing the model's ability to leverage complex information extracted by advanced DL methods.[39] While promising, developing molecular property models using NDTs poses greater challenges than simple integration with traditional

Received: July 8, 2024
Revised: October 18, 2024
Accepted: October 29, 2024
Published: November 5, 2024

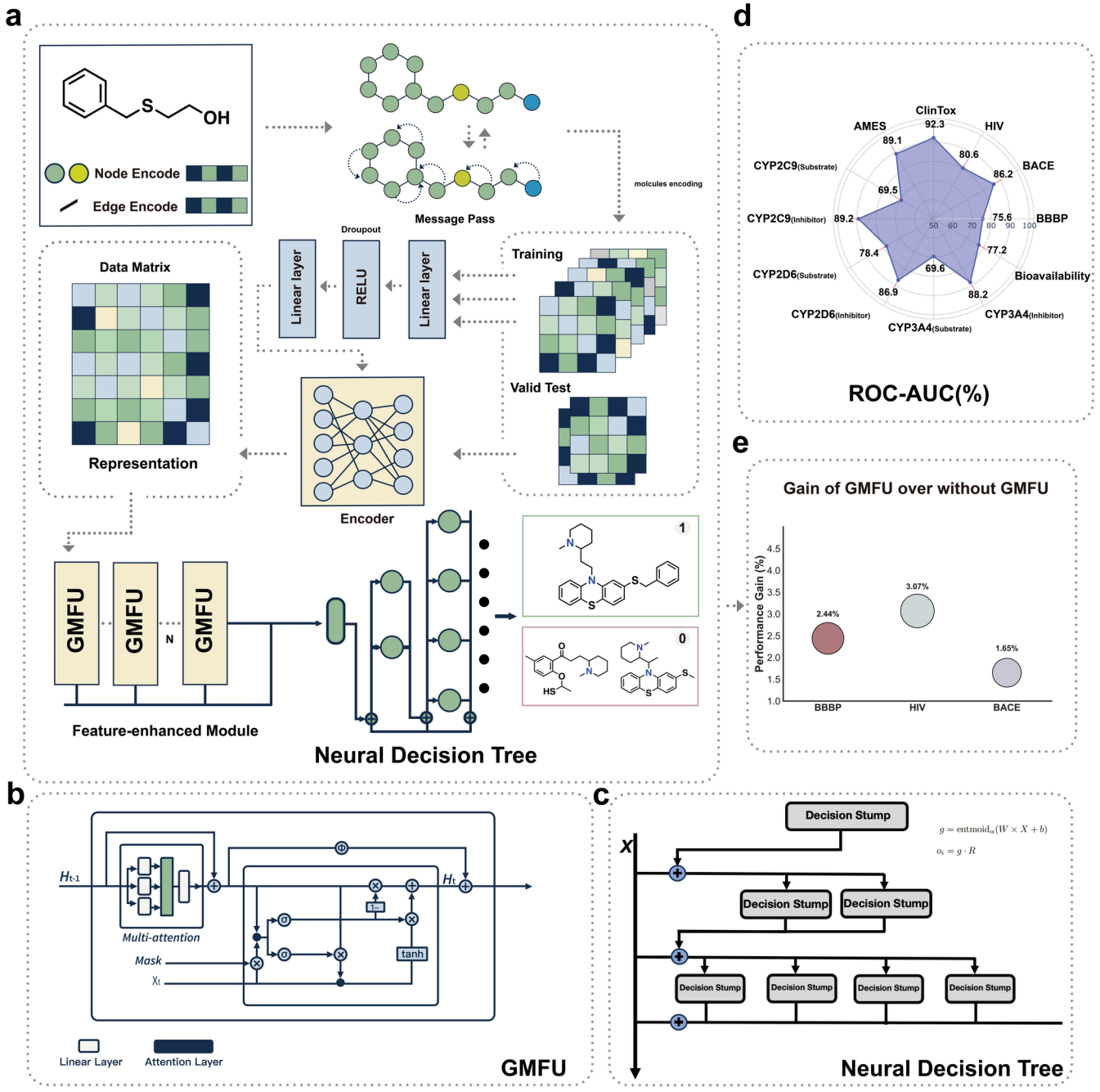


Figure 1. Workflow of ChemXTree: a combination model specifically engineered for the classification of small drug molecules. (a) The framework ChemXTree in training/testing process, including encoding module, GMFU modules, and the neural decision tree. (b) The structure of the GMFU module, illustrating the linear layers, attention layers, and the application of masks and activation functions. (c) The neural decision tree structure is (c), this is a soft decision process. (d) Performance of the ChemXTree across 12 benchmark data sets, showing ROC-AUC scores for different data sets like BBBP, BACE, HIV, and others in ChemXTree. (e) ChemXTree latent space shows a stronger clustering effect in distinguishing between active and inactive molecules in CYP2D6_Substrate testset, compared to other models.

decision trees, making it a relatively unexplored area in the field.

Here, we propose ChemXTree, a feature-enhanced GNN-NDT Framework for molecular property prediction. To boost the feature selection capability, we introduce a new module named the Gate Modulation Feature Unit (GMFU) to refine and select the most informative features, serving as a bridge between feature extraction module and NDT. Following this enhancement, a differentiable NDT is integrated into the model as the predictive output layer. Extensive evaluations were conducted on benchmark data sets from MoleculeNet and eight supplementary drug databases.[40] Results demonstrated that ChemXTree exhibits significant competitiveness with baseline models. Additionally, we conduct permutation experiments on the output layer and perform ablation studies on the GMFU, including the development of an LSTM-based variant, to assess their impacts on the overall model performance. Besides, in comparison to other representation methods, the latent space of ChemXTree's GMFU output has already acquired the crucial pharmacological information

https://doi.org/10.1021/acs.jcim.4c01186
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

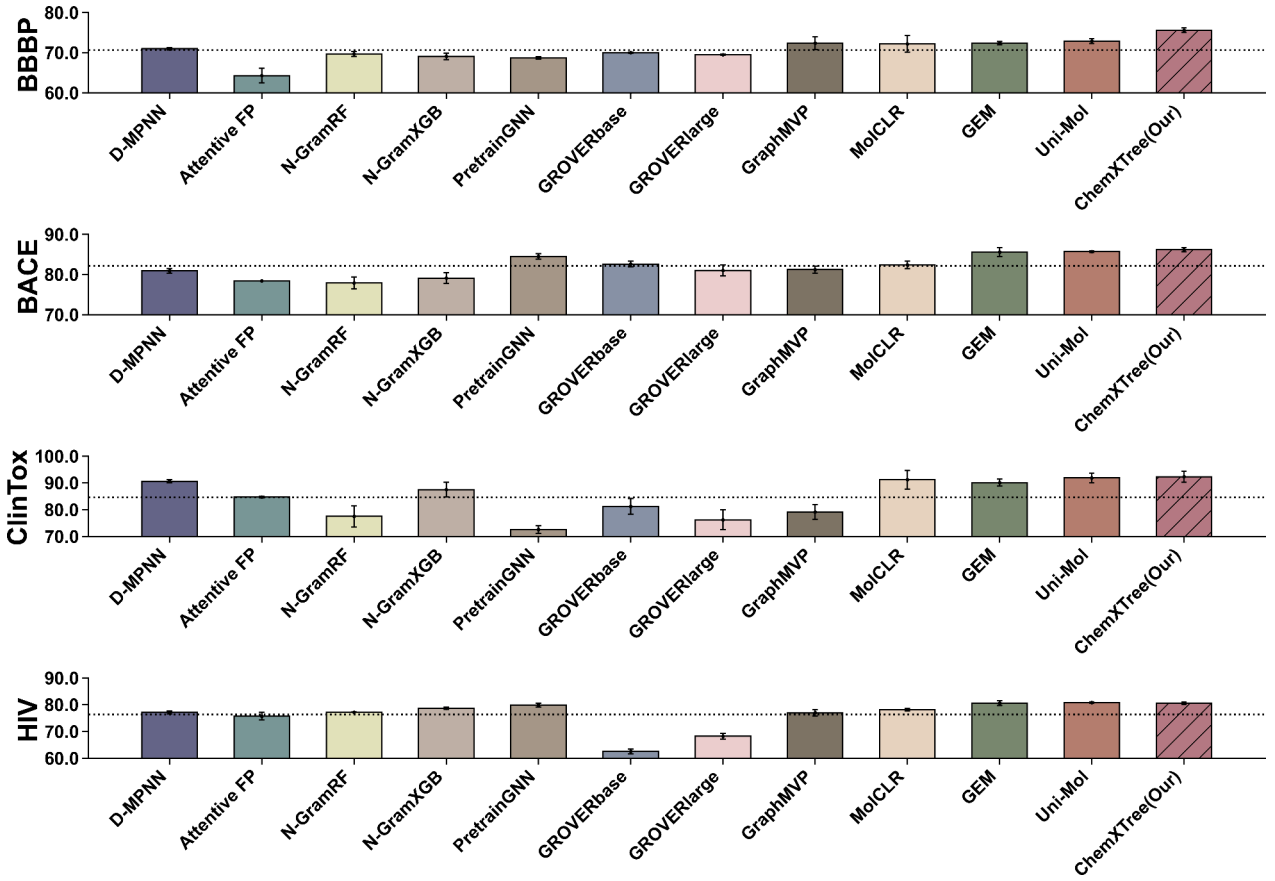


Figure 2. Performance of ChemXTree on MoleculeNet Classification Tasks. This chart compares the ROC-AUC scores of ChemXTree and other models on various MoleculeNet classification tasks, including BBBP, BACE, ClinTox, and HIV data sets. The horizontal axis denotes the models, while the vertical axis represents their ROC-AUC scores (%).

necessary for substrate identification. In general, we demonstrated that even without pretraining, the combination model, ChemXTree, could compete with the performance of existing state-of-the-art(SOTA) models in classification tasks in small data sets. This approach is promising for broader application in smaller data sets with limited drug-related data availability.

RESULTS AND DISCUSSION

ChemXTree Workflow. ChemXTree is specifically designed to address classification problems in small drug molecules. To achieve this, ChemXTree's architecture utilizes Graph Neural Networks (GNN) for encoding molecules. The encoded output is then processed through a specially designed module, known as GMFU, which focuses on optimized feature selection. Following this, ChemXTree uses a Neural Decision Tree to further enhance the model's classification capabilities.

To be more specific, as depicted in Figure 1a, ChemXTree initiates its process by first transforming the simplified Molecular Input Line Entry System (SMILES) of molecules into graphs. In these graphs, atoms, along with multiscale features such as charge and valency, serve as vertices (nodes), while chemical bonds act as edges. Subsequently, in the molecular representation learning stage, these initially encoded molecules are passed to Message Passing Neural Network (MPNN) for further processing and refining of the molecular encoding.[41] Upon completion of the encoding training, the MPNN equips with trained weights and then functions as an encoder, encoding the validation and test sets, respectively. With this method, the encoded representations of the molecules in training, validation, and testing sets are obtained. To enhance the robustness of the molecular classification, we employ an ensemble approach. Specifically, the final molecular encoding is obtained by summing the outputs of five independently trained models for each molecule.

With the encoded molecular representations, ChemXTree aims to intensify the identification of task-relevant features for downstream tasks. To achieve this, we employ a network architecture composed of a series of GMFUs. Inspired by Gated Recurrent Units (GRU) and other variants like Gated Adaptive Network using similar gated mechanism, GMFU is designed for further feature selection.[42−44] Specifically, each GMFU starts with self-attention on the input features, which are then fed into a gated structure equipped with reset and update gates to generate candidate feature representations (Figure 1b). The final output of the GMFU is computed by combining the previous hidden states with the current candidate features based on the update gate. Multiple GMFUs can be cascaded in a hierarchical fashion, where the output of one unit serves as the input for the subsequent unit. Different layers of GMFU engage in distinct feature selections, enabling the model to capture molecular representations at various levels of abstraction. This feature facilitates the integration of both global and local information into the final feature representation.

These features are subsequently fed into a differentiable neural decision tree for further employment (Figure 1c). To enable the latter optimization and backpropagation of

https://doi.org/10.1021/acs.jcim.4c01186
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Table 1. Performance of ChemXTree and Other Models Across BBBP, BACE, HIV, and ClinTox Datasets with Best ROC-AUC Score Denoted in Bold (Values in Parentheses Represent Standard Deviation)

Data sets	BBBP	BACE	HIV	ClinTox
D-MPNN[41]	71.0 (0.3)	80.9 (0.6)	77.1 (0.5)	90.6 (0.6)
Attentive FP[48]	64.3 (1.8)	78.4 (0.02)	75.7 (1.4)	84.7 (0.3)
N-GramRF[33]	69.7 (0.6)	77.9 (1.5)	77.2 (0.1)	77.5 (4.0)
N-GramXGB[33]	69.1 (0.8)	79.1 (1.3)	78.7 (0.4)	87.5 (2.7)
PretrainGNN[45]	68.7 (1.3)	84.5 (0.7)	79.9 (0.7)	72.6 (1.5)
GROVERbase[46]	70.0 (0.1)	82.6 (0.7)	62.5 (0.9)	81.2 (3.0)
GROVERlarge[46]	69.5 (0.1)	81.0 (1.4)	68.2 (1.1)	76.2 (3.7)
GraphMVP[49]	72.4 (1.6)	81.2 (0.9)	77.0 (1.2)	79.1 (2.8)
MolCLR[19]	72.2 (2.1)	82.4 (0.9)	78.1 (0.5)	91.2 (3.5)
GEM[50]	72.4 (0.4)	85.6 (1.1)	80.6 (0.9)	90.1 (1.3)
Uni-Mol[47]	72.9 (0.6)	85.7 (0.2)	**80. 8** (0.3)	91.9 (1.8)
ChemXTree(Our)	**75. 6** (0.6)	**86. 2** (0.5)	80.6 (0.5)	**92. 3** (0.8)

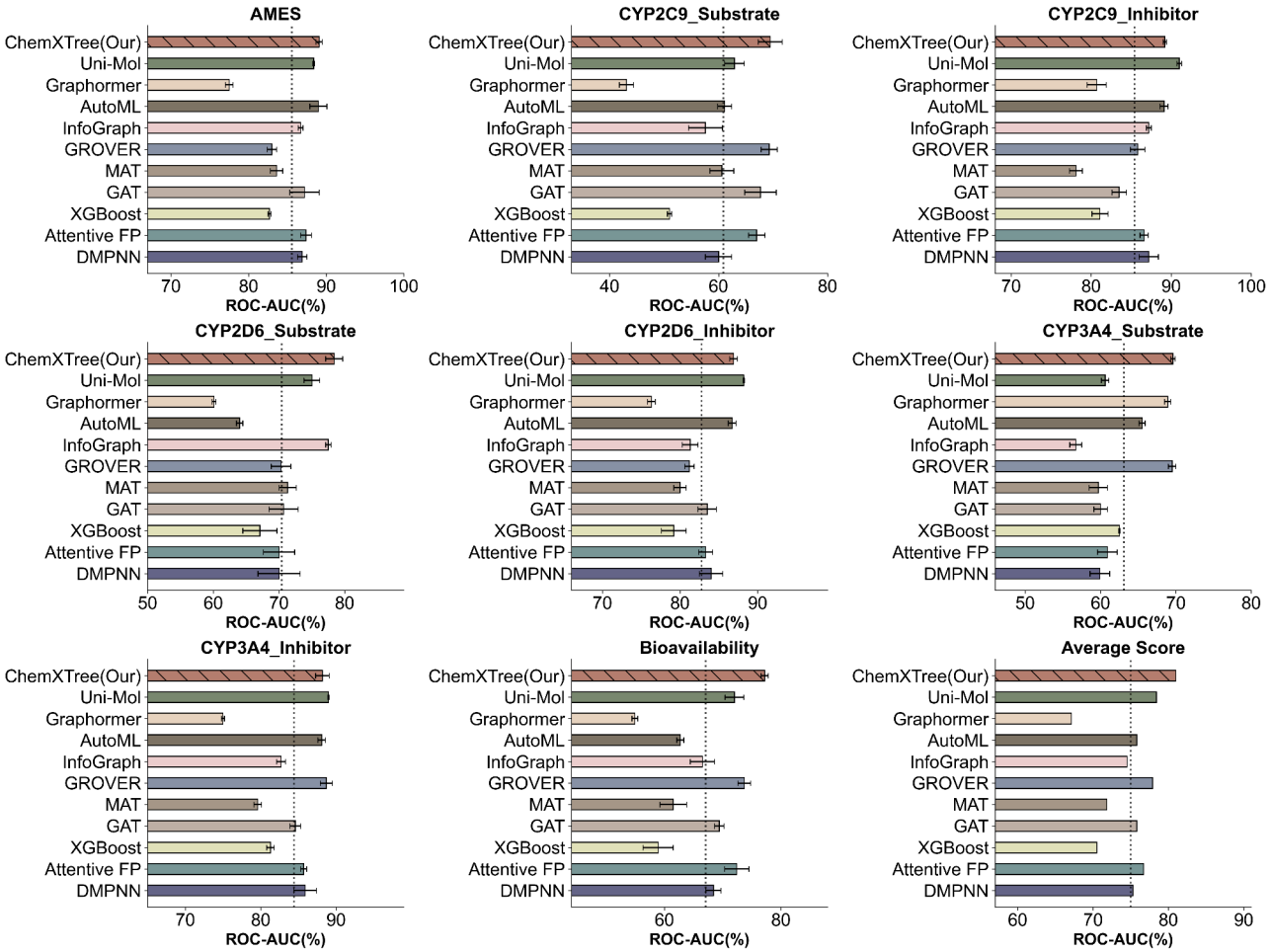


Figure 3. This comprehensive chart showcases the performance comparison of ChemXTree against various baseline models across a diverse range of drug benchmark data sets. The data sets encompass AMES, CYP2C9_Substrate, CYP2C9_Inhibitor, CYP2D6_Substrate, CYP2D6_Inhibitor, CYP3A4_Substrate, CYP3A4_Inhibitor, and Bioavailability. The horizontal axis represents the ROC-AUC percentage, while the vertical axis denotes the model names.

gradients, our decision tree employs soft decisions that output continuous probabilities, as opposed to traditional hard and deterministic decisions that only yield 0 or 1. We leverage an ensemble of different differentiable trees whose outputs are weighted. This configuration leads to a stable and precise binary prediction result and ensures our model's adaptability and effectiveness in handling high-dimensional molecular data spaces.

For more information about the architecture of ChemXTree, please see the Materials and Methods section.

MoleculeNet Benchmarking of ChemXTree. To comprehensively evaluate the ChemXTree, we utilized the MoleculeNet databases developed by Wu et al.[40] ChemXTree

https://doi.org/10.1021/acs.jcim.4c01186
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

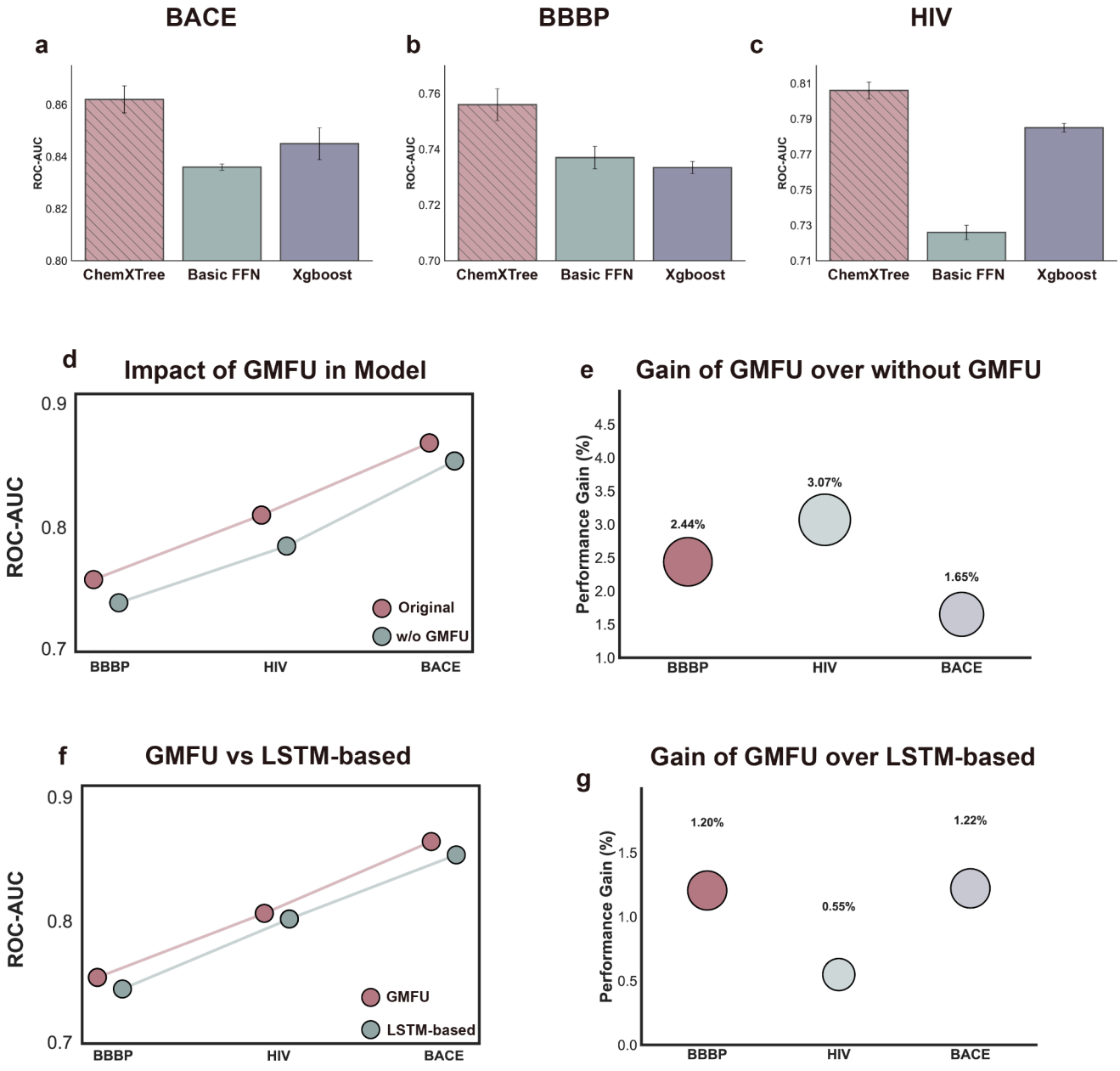


Figure 4. (a−c) Performance comparison of ChemXTree with Basic FFN and XGBoost output layers across the BACE, BBBP, and HIV data sets, with all models using the same molecular encoding. (d) Performance comparison of ChemXTree with and without GMFU module across BACE, BBBP, and HIV data sets. (e) Improvement in performance achieved by the GMFU module across BACE, BBBP, and HIV data sets. (f) Performance comparison of ChemXTree with GMFU module and LSTM-based module across BACE, BBBP, and HIV data sets. (g) Performance advantage of the GMFU module compared to the LSTM-based variant for BACE, BBBP, and HIV data sets.

is mainly developed for single-target classification. Due to ChemXTree's utilization of decision tree algorithms, transforming multitask classification problems into multiple binary classification tasks is a common approach. However, this transformation leads to increased computational costs and deployment difficulties. Considering our limited computational resources, we did not conduct tests on tasks involving multitask binary classifications. In this evaluation part, we adhered to the data processing and evaluation methods presented in previous work. After obtaining the optimal hyperparameters through Bayesian optimization and running the model three times, we conducted a comparison with various existing models.

As shown in Figure 2 and Table 1, ChemXTree performs comparably or slightly better than the SOTA models in all the data sets. Notably, ChemXTree excels in predicting tasks such as BBBP, BACE, and ClinTox. A potential factor could be ChemXTree's strategic feature selection combined with its tree-based output layer, offering an equilibrium between fitting the data and generalizing well, especially with small data sets. Specifically, compared to D-MPNN[41] where similar molecular feature extraction is applied, ChemXTree demonstrates a consistent trend of improvement across all data sets, boosting the ROC-AUC by 6% for both BBBP and BACE tasks. This indicates that combining tree-based models with other approaches can enhance the utilization of feature information

https://doi.org/10.1021/acs.jcim.4c01186
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

and significantly boost the performance of the GNN model. Moreover, models such as N-GramRF,[33] N-GramXGB, PretrainGNN,[45] GROVERlarge,[46] GROVERbase, MolCLR,[19] and Uni-Mol[47] undergo pretraining and fine-tuning process. In contrast, ChemXTree attains comparable results without this step, indicating the potential of its structure and algorithms even in the absence of pretraining.

Comprehensive Performance Evaluation of ChemXTree Across Diverse Drug Data Sets. For a comprehensive benchmark, we tested ChemXTree on 8 additional drug data sets, including AMES, CYP2C9_Substrate, CYP2D6_Substrate, CYP3A4_Substrate, CYP2C9_inhibitor, CYP2D6_inhibitor, CYP3A4_inhibitor, and Bioavailability. These data sets have been widely adopted in previous works as benchmarks.[51] Specifically, our data set selection considers AMES, CYP2C9_inhibitor, CYP2D6_inhibitor, and CYP3A4_inhibitor as large-scale data sets, and CYP2C9_Substrate, CYP2D6_Substrate, CYP3A4_Substrate, and Bioavailability as small-scale data sets, in order to cover diverse data characteristics and model capabilities. We compare our ChemXTree against 10 SOTA methods, spanning from conventional machine learning algorithms, e.g., XGBoost to emerging deep graph neural networks including DMPNN, AttentiveFP, Graph Attention Network (GAT), etc., as well as pretrained models like Uni-Mol and GROVER (GROVERbase is chosen as the representative of GROVER due to their performance in MoleculeNet). The detailed benchmark model information could be found in Materials and Methods. For comparison, we optimize hyperparameters and follow the same data set splitting and evaluation metrics (ROC-AUC) as MoleculeNet recommend.

As shown in Figure 3, our ChemXTree achieves superior average ROC-AUC scores across all 8 data sets compared to other methods, demonstrating its strong generalization capability. Specifically, ChemXTree attains the best ROC-AUC performance on the AMES, CYP2C9, CYP2D6, CYP3A4, and bioavailability data sets, with 89.1%, 69.5%, 78.4%, 69.6%, and 77.2% respectively. For the three larger data sets of CYP2C9 Inhibitor, CYP2D6 Inhibitor, and CYP3A4 Inhibitor, ChemXTree ranks second, only behind the pretrained Uni-Mol model and on par with GROVER. This is reasonable since Uni-Mol benefits from pretraining on external millions of molecular data, granting better generalization to represent unseen cases. In contrast, ChemXTree learns representations from scratch solely based on the training data yet still surpasses other methods without pretraining, verifying its modeling effectiveness. Overall, ChemXTree achieves an average ROC-AUC of 81.0% across the eight data sets, significantly outperforming all other methods and demonstrating its superiority and competitiveness. In summary, the benchmark experiments thoroughly prove ChemXTree's superiority and competitiveness in molecular property prediction tasks on both small- and large-scale data sets.

Ablation and Substitution Analysis in ChemXTree. To analyze ChemXTree's enhancements, we took the molecular encoding from the top-performing ChemXTree on BACE/BBBP/HIV data sets to conduct the ablation experiments by swapping its prediction output with a basic FFN and XGBoost. The basic FFN configuration was from Chemprop's default set and it was trained over 1000 epochs.[52] We conducted a grid search over the following hyperparameters: `batch_sizes` of {16, 32, 64} and `learning_rates` of {0.001, 0.002, 0.004}. For the XGBoost, we employed Bayesian optimization within a defined parameter space: `learning_rate` between 0.004 and 1.0, `max_depth` in the interval [4, 20], and `n_estimators` spanning [20, 400]. We incremented the Bayesian search iterations, starting from 30 and gradually increasing to 50, 200, and 300. Besides, the `lambda` and `alpha`, which are regularization terms, are in the range of [0,10]. For these two layer substitutions, we recorded the best performance model for the comprehensive analysis, respectively.

In Figure 4a−c, a comparison of three output layers using identical encoded inputs shows that ChemXTree outperforms both basic FFN and XGBoost in terms of classification efficacy. We computed the average ROC-AUC score of each output layers across three data sets, with the performance scores ranking as follows: ChemXTree > XGBoost > basic FFN. Specifically for the HIV data set (Figure 2C), which has a significant class imbalance with approximately 1:27 ratio of positives to negatives, tree-based approaches clearly outperform basic FFN. This can be attributed to the tree methods' branching mechanisms, which are highly effective at capturing nonlinear relationships within the data. Besides, while XGBoost does not surpass the basic FFN on the BBBP data set, its respective performance in other cases supports its viability as an effective substitute for the basic FFN layer in a range of computational scenarios.

To assess the effectiveness of the GMFU module in ChemXTree, we also conducted an ablation study to evaluate the effectiveness of GMFU. Employing identical inputs and optimal performance parameters, we compared the original ChemXTree model with variants lacking GMFU using identical inputs and optimal performance parameters across three data sets. In Figure 4d presented, "w/o GMFU" denotes the ChemXTree configuration excluding the GMFU module. The results revealed a decline in model performance for all three data sets when GMFU was removed. Figure 4e further illustrates the importance of the GMFU component for optimizing ChemXTree's performance: on the BBBP data set, GMFU provided a 2.44% improvement, while in the HIV and BACE data sets, it yielded gains of 3.07% and 1.65%, respectively. These results indicate that GMFU is an essential part of optimizing the ChemXTree model, as it enhances the feature selection function and improves the model's performance.

In addition to the GMFU, we explored a variant inspired by long short-term memory (LSTM) architecture.[53] To distinguish between these modules by mechanism, we refer to them as GMFU and LSTM-based, respectively. It is noteworthy that the model adaptation for the LSTM-based variant is nearly identical to the treatment of GMFU. As depicted in Figure 4f,g, the original ChemXTree outperforms its LSTM-based variant on all three data sets, achieving an average performance boost of 1%. In contrast to the LSTM-based approach, which employs a complex three-gate architecture for feature selection, the GMFU strategy adopts a more efficient two-gate design consisting solely of reset and update gates. This streamlined structure not only reduces the number of model parameters but also leads to a more memory-efficient architecture. As evidenced in Table S2, GMFU resulted in faster average training times compared with the LSTM-based variant.

Comparative Visualization of Different Molecular Representations in CYP2D6_Substrate Data Set. We use CYP2D6_Substrate testset for t-SNE analysis to visually compare three different representations: 2048-bit molecular

https://doi.org/10.1021/acs.jcim.4c01186
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Morgan fingerprints,[54] MPNN-encoded embeddings, and hidden layer representations from the ChemXTree GMFU module before feeding into neural decision tree[55] (Figure 5a–c). The t-SNE results revealed dispersed distributions for the Morgan fingerprints, indicating that Morgan fingerprints may not effectively capture the underlying chemical properties relevant to CYP2D6 substrate binding. In contrast, the MPNN embeddings showed some subtle clustering after t-SNE dimensionality reduction. Notably, the GMFU module representations displayed considerably tighter clustering after t-SNE dimensionality reduction. This suggests a superior ability to distinguish between molecules with varying efficacy in the high-dimensional space of CYP2D6 substrate binding.

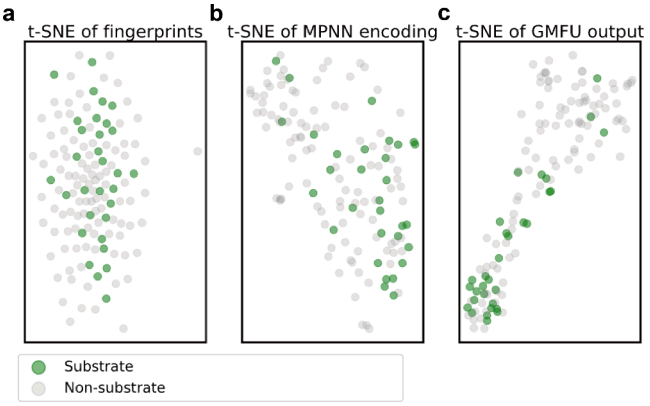


Figure 5. t-SNE visualization of CYP2D6_Substrate testset, showing 2048-bits Morgan Fingerprints, MPNN-encoded Embeddings, and ChemXTree hidden layer Representations. The plots in t-SNE were generated where ground truth labels 1 (substrate) and 0 (non-substrate) were denoted in green and gray, respectively.

CONCLUSION

In this work, we have proposed ChemXTree, a feature-enhanced GNN-NDT Framework for molecular property prediction in drug discovery. We comprehensively evaluated ChemXTree on both the MoleculeNet benchmark and an additional eight drug discovery data sets. The results demonstrated SOTA performance, with ChemXTree outperforming current top models on 8 out of the 12 data sets. Ablation studies and model substitution experiments further validated the effectiveness of the GMFU and neural decision tree modules, highlighting their contributions to feature selection and classification. t-SNE analysis on the CYP2D6 substrate data set indicated that the GMFU module significantly enhanced the separation between substrates and nonsubstrates, indicating its ability to capture key pharmacophoric features.

In the ChemXTree, we adopt a simple representation method based on MPNN to highlight the significance of the nonrepresentation component in ChemXTree. Therefore, to address the current limitations and elevate the application to a higher level, our future work will focus on GNN pretraining representation, feature extraction, and balancing input complexity with computational requirements. Additionally, simplifying the workflow and reducing the number of parameters are also key directions for optimization. In summary, ChemXTree presents a competitive and innovative end-to-end graph representation learning framework for drug discovery, particularly excelling in small sample learning scenarios. ChemXTree holds great promise to accelerate the identification of novel drug candidates and contribute to the advancement of computational drug discovery.

MATERIALS AND METHODS

Message Passing Neural Networks (MPNNs) efficiently capture the molecular graph structures of small organic molecules through nodes and edges. The stochastic nature of their training renders the outputs highly sensitive to hyperparameter settings and variability in training. Although MPNNs offer linear features as an alternative to conventional vertex and edge representations, reliably determining the optimal output from multiple training iterations and evaluating the significance of specific features for particular tasks continue to be challenging.

Decision trees are widely recognized for their capability to partition a feature space into distinct and nonoverlapping regions, based on specific feature values. In statistical learning context, for a given feature's random variable X, the decision tree applies a set of "if-then" rules based on conditional probabilities $P(Y|X)$, showcasing the classification outcomes of these rules. These rules facilitate a mutually exclusive and complete classification of instances into positive or negative categories. In the context of drug discovery, the effectiveness to differentiate between classes is crucial. Hence, binary decision tree models offer an inherent efficiency in modeling binary-class drug molecule data sets.

Ideally, a training data set contains a minimal subset including all necessary "if-then" rules for classification. The goal for a classifier is to accurately identify instances within this subset. In the context of decision trees, Information Gain (IG) is commonly employed as a criterion for feature selection and tree pruning, identifying this minimal subset. However, IG's deterministic nature, depending on data set composition, raises the risk of overfitting in high-dimensional spaces, such as molecular fingerprints. It exposes the limitations of decision trees in adapting loss dynamically compared to deep learning models with backpropagation machanism.

The Gate Recurrent Unit (GRU) stands out in Natural Language Processing (NLP) models for its gating mechanisms, which simplifies configuration and boosts denoising performance with fair computational demands. However, molecular representations typically lack temporal, hierarchical, and sequential characteristics and often show sparsity and uniformity.

Building on these insights, we designed ChemXTree, which integrates a novel fingerprint strategy based on MPNN with a data augmentation architecture, Gate Modulation Feature Unit (GMFU). This new fingerprint representation offers remarkable robustness, while GMFU merges the strengths of boosting trees and gated mechanisms, further enhanced by gradient back-propagation, to improve adaptability and control in molecular property prediction.

Data Sets. MoleculeNet includes 16 data sets covering various scientific domains like Quantum Mechanics, Physical Chemistry, and Biophysics. These data sets are utilized for benchmarking machine learning models in the respective fields. Among the 16 data sets, there are several binary classification tasks in our comparison:

(1) BBBP: The Bloodbrain barrier penetration (BBBP) data set. BBBP data set evaluates the permeability of small molecules across the blood-brain barrier. It comprises 2039

https://doi.org/10.1021/acs.jcim.4c01186
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

validated molecules, featuring a positive-to-negative sample ratio of approximately 3.2:1.

(2) BACE: The BACE data set provides a comprehensive array of both quantitative (IC50 values) and qualitative (binary classification labels) data, evaluating the binding efficacy of inhibitors aimed at human β-secretase 1 (BACE-1). The data set comprises 1513 molecules, with an inactive-to-active label ratio of approximately 1.23:1.

(3) HIV: The HIV data set originates from the Drug Therapeutics Program's AIDS Antiviral Screen. It contains data on 41,127 molecules and their efficacy in inhibiting HIV replication. The data set features a ratio of inactive to active molecules of approximately 27.5:1.

(4) ClinTox: The ClinTox data set provides a comparative analysis between FDA-approved drugs and molecules that failed in clinical trials due to toxicity concerns. It includes 1478 molecules, with ratios of FDA-approved to unapproved drugs at approximately 14.9:1 and clinical toxicity-positive to -negative molecules at approximately 12.2:1.

For benchmarking on additional 8 ADMET data sets. Our ADMET data sets are derived from previous works.[5,51] We thank the Therapeutics Data Commons for providing access to these valuable resources.

Ames Mutagenicity (Toxicity): The Ames Mutagenicity data set focuses on the potential mutagenic effects of molecules using the Ames test, which identifies substances that may damage DNA.[56] In this work, we refer to the Ames Mutagenicity data set as AMES.

CYP2C9, CYP2D6, and CYP3A4 Substrate (Metabolism): These data sets examine the roles of CYP450 enzymes 2C9, 2D6, and 3A4 in metabolizing endogenous and foreign molecules. The tasks of these data sets are predicting whether molecules act as substrates for these enzymes.[57]

CYP2C9, CYP2D6, and CYP3A4 inhibitor (Metabolism): In contrast to the Substrate data sets, these data sets primarily aim to predict whether molecules inhibit the metabolic activity of CYP450 enzymes 2C9, 2D6, and 3A4.[58]

Bioavailability (Absorption): This data set represents the rate and extent at which active molecules are absorbed from a drug product and become effective at the site of action. The data set is used for predicting the activity of bioavailability.[59]

Spilt and Evaluation Metric. Scaffold splitting is commonly adopted in cheminformatics as a way to better evaluate the model generalization. While random splitting offers simplicity in implementation, it often inadequately represents the model's ability to generalize to novel data. Scaffold splitting, on the other hand, aims to divide the data set into subsets based on structurally distinct molecules. This approach poses a greater challenge to learning algorithms compared with random splitting, thereby providing a more robust measure of generalization.

In the case of data sets from MolecuNet, we follow a previously established 8:1:1 ratio for splitting the data into training sets, validation sets, and test sets. For data sets that are not part of MoleculeNet, we employ a 7:1:2 scaffold spliting. To establish a unified standard, we adhere to the guidelines set by MoleculeNet and other previous works, using ROC-AUC as the evaluation metric across all our experiments. This allowed direct comparison to existing benchmarks and studies. Adhering to MoleculeNet's criteria, PRC-AUC is employed for data sets with a positive sample rate below 2%; otherwise, ROC-AUC is preferred. Therefore, we use ROC-AUC for evaluation in these data sets.

Comparison Models. N-Gram: N-Gram model introduced by Liu et al. offers a simple, unsupervised approach to represent molecules by embedding vertices and assembling them in short walks within the graph.[33]

MolCLR: MolCLR proposed by Wang et al. is a framework for molecular representation learning that applies contrastive learning to encode molecular structures to potentially capture the underlying patterns and relationships in molecular data.[19]

GraphMVP (Graph Multiview Pretraining): GraphMVP developed by by Liu et al. represents a pretraining method for graph neural networks, leveraging self-supervised learning to exploit the relationships and consistencies between 2D topological structures and 3D geometric views.[49]

PretrainGNN: PretrainGNN is a new strategy developed by Hu et al. for pretraining Graph Neural Networks (GNNs) that simultaneously trains on individual nodes and entire graphs, enhancing both local and global representations.[45]

GEM (Geometry-enhanced Molecular representation): GEM improves molecular property prediction by integrating molecular geometry into a Graph Neural Network (GNN). It utilizes a specialized GeoGNN architecture to simultaneously consider the influence of atoms, bonds, and bond angles, thereby crafting a more detailed representation of molecules.[50]

AttentionFP: AttentionFP is a model proposed by Xiong et al. This model employs a graph attention mechanism to focus on the most crucial parts of the molecular structure, particularly the nonlocal intramolecular interactions.[48]

GAT (Graph Attention Networks): GAT is a class of graph neural networks distinguished by their utilization of attention mechanisms enabling GATs to selectively prioritize and integrate information from adjacent nodes.[60]

MAT (Molecular Attention Transformer): MAT developed by Maziarka et al. is based on the Transformer designed for molecular representation. It enhances the self-attention mechanisms of the Transformer by incorporating interatomic distances and molecular graph structures, allowing for a deeper understanding of molecular features.[61]

GROVER: This is a pretrained model designed by Rong et al. employing a self-supervised message passing transformer architecture, integrating message passing networks with a transformer framework to effectively learn molecular representations from unlabeled data. GROVER has two sizes: GROVERbase and GROVERlarge with different hidden layers.[46]

InfoGraph: InfoGraph is an graph representation learning method proposed by Sun et al. It maximizes mutual information between graph-level and substructure representations, enabling robust graph-level learning.[62]

Automated Machine Learning (AutoML): AutoML denotes the process automation for selecting, fine-tuning, and training machine learning models. In this work, the input is sourced from a 2048-bit Morgan Fingerprints, followed by the automatic fine-tuning of a Feedforward Neural Network (FFN).[63]

Graphormer: Graphormer is a novel architecture for molecular property prediction proposed by Ying et al. in 2021. It combines graph neural networks (GNNs) with the Transformer structure widely used in natural language processing.[64]

Uni-Mol: Zhou et al. have developed Uni-Mol, a 3D molecular representation learning framework that incorporates a comprehensively pretrained model, adept at processing a

https://doi.org/10.1021/acs.jcim.4c01186
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

broad spectrum of molecular and protein pocket data for diverse molecular representation tasks.[47]

Among these models, N-gram, PretrainGNN, GROVER, GraphMVP, MolCLR, GEM, and Uni-Mol are pretrained model and others are designed without pretrained (namely, fine-tuned model).

Input and Representation. To make the representation simpler, we applied MPNN to encode molecule graphs, which includes vertice and edges infomation, into numerical linear embeddings. For binary classification tasks, we have an input data set $\mathcal{D}$ mapping from feature space $\mathcal{X}$ to the output space $\mathcal{Y}$,

$$\mathcal{D} = (\mathcal{X}, \mathcal{Y}) = \{(\vec{x_i}, y_i)\}_{i=1}^{S}, \quad \mathcal{Y} = \{0, 1\} \tag{1}$$

whereas the ith molecule's representation vector $\vec{x_i} \in \mathbb{R}^d$ is a d-featured array corresponding to a target label $y_i \in \mathcal{Y}$. Initialization involves setting model parameters to reflect the average target value from the training data.

Message Passing Neural Networks Module. Our Message Passing Neural Networks (MPNN) Module is developed from the Chemprop framework and has undergone fine-tuning. To be more specific, as with many existing works on molecular graph representation, the initial required input is canonical SMILES without atom mapping. In our MPNN module, atoms and bonds of the molecules correspond to vertices V and edges E in the graph structure, respectively. We encode the features of the atoms (atomic numbers, numbers of bonds, formal charges, chirality, quantity of hydrogen, hybridization, aromaticity of the atom, and atomic mass) and the features of the bonds (types, in conjugation/ring, stereochemical information) to obtain the initial vectors. Therefore we can get

$$(\mathcal{V}_\alpha | \alpha \in V) \text{ and } \{\mathcal{E}_{\alpha\beta} | (\alpha, \beta) \in \mathrm{E}\}$$

Furthermore, the initial directed edge features are obtained by connecting the atomic features of the first atom of the bond to the corresponding undirected bond feature $\mathcal{E}_{\alpha,\beta}$. Here, $\phi()$ is a simple function of concatenation.

$$\mathcal{E}'_{\alpha,\beta} = \phi(\mathcal{V}_\alpha, \mathcal{E}_{\alpha,\beta})$$

Subsequently, the initialized features are passed into the MPNN. Herein, the edge features are first processed through a simple network layer endowed with a ReLU activation function and learnable weights W_i, where $W_i \in \mathbb{R}^{h \times h_i}$. In our model, the values $h^0_{\alpha,\beta} = 300$ and $\mathcal{E}'_{\alpha,\beta} = 147$ are taken. Therefore:

$$h^0_{\alpha,\beta} = \mathrm{ReLU}(\mathrm{W}_i \mathcal{E}'_{\alpha,\beta})$$

The directed edge attributes are subsequently updated through three iterative rounds of message passing, based on the local topology:

$$h^{(3)}_{\alpha,\beta} = \mathrm{ReLU}\left(\mathrm{h}^0_{\alpha,\beta} + \mathrm{W}_h \sum_{\mathrm{i} \in \mathrm{N}(\alpha)\setminus\beta} \mathrm{h}^t_{vi}\right)$$

where $W_h \in \mathbb{R}^{h \times h}$ and Wo $\in \mathbb{R}^{h \times h_o}$

Let us denote

$$\mathcal{X} = \phi\left(\mathcal{V}_\alpha \sum_{\beta \in N(\alpha)} h^3_{\beta\alpha}\right)$$

With this definition, the final hidden states can be expressed, which are then aggregated to form the atomic embeddings, as follows:

$$h_\alpha = \mathrm{ReLU}(\mathrm{W}_0 \mathcal{X})$$

Therefore, the complete atomic embedding can be represented as

$$h_{\mathrm{embed}} = \sum_{\alpha \in V} h_\alpha$$

Notably, we apply a bagging strategy to MPNN-generated fingerprints, each of size 5, to decrease the error in the graph-to-vector transformation, thereby emphasizing similar features while still preserving less prominent ones. This approach ensures that subsequent learners have the opportunity to evaluate the importance of these less focused features.

Gate Modulation Feature Unit. The attention component in the Gate Modulation Feature Unit (GMFU) operates as follows:

$$Q = hW_Q, \quad K = hW_K, \quad V = hW_V \tag{2}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{F \times d}$, F is the dimension of input features, and d is the dimension for each attention head. Q, K, and V are split along the last dimension d into n_{heads} different heads.

$$\alpha = \left[\mathrm{Softmax}\left(\frac{Q_1 K_1^{\mathrm{T}}}{\sqrt{d}}\right)V_1, \mathrm{Softmax}\left(\frac{Q_2 K_2^{\mathrm{T}}}{\sqrt{d}}\right)V_2, \ldots, \mathrm{Softmax}\left(\frac{Q_n K_n^{\mathrm{T}}}{\sqrt{d}}\right)V_n\right] W_O$$

where, $n = n_{\mathrm{heads}}$, $W_O \in \mathbb{R}^{(n_{\mathrm{heads}} \cdot d) \times F}$.

After application of the self-attention mechanism, the feature vectors are passed into a gate control mechanism that plays a crucial role in the process of feature selection. This gate control mechanism has been adapted from GRU and GANDALF.

Inspired by previous works that have effectively utilized sparsity in neural networks, this work also adopts the t-softmax activation function proposed by Baazy et al. in 2023 for feature masking. Specifically, the mask M_n is generated using the t-softmax function applied to a learnable parameter vector F_n and a tuning parameter t, as shown in the equation M_n = t-softmax(F_n, t). This mask is then used to perform an element-wise multiplication with the input features X, resulting in the masked features δ_t as described in the equation $\delta_t = M_n \odot X$. Also, priors are introduced by initializing the masks using Beta Distribution: $F_n \sim \mathrm{Beta}(\alpha_n, \beta_n)$, where α_n and β_n are drawn from uniform distributions: $\alpha_n \sim \mathrm{Uniform}(0.5, 10)$, $\beta_n \sim \mathrm{Uniform}(0.5, 10)$.

Similar to the standard GRU and GANDALF, we set the corresponding reset gate r_t and update gate z_t as follows:

$$z_t = \sigma(W_z \cdot [\alpha, \delta_t] + b_z)$$

$$r_t = \sigma(W_r \cdot [\alpha, \delta_t] + b_r)$$

σ is the sigmoid activation function, which ensures the output is in the range of [0,1]. With these gates, we can further specify the hidden state H_t and the candidate feature $\tilde{H}_t$. They are defined as

https://doi.org/10.1021/acs.jcim.4c01186
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

$$\tilde{H}_t = \tanh(W\cdot[r_t \odot \alpha, x_t] + b),$$
$$H_t = (1 - z_t) \odot \alpha + z_t \odot \tilde{H}_t) + \Phi\alpha$$

where Φ is a parameter in the range of [0,1]. The default value of Φ in the model is 0.05.

Differentiable Neural Decision Tree. We input the features processed through GMFU into a differentiable binary decision tree. For such a decision tree, its role is to transform a k-dimensional input into a 2D-dimensional output. Although traditional decision trees have advantages like ease of deployment and straightforward algorithms, their methods are nondifferentiable. Therefore, definite soft decision binary trees are considered in this work.

Inspired by prior research,[44,65] the Soft Binning Function g and the decision stump o_i in this work serve as specialized counterparts to the splitting criterion and the decision node found in traditional decision trees, respectively.

$$g = \text{entmoid}_\alpha(\mathrm{W} \times \mathrm{X} + \mathrm{b})$$
$$o_i = g\cdot R$$

where $W \in \mathbb{R}^{d\times 2}$, $b \in \mathbb{R}^{d\times 2}$

Definite decision binary trees in the model utilize all available features for each split through a linear combination of nonlinear functions. A learnable feature mask $M \in \mathbb{R}^{d'}$ is introduced to efficiently combine the outputs o_i, allowing for scalability and comprehensive feature consideration.

$$o = \left[\sum_{i=0}^{\tilde{d}} M_i \times R_i^L, \quad \sum_{i=0}^{\tilde{d}} M_i \times R_i^R\right]$$

where $\tilde{d}$ is the dimensions of the learned feature representation from GFLUs and D is the depth of the tree.

In the Probably Approximately Correct (PAC) learning framework, if a differentiable binary decision tree of a fixed depth (i.e., within a polynomial time complexity) can evidently perform better than random guessing, then it is qualified as an effective learner. To further enhance this, it is feasible to rationally select a boosting algorithm tailored to this context.[66] Inspired by this theorem, we boosted our finite neural decision tree with specific tricks. Unlike classical boosting techniques, we chain the trees sequentially, with each tree's output feeding into the next.[44,67,68] Formally the output for the ith tree follows

$$O_i = \mathcal{F}_i([\mathcal{H}; O_{i-1}])$$

where $\mathcal{H} \in \mathbb{R}^{\tilde{d}}$, $O_{i-1} \in \mathbb{R}^{2^D}$ and [;] represents a concatenation operation.

Denoting the previously described tree as t_i, the boosting tree ensemble is expressed as

$$\mathcal{T} = \{t_i\}_{i=1}^T$$

Hence, the set of leaf responses O from $\mathcal{T}$ is

$$O = \{O_i\}_{i=1}^T, \text{ where each } O_i \in \mathbb{R}^{2^D}$$

After scaling with multihead attention, where $Q, K, V \in \mathbb{R}^{T\times 2^D}$, we obtain the leaf responses $\tilde{O}$,

$$\tilde{O} = \left[\text{Softmax}\left(\frac{Q_1K_1^T}{\sqrt{d}}\right)V_1, \text{Softmax}\left(\frac{Q_2K_2^T}{\sqrt{d}}\right)V_2, \ldots, \text{Softmax}\left(\frac{Q_nK_n^T}{\sqrt{d}}\right)V_n\right] \quad (3)$$

passes through T distinct output linear layers, corresponding to each tree, resulting in the desired output vector. Then we denote the set of linearly transformed outputs by $\mathcal{Y}$, where

$$\mathcal{Y} = \{y_i\}_{i=1}^T, \text{ where } y_i = \mathrm{W}_i\tilde{O}_i + \mathrm{b}_i$$

and for each tree, W_i and b_i are learnable parameters which transforms the leaf response vector into the desired output. The final prediction is formulated as an estimator of the targets, represented by

$$\hat{y} = \sigma\left(\sum_{i=1}^{T} \eta_i y_i\right) \quad (4)$$

Training. The ChemXTree and baseline models were trained on the NYU Greene and NYUAD Jubail high-performance computing (HPC) clusters. Specifically, we utilized NVIDIA's A100/H100 GPUs for our experiments in most small data sets. For large size data sets, we escalated our computational capacity by deploying dual GPUs. In every experiment conducted, drawing inspiration from the parameter tuning practices customary in traditional machine learning, we harnessed the prowess of Bayesian optimization. This method was utilized to meticulously traverse the parameter space comprising the learning rate, batch size, tree depth, tree breadth, the count of heads in the multihead attention mechanism within GMFU, and the quantity of GRUs, alongside the dropout rate. This meticulous exploration was performed through 100 Bayesian searches in each experiment. In the ablation experiment and comparative experiment in ADMET data sets, we fine-tuned key parameters like learning rate, epoch, and batch size for each model. For the detailed hyperparameters, please see the Supporting Information.

All the benchmark data can be found in the https://moleculenet.org/ and https://tdcommons.ai/. Also, the code and the data sets from section 2.6 for the models and results can be found in the codeocean platform https://codeocean.com/capsule/2818241/tree.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.4c01186.

Comparative analysis of ROC-AUC scores, dataset summary, analysis of feature importance, parameter specification, number of model parameters, and model optimization (PDF)

■ AUTHOR INFORMATION

Corresponding Author

John Z.H. Zhang − *Shanghai Frontiers Science Center of Artificial Intelligence and Deep Learning and NYU-ECNU Center for Computational Chemistry, NYU Shanghai, Shanghai 200062, China; Department of Chemistry, New York University, New York, New York 10003, United States; Faculty of Synthetic Biology, Shenzhen Institute of Advanced*

https://doi.org/10.1021/acs.jcim.4c01186
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Technology, Shenzhen 518055, China; Shanghai Engineering Research Center of Molecular Therapeutics and New Drug Development, School of Chemistry and Molecular Engineering, East China Normal University, 200062 Shanghai, China; orcid.org/0000-0003-4612-1863; Email: john.zhang@nyu.edu

Authors

Yuzhi Xu − *Shanghai Frontiers Science Center of Artificial Intelligence and Deep Learning and NYU-ECNU Center for Computational Chemistry, NYU Shanghai, Shanghai 200062, China; Department of Chemistry, New York University, New York, New York 10003, United States;* orcid.org/0000-0002-3325-5427

Xinxin Liu − *Department of Computer and Information Science, University of Pennsylvania, Philadelphia, Pennsylvania 19104, United States; Department of Materials Science and Engineering, University of Pennsylvania, Philadelphia, Pennsylvania 19104, United States*

Wei Xia − *Shanghai Frontiers Science Center of Artificial Intelligence and Deep Learning and NYU-ECNU Center for Computational Chemistry, NYU Shanghai, Shanghai 200062, China; Department of Chemistry, New York University, New York, New York 10003, United States*

Jiankai Ge − *Chemical and Biomolecular Engineering, University of Illinois at Urbana−Champaign, Urbana, Illinois 61801, United States;* orcid.org/0000-0002-7370-2797

Cheng-Wei Ju − *Pritzker School of Molecular Engineering, The University of Chicago, Chicago, Illinois 60615, United States;* orcid.org/0000-0002-2250-8548

Haiping Zhang − *Faculty of Synthetic Biology, Shenzhen Institute of Advanced Technology, Shenzhen 518055, China;* orcid.org/0000-0003-2133-1768

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.4c01186

Author Contributions

○Y.X., X.L., and W.X. contributed equally to this work.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (Grant nos. 21933010, 22250710136, and 22333006). We thank Ms. Bihui Guo from the University of Southampton for valuable discussions and contributions to the figures of our manuscript. We sincerely thank the High-Performance Computing (HPC) resources at NYU Abu Dhabi and Greene at New York University, as well as the dedicated staff and their technical support.

■ REFERENCES

(1) Tian, S.; Wang, J.; Li, Y.; Li, D.; Xu, L.; Hou, T. The application of in silico druglikeness predictions in pharmaceutical research. *Adv. Drug Delivery Rev.* **2015**, *86*, 2−10.

(2) Shen, C.; Ding, J.; Wang, Z.; Cao, D.; Ding, X.; Hou, T. From machine learning to deep learning: Advances in scoring functions for proteinligand docking. *WIREs Comput. Mol. Sci.* **2020**, *10*, e1429.

(3) Zhang, X.; Mao, J.; Wei, M.; Qi, Y.; Zhang, J. Z. J. Hergspred: Accurate classification of herg blockers/nonblockers with machine-learning models. *J. Chem. Inf. Model.* **2022**, *62*, 1830−1839.

(4) Pan, X.; Wang, H.; Li, C.; Zhang, J. Z.; Ji, C. MolGpka: A web server for small molecule p K a prediction using a graph-convolutional neural network. *J. Chem. Inf. Model.* **2021**, *61*, 3159−3165.

(5) Cheng, F.; Li, W.; Zhou, Y.; Shen, J.; Wu, Z.; Liu, G.; Lee, P. W.; Tang, Y. admetSAR:a comprehensive source and free tool for assessment of chemical ADMET properties. *J. Chem. Inf. Model.* **2012**, *52*, 3099−3105.

(6) Zhang, D.; Xia, S.; Zhang, Y. Accurate prediction of aqueous free solvation energies using 3d atomic feature-based graph neural network with transfer learning. *J. Chem. Inf. Model.* **2022**, *62*, 1840−1848.

(7) Pan, X.; Zhao, F.; Zhang, Y.; Wang, X.; Xiao, X.; Zhang, J. Z.; Ji, C. MolTaut: A Tool for the Rapid Generation of Favorable Tautomer in Aqueous Solution. *J. Chem. Inf. Model.* **2023**, *63*, 1833−1840.

(8) Ignacz, G.; Szekely, G. Deep learning meets quantitative structureactivity relationship (QSAR) for leveraging structure-based prediction of solute rejection in organic solvent nanofiltration. *J. Membr. Sci.* **2022**, *646*, No. 120268.

(9) Tang, B.; Kramer, S. T.; Fang, M.; Qiu, Y.; Wu, Z.; Xu, D. A self-attention based message passing neural network for predicting molecular lipophilicity and aqueous solubility. *J. Cheminform.* **2020**, *12*, 1−9.

(10) Zhang, Y.-F.; Wang, X.; Kaushik, A. C.; Chu, Y.; Shan, X.; Zhao, M.-Z.; Xu, Q.; Wei, D.-Q. SPVec: a Word2vec-inspired feature representation method for drug-target interaction prediction. *Front. Chem.* **2020**, *7*, 895.

(11) Zeng, X.; Xiang, H.; Yu, L.; Wang, J.; Li, K.; Nussinov, R.; Cheng, F. Accurate prediction of molecular properties and drug targets using a self-supervised image representation learning framework. *Nat. Mach. Intell.* **2022**, *4*, 1004−1016.

(12) Yi, J.; Wu, C.; Zhang, X.; Xiao, X.; Qiu, Y.; Zhao, W.; Hou, T.; Cao, D. MICER: a pretrained encoderdecoder architecture for molecular image captioning. *Bioinformatics* **2022**, *38*, 4562−4572.

(13) Thongsuwan, S.; Jaiyen, S.; Padcharoen, A.; Agarwal, P. ConvXGB: A new deep learning model for classification problems based on CNN and XGBoost. *Nucl. Eng. Technol.* **2021**, *53*, 522−531.

(14) Ren, X.; Guo, H.; Li, S.; Wang, S.; Li, J. A Novel Image Classification Method with CNN-XGBoost Model. *Digital Forensics and Watermarking* **2017**, *10431*, 378−390.

(15) Gao, J.; Shen, Z.; Xie, Y.; Lu, J.; Lu, Y.; Chen, S.; Bian, Q.; Guo, Y.; Shen, L.; Wu, J.; Zhou, B.; Hou, T.; He, Q.; Che, J.; Dong, X.; et al. TransFoxMol: predicting molecular property with focused attention. *Brief. Bioinform.* **2023**, *24*, bbad306.

(16) Walters, W. P.; Barzilay, R. Applications of deep learning in molecule generation and molecular property prediction. *Acc. Chem. Res.* **2021**, *54*, 263−270.

(17) Han, S.; Fu, H.; Wu, Y.; Zhao, G.; Song, Z.; Huang, F.; Zhang, Z.; Liu, S.; Zhang, W. HimGNN: a novel hierarchical molecular graph representation learning framework for property prediction. *Brief. Bioinform.* **2023**, *24*, bbad305.

(18) Cai, H.; Zhang, H.; Zhao, D.; Wu, J.; Wang, L. FP-GNN: a versatile deep learning architecture for enhanced molecular property prediction. *Brief. Bioinform.* **2022**, *23*, bbac408.

(19) Wang, Y.; Wang, J.; Cao, Z.; Barati Farimani, A. Molecular contrastive learning of representations via graph neural networks. *Nat. Mach. Intell.* **2022**, *4*, 279−287.

(20) Wang, Y.; Li, Y.; Chen, X.; Zhao, L. HIV-1/HBV Coinfection Accurate Multitarget Prediction Using a Graph Neural Network-Based Ensemble Predicting Model. *Int. J. Mol. Sci.* **2023**, *24*, 7139.

(21) Wang, Y.; Qi, J.; Chen, X. Accurate Prediction of Epigenetic Multi-Targets with Graph Neural Network-Based Feature Extraction. *Int. J. Mol. Sci.* **2022**, *23*, No. 13347.

(22) Yang, J.; Jiang, C.; Chen, J.; Qin, L.-P.; Cheng, G. Predicting GPR40 Agonists with A Deep Learning-Based Ensemble Model. *ChemOpen* **2023**, *12*, No. e202300051.

(23) Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular graph convolutions: moving beyond fingerprints. *J. Comput.-Aided Mol. Des.* **2016**, *30*, 595−608.

(24) Feng, H.; Wei, G.-W. Virtual screening of DrugBank database for hERG blockers using topological Laplacian-assisted AI models. *Comput. Biol. Med.* **2023**, *153*, No. 106491.
(25) Jiang, D.; Wu, Z.; Hsieh, C.-Y.; Chen, G.; Liao, B.; Wang, Z.; Shen, C.; Cao, D.; Wu, J.; Hou, T. Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models. *J. Cheminform.* **2021**, *13*, 1−23.
(26) Zhou, H.; Shan, M.; Qin, L.-P.; Cheng, G. Reliable prediction of cannabinoid receptor 2 ligand by machine learning based on combined fingerprints. *Comput. Biol. Med.* **2023**, *152*, No. 106379.
(27) Dou, B.; Zhu, Z.; Merkurjev, E.; Ke, L.; Chen, L.; Jiang, J.; Zhu, Y.; Liu, J.; Zhang, B.; Wei, G.-W. Machine learning methods for small data challenges in molecular science. *Chem. Rev.* **2023**, *123*, 8736−8780.
(28) Jiang, J.; Wang, R.; Wang, M.; Gao, K.; Nguyen, D. D.; Wei, G.-W. Boosting Tree-Assisted Multitask Deep Learning for Small Scientific Datasets. *J. Chem. Inf. Model.* **2020**, *60*, 1235−1244.
(29) Bashir, S. B.; Farag, M. M.; Hamid, A. K.; Adam, A. A.; Abo-Khalil, A. G.; Bansal, R. A Novel Hybrid CNN-XGBoost Model for Photovoltaic System Power Forecasting. *2024 6th International Youth Conference on Radio Electronics,Electrical and Power Engineering (REEPE)*, Cairo, Egypt, Feb 29−Mar 02, 2024; pp 1−6, .
(30) Shi, S.; Qiao, K.; Yang, J.; Song, B.; Chen, J.; Yan, B. RF-GNN: Random Forest Boosted Graph Neural Network for Social Bot Detection. *arXiv:2304.08239*, 2023.
(31) Deng, D.; Chen, X.; Zhang, R.; Lei, Z.; Wang, X.; Zhou, F. XGraphBoost: extracting graph neural network-based features for a better prediction of molecular properties. *J. Chem. Inf. Model.* **2021**, *61*, 2697−2705.
(32) Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016; Association for Computing Machinery: New York, NY, 785−794, .
(33) Liu, S.; Demirel, M. F.; Liang, Y. N-gram graph: Simple unsupervised representation for graphs, with applications to molecules. *Advances in Neural Information Processing Systems*; Advances in Neural Information Processing Systems 32 (NeurIPS 2019), 2019; Vol. *32*.
(34) Yang, M.; Jiang, H.; Yang, Z.; Liu, X.; Sun, H.; Hao, M.; Hu, J.; Chen, X.; Jin, J.; Wang, X. Design, synthesis, and biological evaluation of pyrrolopyrimidine derivatives as novel Brutons tyrosine kinase (BTK) inhibitors. *Eur. J. Med. Chem.* **2022**, *241*, No. 114611.
(35) Li, D.; Ru, Y.; Liu, J. GATBoost: Mining graph attention networks-based important substructures of polymers for a better property prediction. *Mater. Today Commun.* **2024**, *38*, No. 107577.
(36) Frosst, N.; Hinton, G. Distilling a neural network into a soft decision tree *arXiv:1711.09784*, 2017.
(37) Luo, H.; Cheng, F.; Yu, H.; Yi, Y. SDTR: Soft Decision Tree Regressor for Tabular Data. *IEEE Access* **2021**, *9*, 55999−56011.
(38) Silva, A.; Gombolay, M.; Killian, T.; Jimenez, I.; Son, S.-H. Optimization Methods for Interpretable Differentiable Decision Trees Applied to Reinforcement Learning. *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics* **2020**, *108*, 1855−1865.
(39) Graph Neural Tree: A novel and interpretable deep learning-based framework for accurate molecular property predictions. *Anal. Chim. Acta* **2023**, *1244*, 340558.
(40) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: a benchmark for molecular machine learning. *Chem. Sci.* **2018**, *9*, 513−530.
(41) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; et al. Analyzing learned molecular representations for property prediction. *J. Chem. Inf. Model.* **2019**, *59*, 3370−3388.
(42) Cho, K.; Van Merriënboer, B.; Bahdanau, D.; Bengio, Y. et al. Analyzing learned molecular representations for property prediction. *arXiv preprint arXiv:1409.1259*, 2014.
(43) Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1412.3555*, 2014.
(44) Manu Joseph, H. R. GANDALF: Gated Adaptive Network for Deep Automated Learning Learning of Features. *arXiv preprint arXiv:2207.08548*, 2023.
(45) Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.; Leskovec, J. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.
(46) Rong, Y.; Bian, Y.; Xu, T.; Xie, W.; Wei, Y.; Huang, W.; Huang, J. Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems (NeurIPS 2020)*, Vancouver, Canada, 2020; Vol. 33, pp 12559−12571.
(47) Zhou, G.; Gao, Z.; Ding, Q.; Zheng, H.; Xu, H.; Wei, Z.; Zhang, L.; Ke, G. Uni-Mol: a universal 3D molecular representation learning framework. *International Conference on Learning Representations*, 2023.
(48) Xiong, Z.; Wang, D.; Liu, X.; Zhong, F.; Wan, X.; Li, X.; Li, Z.; Luo, X.; Chen, K.; Jiang, H.; et al. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *J. Med. Chem.* **2020**, *63*, 8749−8760.
(49) Liu, S.; Wang, H.; Liu, W.; Lasenby, J.; Guo, H.; Tang, J. Pre-training Molecular Graph Representation with 3D Geometry. *International Conference on Learning Representations*, 2022.
(50) Fang, X.; Liu, L.; Lei, J.; He, D.; Zhang, S.; Zhou, J.; Wang, F.; Wu, H.; Wang, H. Geometry-enhanced molecular representation learning for property prediction. *Nat. Mach. Intell.* **2022**, *4*, 127−134.
(51) Huang, K.; Fu, T.; Gao, W.; Zhao, Y.; Roohani, Y.; Leskovec, J.; Coley, C. W.; Xiao, C.; Sun, J.; Zitnik, M. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *arXiv:2102.09548*, 2021.
(52) Heid, E.; Greenman, K. P.; Chung, Y.; Li, S.-C.; Graff, D. E.; Vermeire, F. H.; Wu, H.; Green, W. H.; McGill, C. J. Chemprop: a machine learning package for chemical property prediction. *J. Chem. Inf. Model.* **2024**, *64*, 9−17.
(53) Graves, A.; Graves, A. *Supervised sequence labelling*; Springer, 2012.
(54) Rogers, D.; Hahn, M. J. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.
(55) Van der Maaten, L.; Hinton, G. J. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579−2605.
(56) Xu, C.; Cheng, F.; Chen, L.; Du, Z.; Li, W.; Liu, G.; Lee, P. W.; Tang, Y. J. In silico prediction of chemical Ames mutagenicity. *J. Chem. Inf. Model.* **2012**, *52*, 2840−2847.
(57) Carbon-Mangels, M.; Hutter, M. C. Selecting relevant descriptors for classification by bayesian estimates: a comparison with decision trees and support vector machines approaches for disparate data sets. *Mol. Inform.* **2011**, *30*, 885−895.
(58) Veith, H.; Southall, N.; Huang, R.; James, T.; Fayne, D.; Artemenko, N.; Shen, M.; Inglese, J.; Austin, C. P.; Lloyd, D. G.; et al. Comprehensive characterization of cytochrome P450 isozyme selectivity across chemical libraries. *Nat. Biotechnol.* **2009**, *27*, 1050−1055.
(59) Ma, C.-Y.; Yang, S.-Y.; Zhang, H.; Xiang, M.-L.; Huang, Q.; Wei, Y.-Q. J. Prediction models of human plasma protein binding rate and oral bioavailability derived by using GACGSVM method. *J. Pharm. Biomed. Anal.* **2008**, *47*, 677−682.
(60) Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv:1710.10903*, 2017.
(61) Maziarka, Ł.; Danel, T.; Mucha, S.; Rataj, K.; Jastrzebski, S. Molecule attention transformer. *arXiv:2002.08264*, 2020.
(62) Sun, F.-Y.; Hoffmann, J.; Verma, V.; Tang, J. Infograph: Unsupervised and semisupervised graph-level representation learning via mutual information maximization. *arXiv:1908.01000*, 2019.
(63) Feurer, M.; Klein, A.; Eggensperger, K.; Springenberg, J.; Blum, M.; Hutter, F. Efficient and Robust Automated Machine Learning. *Advances in Neural Information Processing Systems 28 (NIPS 2015)*; Vol. *28*.

https://doi.org/10.1021/acs.jcim.4c01186
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

(64) Ying, C.; Cai, T.; Luo, S.; Zheng, S.; Ke, G.; He, D.; Shen, Y.; Liu, T.-Y. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, 2021; Vol. *34*, pp 28877−28888.
(65) Yang, Y.; Morillo, I. G.; Hospedales, T. M. Deep neural decision trees, *arXiv:1806.06988*, 2018.
(66) Schapire, R. The strength of weak learnability. *Machine Learning* **1990**, *5*, 197−227.
(67) Badirli, S.; Liu, X.; Xing, Z.; Bhowmik, A.; Keerthi, S. Gradient Boosting Neural Networks: GrowNet. *arXiv:2002.07971*, 2020.
(68) Popov, S.; Morozov, S.; Babenko, A. Gradient Boosting Neural Networks: GrowNet. *International Conference on Learning Representations*, 2020.

https://doi.org/10.1021/acs.jcim.4c01186
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX