Novel Molecular Representations using Neumann-Cayley Orthogonal Gated Recurrent Unit

Edison Mucllari,[†,‡] Vasily Zadorozhnyy,[†,‡] Qiang Ye,[*,†] and Duc Nguyen[*,†]

[†]*Department of Mathematics, University of Kentucky, Lexington, Kentucky, USA*

[‡]*These authors contributed equally*

E-mail: qye3@uky.edu; ducnguyen@uky.edu

Abstract

Advancements in Deep Neural Networks (DNNs) have made a very powerful machine learning method available to researchers across many fields of study, including the bio-medical and cheminformatics communities, where DNNs help to improve tasks such as protein performance, molecular design, drug discovery, etc. Many of those tasks rely on molecular descriptors for representing molecular characteristics in cheminformatics. Despite significant efforts and the introduction of numerous methods that derive molecular descriptors, quantitative prediction of molecular properties remains challenging. One widely used method of encoding molecule features into bit strings is molecular fingerprint. In this work, we propose using new Neumann-Cayley Gated Recurrent Units (NC-GRU) inside the Neural Nets encoder (AutoEncoder) to create neural molecular fingerprints, NC-GRU fingerprints. NC-GRU AutoEncoder introduces orthogonal weights into widely used GRU architecture, resulting in faster, more stable training, and more reliable molecular fingerprints. Integrating novel NC-GRU fingerprints and Multi-Task DNN schematics improves the performance of various molecular-related

tasks such as toxicity, partition coefficient, lipophilicity, and solvation-free energy, producing state-of-the-art results on several benchmarks.

1 Introduction

In recent years, there have been many advancements in drug discovery; however, building cheap and efficient compounds with desirable pharmacological and biochemical properties[1] remains challenging. Binding affinity, toxicity, and octanol-water partition coefficient (logP) are crucial properties needed to evaluate a drug candidate.[2] Drug discovery consists of several phases before launching to the market, such as target discovery, lead optimization, preclinical development, and three phases of clinical trial.[1] Unpleasant results on the toxicity and the pharmacokinetic properties are responsible for approximately half of drug candidates failing to reach the market.[3] In the past, some of the most popular experiments are conducted *in vivo* or *in vitro* to measure the drug properties. These approaches are very expensive and time-consuming, not to mention that testing with animals raise important ethical issues and concerns.

Machine Learning (ML) and Deep Learning (DL) algorithms have been introduced into drug discovery and have achieved much success recently. A significant amount of work has been devoted to deriving molecular descriptors from the representation of a molecule,[1,4,5] particularly molecular fingerprints that profile a molecule, usually in the form of a bit string or a vector, with each vector element indicating the existence, the degree, or the frequency of one particular structure feature.[6–8] Most molecular fingerprints are derived from either two-dimensional (2D)[6,9–13] or three-dimensional (3D)[14,15] molecular structural formulas where 2D structure can be viewed as if molecules were flat. Some of the most popular fingerprints are Molecular Access System (MACCS),[9] FP2,[10] Daylight,[11] Electro Topological State (Estate),[12] Extended-Connectivity Fingerprint(ECFP),[6] Extended Reduced Graph (ERG),[13] etc. DL methods have proven beneficial in obtaining valuable information about molecular

fingerprints.[4,5,16–19]

For example, 2D DL algorithms aim to learn a suitable data representation from a simple embedding layer, where the input is a one-hot vector[20] of each atom in a molecule. Such embedding is usually a part of the encoding mechanism where one of the widely used DL algorithms is AutoEncoder,[21] which learns the descriptors in an unsupervised and data-driven way.[4,5,18,19] In principle, AutoEncoder can take an arbitrary molecular representation/nomenclatures as an input; however, in practice, researchers are usually focused on sequence-based representations such as International Union of Pure and Applied Chemistry (IUPAC),[22] Simplified Molecular-Input Line-Entry System (SMILES),[23] International Chemical Identifier (InChI)[24] etc. A common AutoEncoder consists of Encoder and Decoder networks, where embedded input passes through the Encoder and outputs a latent representation. Then the Decoder network takes that latent vector and aims to transform it back into the input sequence of either the same or different nomenclature depending on the settings. The latent representation vector is associated with an information bottleneck between the Encoder and the Decoder. Since the information is compressed, the latent representation vector learns more general information related to the molecules.[19] The structure of AutoEncoder can be different; however, as mentioned in,[4] Gated Recurrent Unit (GRU)[25] is one of the most optimized architectures to implement as the AutoEncoder cells when compared with Long-Short Term Memory (LSTM)[26] or Convolutional Neural Network (CNN).[27] Lastly, DL methods benefit from a large number of training samples; that is why large training datasets such as ChEMBL,[28] ZINC15,[29] PubChem,[30] etc., allow DL models to derive better and more efficient molecular descriptors. Derived fingerprints can later be used on various prediction tasks such as toxicity prognosis, partition coefficient analysis, solubility predictions, etc., where the latent representation vector acts as the input corresponding to the prediction model.[1,4,31]

This work proposes a novel AutoEncoder equipped with Neumann-Cayley Orthogonal Gated Recurrent Units (NC-GRU)[32] to generate high-quality fingerprints for complex and

diverse molecules. To this end, we trained our NC-GRU AutoEncoder using the ChEMBL 28[28] dataset; see section 2.2 for more details about this dataset. Our AutoEncoder takes the Canonical SMILES representation of a molecule as input and outputs the same nomenclature back. The advantage of employing an NC-GRU architecture instead of a standard GRU cell is the ability of NC-GRU to capture long-term dependencies using the orthogonal matrices and the capability of GRU gates to forget unnecessary information. The combination of training AutoEncoder on the ChEMBL 28 dataset with NC-GRU cells results in NC-GRU FingerPrints (FPs), leading to improvements in many benchmarks during the inference phase. Indeed, using NC-GRU FPs has resulted in state-of-the-art outcomes on prediction tasks such as toxicity, partition coefficient, solubility, and solvation-free energy.

2 NC-GRU based AutoEncoder

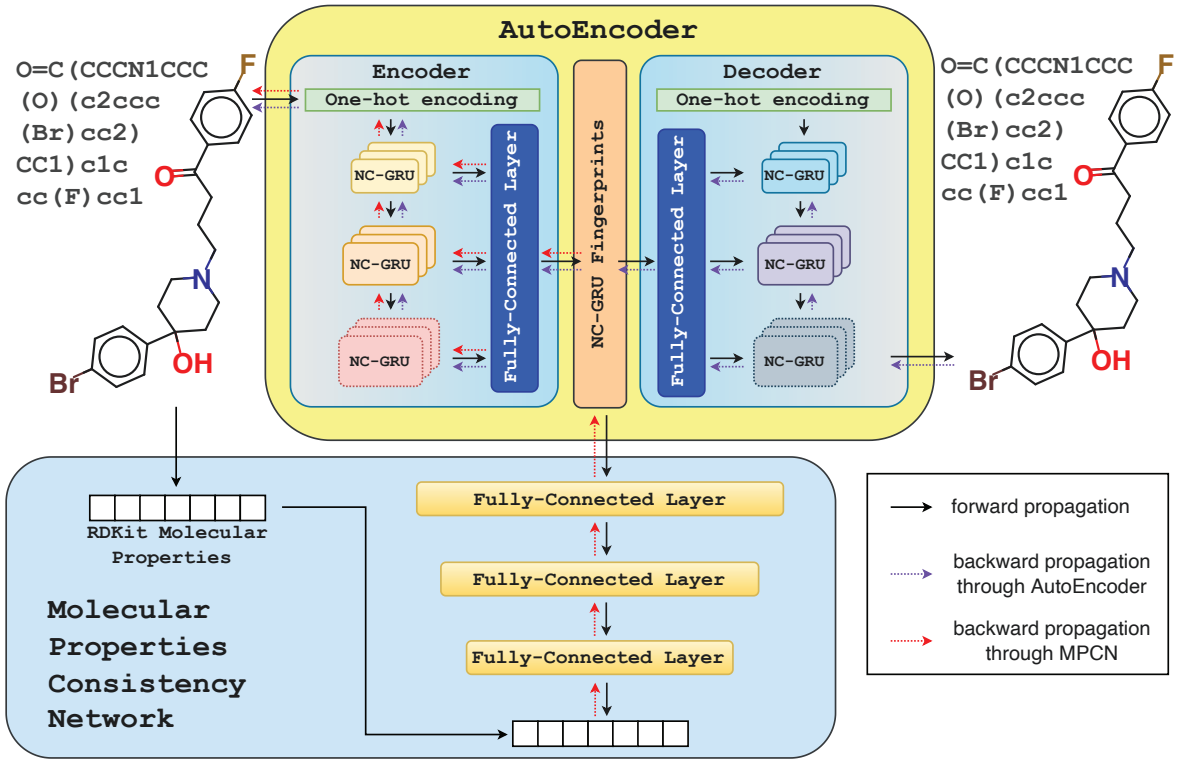
2.1 Architecture

In this work, our main focus is to apply NC-GRU[32] into the hidden layers of the AutoEncoder.[21] Figure 1b and 1c shows NC-GRU cell architecture and update diagram for the orthogonal weight $U_c(A_c)$, respectively. The input sequence-based molecular representation given as canonical SMILES is tokenized and encoded in a one-hot vector representation before we feed it to the AutoEncoder. Depending on the quality of the fine-tuning process, our AutoEncoder contains 2 or 3 stacked NC-GRU cells. If there are two cells, the hidden layer dimensions will be 160 and 320. In the 3-cell AutoEncoder, the third cell has a size of 640. Afterward, the state of each cell from the Encoder is concatenated and used as an input vector in a Fully-Connected Layer (FCL) with 512 neurons. The activation function applied to the FCL is a hyperbolic tangent (`tanh`). The extracted features vector with 512 units is implemented as the input vector in another FCL. The output of this FCL is divided into three parts, corresponding to each dimension from the encoder, and used as the initialization in every Decoder cell. At the same time, the extracted features vector is employed

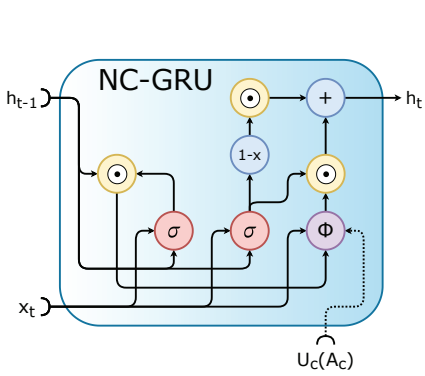
as the input to a Molecular Properties Consistency Network (MPCN), a prediction (regression) network with two FCLs of dimensions 512 and 128 with ReLU activation function after each, and the output FCL with seven neurons and no activation function. This extended Feed-Forward Neural Network (FNN) predicts specified molecular properties, namely logP, the Molar refractivity, Balaban's J-value, the number of acceptors, the number of hydrogen bond donors, the number of valence electrons, and the Topological polar surface area. These properties are derived from the molecular structure of the encoder input sequence using the RDKit Python library. The purpose of the classifier network is to act as a regularizer for the AutoEncoder and assist in obtaining better molecular descriptors from the trained AutoEncoder while still preserving the RDKit properties. The AutoEncoder needs to be trained to minimize the softmax cross-entropy between every input sequence and the Decoder output, $\mathcal{L}_{AutoEncoder}$ and, at the same time, minimize the Mean Squared Error associated with the MPCN, $\mathcal{L}_{MPCN}$.

$$\mathcal{L}_{total} = \mathcal{L}_{AutoEncoder} + \mathcal{L}_{MPCN} \tag{1}$$

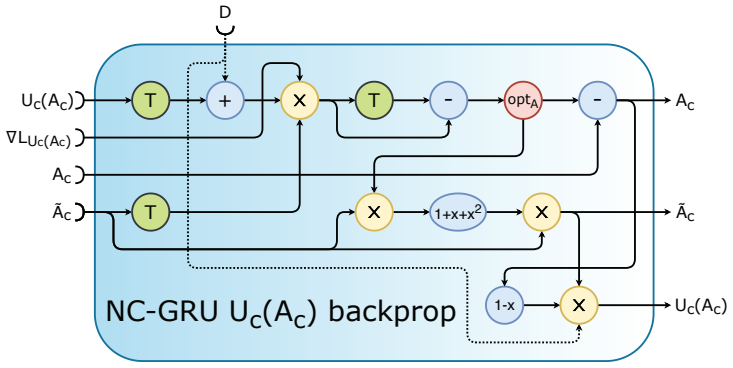
Besides our proposed NC-GRU AutoEncoder, we implement standard GRU[25] in AutoEncoder for comparison, the identical network where NC-GRU cells were replaced with GRU cells; see section 2.4 for more details. To better understand the AutoEncoder architecture described above, Figure 1a is provided as a visual aid. Due to the extra calculation steps for updating the orthogonal weight $U_c(A_c)$, the proposed NC-GRU AutoEncoder is slightly slower than its counterpart. Particularly, two-layer NC-GRU AutoEncoder takes 17.9 seconds to run 100,000 iterations, but only 11.8 seconds for two-layer GRU. Further comparisons between GRU and NC-GRU models, including computation time study, can be found in section 1 of the Supplementary Material.



(a) Architecture for training NC-GRU fingerprints using NC-GRU AutoEncoder together with Molecular Properties Consistency Network (MPCN)



(b) NC-GRU forward pass



(c) NC-GRU backpropagation of orthogonal $U_c(A_c)$ weight

Figure 1: Visualization of (a) AutoEncoder training process, (b) NC-GRU cell, and (c) NC-GRU weight $U_c(A_c)$ update scheme.
Notation: σ - sigmoid function, Φ - modReLU,[33] $\odot$ - Hadamard product (entrywise multiplication), T - transpose, $\times$ - matrix multiplication, opt_A - weight A optimizer, any algebraic expression (e.g., $1 - x$) is evaluated with previous step output as input (1 represents an identity matrix); refer to Algorithm 1[32] for the order of non-commutative operations

2.2 Data Processing

We have trained AutoEncoder architecture with Molecular Properties Consistency Network on the ChEMBL 28 dataset.[28] The RDKit Python library was used to process the ChEMBL 28 dataset. All the duplicates were removed, and the remaining molecules were filtered with the following criteria: only organic molecules, molecules with molecular weight between 12 and 600, molecules with at least three heavy atoms, molecules with a partition coefficient $\log P$ between -7 and 5, only non-stereochemistry molecules, no salts, and molecules that RDKit could not process were removed. The post-filtered dataset has 1,852,637 chemical compounds, split into training and testing sets of sizes 1,667,373 and 185,264, respectively. Furthermore, seven RDKit molecular properties were extracted for each molecule: $\log P$ (`MolLogP` in the RDKit), number of valence electrons (`NumValenceElectrons`), number of hydrogen bond donors (`NumHDonors`), number of acceptors (`NumHAcceptors`), Balaban's J-value (`BalabanJ`), molar refractivity (`MolMR`), and topological polar surface area (`TPSA`). Further, each of the above properties is normalized using

$$\hat{x} = \frac{x - \mu}{\sigma}, \tag{2}$$

where μ and σ represent the mean and standard deviation of the property for the whole dataset, and x and $\hat{x}$, represent each element of the dataset under that property and its normalized version, respectively. The above molecular properties were chosen to follow setups from[4] and.[18] These published works have testified that the constraints of RDKit features on the latent space are necessary to avoid dead areas in molecular representation learning. That causes the decoder network to produce invalid SMILES strings. Moreover, our experiments have shown that removing logP constraint in Autoencoder does not affect the quality of molecular representations, see Fig. 5a.

The data processing criteria and selected statistics about normalized RDKit molecular properties are provided in Table 1.

Table 1: ChEMBL 28 processing criteria and statistics of RDKit processed and normalized molecular properties

ChEMBL 28 Dataset			
Processing Criteria	**Normalized Molecular Properties**		
Removal of duplicates		min	max
Only organic molecules	$\log P$	-5.08	2.26
Molecular weight between 12 and 600	# of valence electrons	-3.46	3.16
More than three heavy atoms	# of hydrogen bond donors	-1.18	10.89
A partition coefficient $\log P$ between 7 and 5	# of acceptors	-2.47	7.86
Only non-stereochemistry compounds	Balaban's J-value	-2.53	12.86
No salts	Molar refractivity	-4.08	3.40
Molecules that RDKit could not process were removed	Topological polar surface area	-2.26	8.68

2.3 NC-GRU Fingerprints

Molecular descriptors are essential in chemoinformatics as they encode crucial chemical information of molecules in a computer-interpretable format.[34] Compared to the classical fingerprints, the advantage of the AutoEncoder models is the ability to learn a large and diverse set of molecules and yield encoded information in the latent space,[19] the desired fingerprints. In this work, we train the proposed NC-GRU AutoEncoder on the ChEMBL 28 dataset. The Decoder of NC-GRU AutoEncoder is asked to output the same Canonical SMILES representation as the one fed into the Encoder network. As mentioned, GRU[25] is one of the most optimized architectures for deriving neural fingerprints;[4] however, the newly proposed NC-GRU cell has better theoretical properties than GRU cell. At its core, the NC-GRU is equipped with orthogonality techniques to capture long-term dependencies. Still, at the same time, its gates help to forget the redundant information in the memory. Inheriting those advanced features, we expect our proposed NC-GRU FingerPrint (NC-GRU FP) to provide robust and reliable molecular descriptors ready for use on various applications. As a result, one expects the NC-GRU FP vector representation between the Encoder and Decoder to gain a more extensive understanding of the input molecule sequence.[19] To demonstrate the significance of our fingerprints, we tested NC-GRU FP on several prediction tasks, such as toxicity, solubility, partition coefficient, and solvation-free energy predictions, using seven

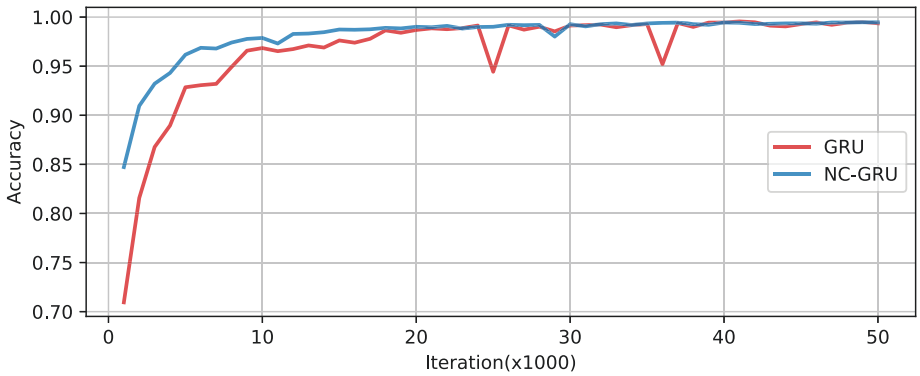


Figure 2: Comparison of Testing Accuracies between NC-GRU and GRU-based AutoEncoders on CheMBL 28 Dataset

benchmark datasets.

2.4 Translation Accuracy

To demonstrate the advantages of NC-GRU-based AutoEncoder compared to GRU-based one, we have analyzed training accuracies on ChEMBL 28. Both models were trained for 100,000 steps, with test accuracy recorded every 1,000.

We show the corresponding results in Figure 2, where the number of hidden layers for both AutoEncoders is two with dimensions 160 and 320 (similar performance was observed with three layers AutoEncoders). Thanks to the orthogonal gated units, we noticed a considerable improvement in training accuracy in early training when implementing NC-GRU AutoEncoder. However, in the later iterations, both models' accuracies approached 99%.

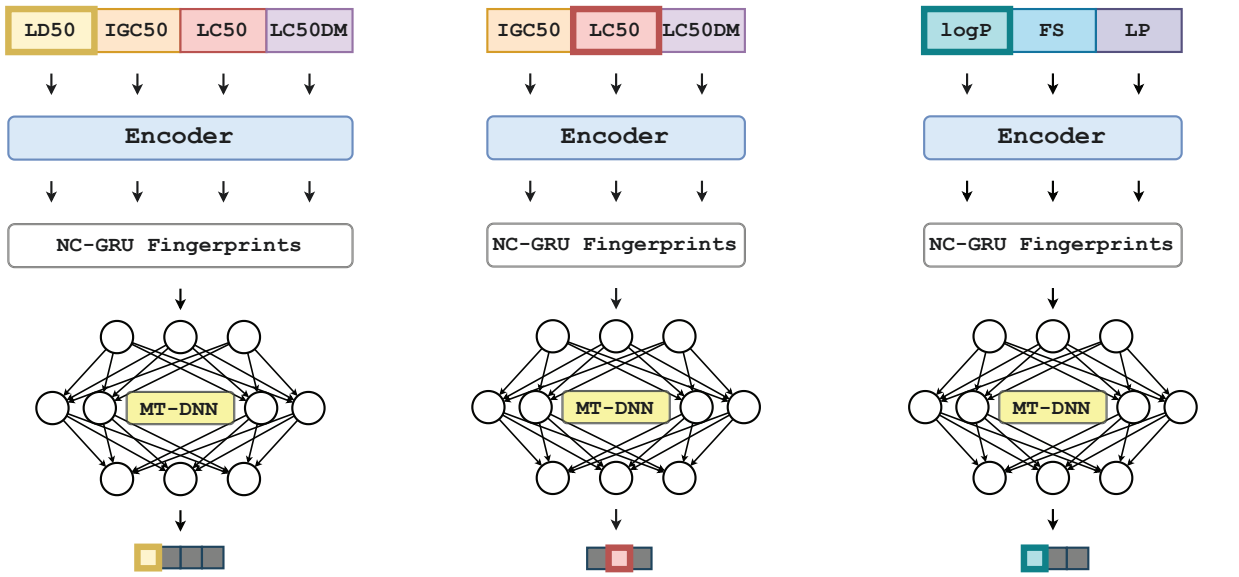
Based on our molecular property prediction tasks experiments, we believe that the earlier-faster convergence of NC-GRU-based AutoEncoder produces a more reliable and robust AutoEncoder fingerprint.

3 Prediction Models with Molecular Fingerprints

A prediction model or predictive modeling helps to predict future outcomes using input data by recognizing patterns within it. Many ML and DL algorithms have been very effective in

predictive tasks, including predicting molecular properties. The classical predicting models include linear and logistic regressions, logistic classification, k-nearest neighbors, support vector machine,[35] etc. More recently, ML and DL methods based on ideas from algebraic topology,[36,37] differential geometry,[38] geometric graph theory,[39,40] and algebraic graph theory[41] show promising results on predictive modeling. However, many advanced ML and DL algorithms use a combination of methods mentioned above with molecular fingerprints (FPs) to boost the performance and obtain a more accurate model, e.g., Random Forest (RF),[42] Gradient Boosting Decision Tree (GBDT),[43] Single-Task Deep Neural Networks (ST-DNN),[44] Multi-Task Deep Neural Networks (MT-DNN)[45] etc. These models use FPs as input into prediction models since they carry more structural information about molecules, particularly stereochemical descriptions, than chemical formulas or other not neural-FP representations. Such algorithms have often proven very efficient when employing either 2D or 3D molecular FPs.

In our prediction experiments, we employ the MT-DNN to improve the performance of molecular property prediction.



(a) LD50 and IGC50

(b) LC50 and LC50DM

(c) logP, FreeSolv (FS), and LipoPhilicity(LP)

Figure 3: MT-DNN models for prediction tasks;

3.1 Multitask Deep Neural Network

Multi-Task Deep Neural Network (MT-DNN)[46] is a powerful tool where a shared model simultaneously learns multiple tasks. MT-DNN has been utilized effectively in various applications, including computer vision,[47] speech recognition,[48] natural language processing,[49] and drug discovery.[2,50–53] The training process of MT-DNN consists of a joint representation of trainable parameters to gain knowledge from several tasks and boost performance. Its strength comes from learning multiple datasets simultaneously. However, MT-DNN depends heavily on the assumption that there is a correlation between the input datasets. Regarding the MT-DNN architecture, the number of neurons in the output layer depends on the number of tasks employed in the input data. Even though the output layer has more than one neuron, the loss function updates the parameters by focusing only on the particular output corresponding to the input task. For example, the MT-DNN model with four tasks starts training by taking a batch of data from the first task and updating the shared and first task output weights while not involving the other tasks' output weights. When it finishes with the first task's entire dataset (i.e., finishes the first task-epoch), the MT-DNN model moves to the second dataset and trains the shared and only the second task output weights again. Then the process continues with the third and then the fourth tasks. This process comprises a complete single epoch of MT-DNN training with four tasks. Note that a traditional MT-DNN model is trained using standard backpropagation algorithms such as Stochastic Gradient Descent (SGD), RMSProp,[54] and Adam[55] while using a single optimizer throughout all the training and tasks.

An illustration of the MT-DNN implemented in our experiments is given in Figure 3.

3.2 Prediction Datasets

We have used several datasets for toxicity, partition coefficient, solubility, and solvation-free energy prediction tasks.

Table 2: Selected Statistics for Prediction Tasks Datasets; " - " - no validation data; Part. Coeff. - Partition Coefficient.

Dataset	Train	Valid.	Test	Min. Value	Max. Value	Units	Category
LD50	7,413	-	1,482	0.291	7.201	$-\log_{10}$ mol/L	Toxicity
IGC50	1,434	-	358	0.334	6.36	$-\log_{10}$ mol/L	Toxicity
LC50	659	-	164	0.037	9.261	$-\log_{10}$ mol/L	Toxicity
LC50DM	283	-	70	0.117	10.064	$-\log_{10}$ mol/L	Toxicity
logP	8,199	-	406	-4.64	8.42	n/a	Part. Coeff.
FreeSolv	513	65	65	-25.47	3.43	kcal/mol	Free energy
Lipophilicity	3,360	420	420	-1.5	4.5	n/a	Solubility

3.2.1 Toxicity Prediction Datasets

Toxicology forecasting is crucial for public health. Toxicity prediction has various uses, but one of its most important is lowering the expense and labor of a medicine's preclinical and clinical trials. Many drug studies can be avoided because of the expected toxicity. In our toxicity prediction experiments, we have used four datasets: oral rate LD50 (LD50), 40 h Tetrahymenapyriformis IGC50 (IGC50), 96 h fathead minnow LC50 (LC50), and 48 h Daphnia Magna LC50DM (LC50DM).

The LD50[56,57] task measures the number of chemicals that can kill half of the rats when orally ingested. The IGC50[58,59] records the 50% growth inhibitory concentration of Tetrahymena pyriformis organism after 40 hours. The LC50[60,61] reports the concentration of test chemicals in the water in milligrams per liter that cause 50% of fathead minnows to die after 96 hours. The last toxicity prediction task, LC50DM,[60,61] represents the concentration of test chemicals in the water in milligrams per liter that cause 50% Daphna Magna to die after 48 hours. The unit of toxicity reported in these four datasets is $-\log_{10}$ moles per liter (mol/L). Among these four toxicity datasets, the sizes vary from 353 to 8,895; Table 2 provides more information about these datasets. Unfortunately, the small size of the dataset (LC50 and LC50DM) and some data being very uncertain $(\text{LD50})^2$ are only some of the reasons why training these datasets can be very challenging.

3.2.2 Partition Coefficient Prediction Dataset

For the task of Partition Coefficient Prediction, we have been working with the logP dataset. The term "logP" refers to the logarithm of a compound's octanol-water partition coefficient. The partition coefficient is the ratio of a compound's concentrations in an equilibrium two-phase system. It is a quantitative way to describe lipophilicity, the ability to dissolve, which impacts a pharmacological compound's absorption, distribution, metabolism, elimination, and toxicity. The logP dataset consists of 8,199 molecules for the training data, 406 molecules for the testing data, and no validation data; see Table 2 for more. The Food and Drug Administration (FDA) approved all the components in the test data as organic drugs. The logP values for the partition coefficient data are compiled by.[62]

3.2.3 Lipophilicity Prediction Dataset

The lipophilicity of a drug determines its potency, distribution, and elimination in the body. The current work's dataset is curated from the ChemBL database resulting in 4,200 compounds[63] where the lipophilicity index is determined by the distribution coefficient of octanol/water at pH 7.4.

3.2.4 Solvation Free Energy Prediction Dataset

Solvation-free energy transfers a solute molecule from an ideal gas to water. Therefore, accurately modeling solvation-free energy can give insight into the uncertainty of estimating binding free energy between small molecules and proteins. This is a significant area of interest for computer-aided drug discovery. The solvation-free energy data used in this work is FreeSolv, originally developed by Mobley and Guthrie,[64] containing 643 molecules. This set is divided into three sub-datasets in accordance with MoleculeNet's suggestions:[63] Train (513), Validation (568), and Test (65); some additional information is provided in Table 2.

3.2.5 Virtual Screening: Kinases dataset

This dataset was provided on request by Pogodin et al.[65] The Kinases dataset is a collection of SDFs divided into five subsets for 5-Fold Cross-Validation in every subset. It consists of 180,020 (175,076 after processing) compounds, with a number of unique ones being 55,594 (53,834 after processing). Every subset of this dataset was formed based on the data contained in the ChEMBL database, and the activities are measured on 160 different human-protein kinases. The ligands are classified as ATP-competitive and their score is recorded as active or inactive (1 or 0, respectively), depending on their inhibition rate. Furthermore, the data on the inhibition of non-human kinases was excluded from the dataset. Note that the dataset lacks activity information on many kinases since the ligands are only tested on a few. Despite several publications where the missing information is classified as inactive by default,[65,66] we only use the information provided in the dataset without any further assumptions or modifications of the dataset.

3.3 Optimized FingerPrints for Prediction Tasks

As mentioned in section 3.1, MT-DNN models can improve prediction tasks significantly. However, a nearly optimal AutoEncoder structure will deliver desirable molecular representations, further improving downstream prediction networks. This section discusses the process we have followed in choosing ideal NC-GRU AutoEncoders to get more desirable molecular FingerPrints (FPs) for the prediction task datasets and the following MT-DNN training.

The work done in the NC-GRU paper[32] suggests that different gate initializations and the number of layers can improve the performance of a model. Based on this argument, we have studied and analyzed a total of six AutoEncoder FingerPrint (FPs) extraction models. Four of which were based on the NC-GRU AutoEncoder with two and three layers and two different initializations (He Normal[67] and Glorot Uniform[68]), and the other two are GRU-based with two and three layers.

To choose a more desirable FP for a specific prediction task and future MT-DNN training, we have considered a Single-Task DNN (ST-DNN) model with simple fully-connected architecture. ST-DNN model consists of two fully-connected layers with dimensions 256, 128 for the prediction datasets with more than 1,000 data points and 128, 64 for the ones with less than 1,000 molecules. We have considered different sizes of ST-DNN models because of the problem with overfitting; the smaller datasets are more likely to overfit on structures of higher complexity,[69] and our experiments supported that. All ST-DNN models have trained for 1,000 epochs with Adam[55] optimizer, the learning rate of $5 \cdot 10^{-3}$, and the batch size of 32. We have trained 42 ST-DNN models using seven prediction task datasets and six pretrained AutoEncoders.

After ST-DNN models have finished training, we use 10-Fold Cross-Validation (CV) for the datasets without validation sets (LD50, IGC50, LC50, LC50DM, and logP) and validation sets for the ones with one (Lipophilicity and FreeSolv) to choose a suitable AutoEncoder FP extraction models by comparing the average r^2/RMSE values over ten independent runs of the ST-DNN models. See Table S1 in Supporting Information for these results. Table 3 summarizes the selected AutoeEncoder architecture for each benchmark.

Table 3: **NC-GRU/GRU Autoencoder hyperparameters**
The autoencoder for every dataset is selected using ten-fold cross-validation for all data except FreeSolv and Lipophilicity, where the respective validation data is used

	NC-GRU		**GRU**
Dataset	**Hidden sizes**	**Gate Init.**	**Hidden sizes**
IGC50	160, 320	He Normal	160,320, 640
LC50	160, 320	Glorot Uniform	160, 320, 640
LC50DM	160, 320, 640	He Normal	160, 320
LD50	160, 320, 640	He Normal	160, 320, 640
logP	160, 320	He Normal	160, 320, 640
FreeSolv	160, 320	He Normal	160, 320
Lipophilicity	160, 320, 640	He Normal	160, 320

3.4 MT-DNN Models: Hyperparameters and Setup

As mentioned before, there are many ML and DL algorithms that can be employed to learn various properties from a given molecular FP. In this work, MT-DNNs are the models applied to predict toxicity, partition coefficient, solubility, and solvation-free energy. The input size is set to 512, corresponding to the latent representation vector from the FP's extraction AutoEncoder models; see section 3.3. The MT-DNN models consist of four hidden layers with dimensions 1024, 512, 256, and 64, and the ReLU[70] activation function in between each hidden layer. All models were trained using a batch size of 18*, the SGD optimizer with a momentum of 0.5, an initial learning rate of 10^{-2}, and a step-learning rate decay, where the initial learning rate was used for the first 2,000 epochs and then reduced to 10^{-3} for the rest 1,000 epochs (total of 3,000 training epochs) except the FreeSolv dataset, where validation set was used to determine the termination of training criteria. Moreover, the Batch Normalization[71] was applied for every task to enhance the models' predictive power; a list of MT-DNN hyperparameters is provided in Table 4.

Table 4: MT-DNN Prediction Model hyperparameters

Hyperparam.	Input size	Hidden sizes	Learning rate	Optimizer	Momentum	Batch size
Values	512	(1024,512,256,64)	10^{-2}	SGD	0.5	18

As indicated in Gao *et al.*,[2] there is a physicochemical correlation between the toxicity datasets. Using this assumption, we have considered two MT-DNN models to train toxicity datasets. One for training LD50 and IGC50 predictors where we have used all of the toxicity data, and another one, for training LC50 and LC50DM where the LD50 dataset is not used; Figures 3a and 3b depict those models. The reason to exclude the LD50 dataset from the second model is the high uncertainties of the LD50 dataset,[2] which can potentially harm the learning process of MT-DNN when the test datasets are small. Similarly, logP, FreeSolv (FS), and Lipophilicity (LP) datasets have a chemical correlation, we use three of them

*such batch size was chosen to minimize the cutoff of data in the last batch, particularly important for the small datasets

altogether to implement the MT-DNN model; see Figure 3c.

Note that we do not implement any type of transfer learning on the pretrained AutoEncoders; we only get the FingerPrints. The Encoder part of the pretrained AutoEncoder was only used to obtain the molecular FingerPrints. Then, the obtained FingerPrints were fed into the prediction models.

4 Experiments

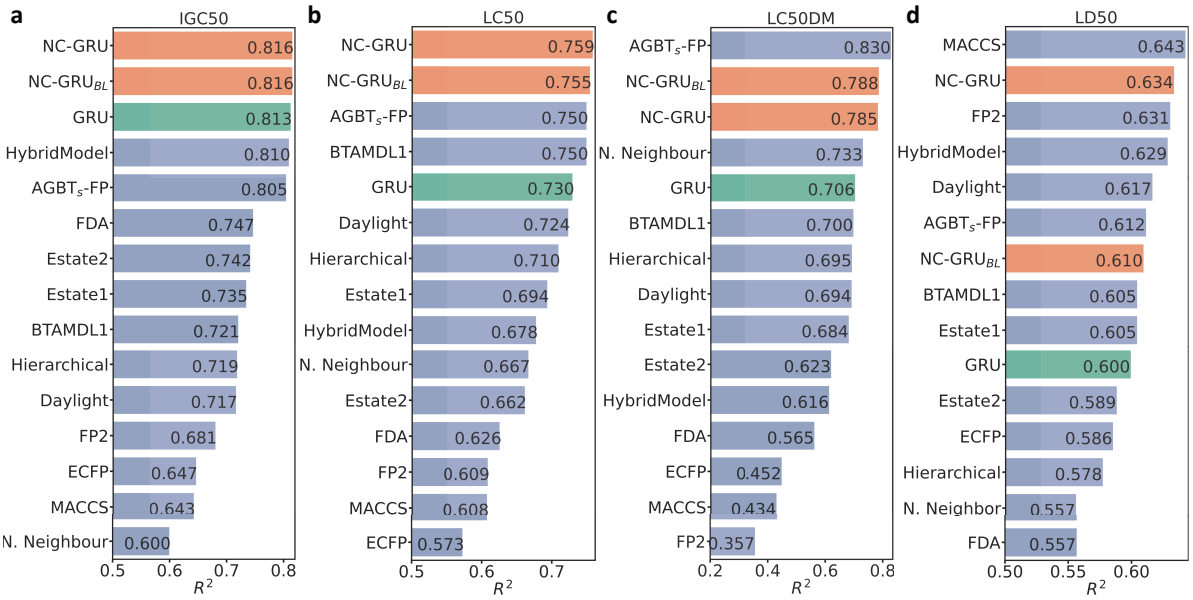


Figure 4: Performance comparison of different models on toxicity prediction tasks. Our proposed model in this work, NC-GRU and baseline NC-GRU$_{BL}$ (using uniform architectures and parameters) are highlighted in orange, the standard GRU-based model is in green, and the rest is in purple. The performance of the purple models is taken from previous studies.[2,5,56,72,73]

In this section, we present the results of various experiments to demonstrate the robustness and efficiency of the proposed NC-GRU FPs using four types of molecular properties: toxicity, partition coefficient, solubility, and solvation-free energy predictions where we have used seven benchmark datasets: IGC50, LC50DM, LC50, LD50, logP, Lipophilicity, and FreeSolv; see section 3.2 for details about these tasks and datasets. At the same time,

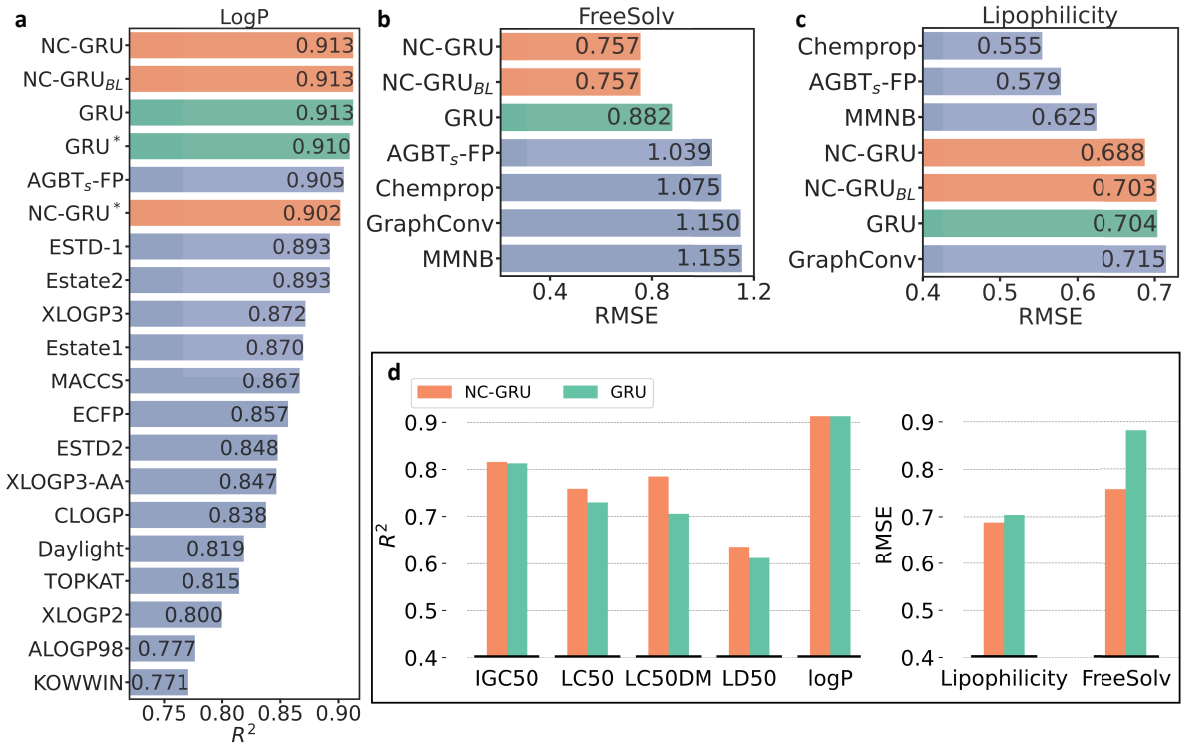


Figure 5: Results from NC-GRU, baseline NC-GRU$_{BL}$, using uniform architectures and parameters (in orange) and GRU (in green) models. a) Comparison of various models on the partition coefficient (logP) prediction, the other models in purple are taken from the literature;[2,5,62,74] and * indicates the logP constraint is not used in training the AutoEncoders. b) Illustrate the performances of different models on the solvation-free energy prediction on the FreeSolv dataset, RMSE values of other models are obtained from the previous studies.[5,63,75,76] c) Demonstrate the RMSE of several models on the Lipophilicity prediction, besides our models, the rest is based on the published work.[5,63,75,76] d) A summary of our NC-GRU and standard GRU performances on all considered benchmarks. Our proposed NC-GRU consistently outperforms its predecessor.

we compare with other available constructed models incorporating 2D/3D molecular FPs, including results for GRU-based FPs as a baseline to validate our proposed models. The accuracy of the models is measured in terms of the squared Pearson correlation coefficient (r^2) for all experiments, except for FreeSolv and Lipophilicity datasets, where the Root Mean Square Error (RMSE) is considered.

To reduce the variance in deep learning model performances, the presented NC-GRU and GRU results are the consensuses amidst five randomly selected seeds. The performance of

our models and others from the previous studies are illustrated in Figures 4 and 5. Our NC-GRU FP-based models demonstrate promising results, ranking first in three of seven experiments. Specifically, NC-GRU predictors achieve the best r^2 values on IGC50 (0.816) and LC50 (0.759) datasets. Our NC-GRU is still the best model in the solvation-free energy prediction task, with RMSE being 0.757 kcal/mol. On LC50DM and LD50 benchmarks, our NC-GRU is ranked in second place, where our r^2 coefficients are found to be 0.785 and 0.634, respectively. The top model on LC50DM is AGBT$_s$-FP[5] (0.830) and the best performance on LD50 is MACCS[2] (0.643). In the Lipophilicity dataset, our model is ranked fourth but still above the GRU model, with RMSE=0.688, while the first rank predictor is Chemprop[75] attaining RMSE=0.555.

In all the interested experiments, we include GRU FP-based models for a direct comparison with its successor, NC-GRU. As seen in Figure 5d, our NC-GRU outperforms GRU in all the benchmarks except for the logP task, where both models produce the same r^2=0.913. One might be concerned whether the information on logP constraint in MPCN significantly boosts the performance of the proposed fingerprint. We retrain the AutoEncoder network without the logP property to address that issue. As expected, we observe slightly reduced accuracy on both GRU and NC-GRU models. Specifically, while the R^2 of GRU decreased from 0.913 to 0.910, the one of NC-GRU went down from 0.913 to 0.902. Despite that, these performances remain at the top among state-of-the-art models, as shown in Fig. 5a. The NC-GRU can improve GRU as high as 14%, which is measured at the FreeSolv benchmark (NC-GRU RMSE=0.757 kcal/mol, GRU RMSE=0.882 kcal/mol). The superiority of NC-GRU over GRU for seven benchmarks illustrates the advantage of integrating Neumann-Cayley Gated Recurrent Units within the AutoEncoder architecture rather than standard gate components. Specifically, NC-GRU can store long-term information, which is crucial when encoding SMILES at various lengths. Furthermore, we conduct experiments on NC-GRU using the same MT-DNN as discussed before and the two-layer NC-GRU AutoEncoder with He Normal gate initialization across all seven datasets. This model is the baseline NC-

GRU, denoted as NC-GRU$_{BL}$. As shown in Figs. 4 and 5, despite slightly less accuracy than the NC-GRU that has been fine-tuned, the baseline has still performed top among state-of-the-art models.

Finally, we have conducted a similarity-based virtual screening experiment using the Kinases dataset. For each of the 160 human-protein kinases, we have concatenated all the provided ligands (since data came in a 5-Fold split for each protein) and then compared each concatenated ligand to the remaining concatenated ligands one protein at a time, i.e., leave-one-out similarity search. We use seven standard quality metrics to evaluate the results of virtual screening of kinase inhibitors: Recall, Specificity, Balanced Accuracy, Precision, Area Under the Receiver Operating Characteristics Curve (ROCAUC), and Enrichment Factor at 1% and 2.5%. Details about these metrics can be found in.[65,66] For each of the above metrics, the NC-GRU-based model obtains better results than the GRU-based. The average (over the 160 proteins) results of this experiment are provided in Figure 6 with an extended version in Supplementary Material.

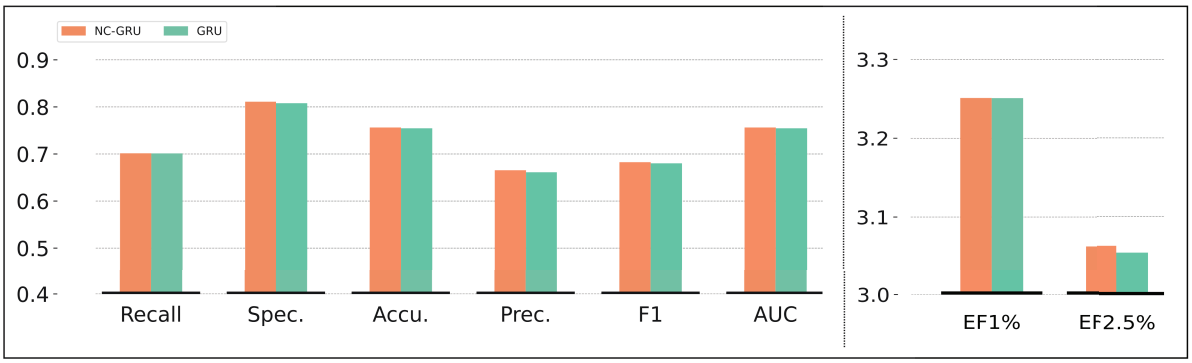


Figure 6: Results from similarity-based virtual screening for NC-GRU (in orange) and GRU (in green) models on Kinases dataset. Metrics: Recall, Spec. - Specificity, Accu. - Balanced Accuracy, Prec. - Precision, ROCAUC - Area Under the Receiver Operating Characteristics (ROC) Curve, EF·% - Enrichment Factor

5 Conclusion

A fingerprint-based AutoEncoder, commonly equipped with the Gated Recurrent Unit (GRU), is reported to provide a reliable molecular representation for the downstream task of predicting molecular properties. However, due to the exploding gradient issue and long-term dependence limitation, the GRU-AutoEncoder frameworks fail to achieve state-of-the-art accuracy when handling diverse biological datasets. This problem motivated us to develop an advanced GRU version, named NC-GRU, for the AutoEncoder to encode small molecular structures more efficiently by training orthogonal matrices.

Combined with multitasking deep neural networks (MT-DNN), our NC-GRU fingerprint-based models achieve promising results in predicting various molecular properties, namely toxicity, partition coefficient, lipophilicity, and solvation-free energy. Specifically, our proposed models earned the top ranking in four of seven benchmark studies: IGC50, LC50, logP, and FreeSolv. NC-GRU still performed well in the other two data sets, LC50DM and LD50, ranking second overall. Furthermore, it is encouraging to observe that NC-GRU models outperformed GRU versions in almost every experiment. State-of-the-art performances indicate that the newly developed fingerprints and their corresponding predictors could be used in various drug discovery applications.

Conflict of interest

The authors declare that they have no conflict of interest.

Data and Software Availability

The source code is available at GitHub: github.com/Edison1994/NC-GRU-molecular-representation

Acknowledgement

We would like to thank the Computational Sciences and College of Art and Science at the University of Kentucky for their support, the use of the Computing Clusters, and associated research computing resources. This research was supported in parts by NSF under grants DMS-1821144, DMS-2053284, DMS-2151802, DMS-2208314, and the University of Kentucky Start-up Fund.

Supporting Information Available

Document supplementary.pdf contains

- Supporting Table S1 provides the cross-validation/validation results from ST-DNN models for both NC-GRU and GRU frameworks.
- Supporting Table S2 provides the detailed performances of various models on toxicity data.
- Supporting Table S3 provides the detailed performances of various models on logP data.
- Supporting Table S4 provides the detailed performances of various models on FreeSolv and Lipophilicity data.
- Supporting Table S5 provides the results of applying a two-layer NC-GRU AutoEncoder with He Normal gate initialization to obtain the molecular fingerprints across seven datasets.
- Supporting Table S6 provides the Similarity-based Virtual Screening on Kinases Data Results.

- Supporting Figure S1 shows the visual representation of the GRU and NC-GRU architecture including forward propagation for both models and rules for updating weight U_c and $U_c(A_c)$ for GRU and NC-GRU models, respectively.

- Supporting Figure S2 provides information related to the computational time comparison between GRU and NC-GRU AutoEncoders.

- Supporting Figure S3 provides the comparison plots of the predicted data points of NC-GRU and GRU models vs. target points for each prediction dataset. In addition, the mean absolute and the mean square residual errors are provided.

References

(1) Gao, K.; Nguyen, D. D.; Tu, M.; Wei, G.-W. Generative network complex for the automated generation of drug-like molecules. *Journal of chemical information and modeling* **2020**, *60*, 5682–5698.

(2) Gao, K.; Nguyen, D. D.; Sresht, V.; Mathiowetz, A. M.; Tu, M.; Wei, G.-W. Are 2D fingerprints still valuable for drug discovery? *Physical chemistry chemical physics* **2020**, *22*, 8373–8390.

(3) Van De Waterbeemd, H.; Gifford, E. ADMET in silico modelling: towards prediction paradise? *Nature reviews Drug discovery* **2003**, *2*, 192–204.

(4) Winter, R.; Montanari, F.; Noé, F.; Clevert, D.-A. Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chemical science* **2019**, *10*, 1692–1701.

(5) Chen, D.; Gao, K.; Nguyen, D. D.; Chen, X.; Jiang, Y.; Wei, G.-W.; Pan, F. Algebraic graph-assisted bidirectional transformers for molecular property prediction. *Nature Communications* **2021**, *12*, 1–9.

(6) Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *Journal of chemical information and modeling* **2010**, *50*, 742–754.

(7) Lo, Y.-C.; Rensi, S. E.; Torng, W.; Altman, R. B. Machine learning in chemoinformatics and drug discovery. *Drug discovery today* **2018**, *23*, 1538–1546.

(8) Cereto-Massagué, A.; Ojeda, M. J.; Valls, C.; Mulero, M.; Garcia-Vallvé, S.; Pujadas, G. Molecular fingerprint similarity search in virtual screening. *Methods* **2015**, *71*, 58–63.

(9) Durant, J. L.; Leland, B. A.; Henry, D. R.; Nourse, J. G. Reoptimization of MDL keys for use in drug discovery. *Journal of chemical information and computer sciences* **2002**, *42*, 1273–1280.

(10) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An open chemical toolbox. *Journal of cheminformatics* **2011**, *3*, 1–14.

(11) Toolkit, D. Daylight chemical information systems. *Inc.: Aliso Viejo, CA* **2007**,

(12) Hall, L. H.; Kier, L. B. Electrotopological state indices for atom types: a novel combination of electronic, topological, and valence state information. *Journal of Chemical Information and Computer Sciences* **1995**, *35*, 1039–1045.

(13) Stiefl, N.; Watson, I. A.; Baumann, K.; Zaliani, A. ErG: 2D pharmacophore descriptions for scaffold hopping. *Journal of chemical information and modeling* **2006**, *46*, 208–220.

(14) Verma, J.; Khedkar, V. M.; Coutinho, E. C. 3D-QSAR in drug design-a review. *Current topics in medicinal chemistry* **2010**, *10*, 95–115.

(15) Nguyen, D. D.; Cang, Z.; Wei, G.-W. A review of mathematical representations of biomolecular data. *Physical Chemistry Chemical Physics* **2020**, *22*, 4343–4367.

(16) Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems* **2015**, *28*.

(17) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B. P.; Palmer, A.; Settels, V., et al. Are learned molecular representations ready for prime time? **2019**,

(18) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science* **2018**, *4*, 268–276.

(19) Xu, Z.; Wang, S.; Zhu, F.; Huang, J. Seq2seq fingerprint: An unsupervised deep molecular embedding for drug discovery. Proceedings of the 8th ACM international conference on bioinformatics, computational biology, and health informatics. 2017; pp 285–294.

(20) Harris, D.; Harris, S. *Digital Design and Computer Architecture, Second Edition*, 2nd ed.; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2012.

(21) Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. *Learning internal representations by error propagation*; 1985.

(22) Favre, H.; Powell, W. *Nomenclature of Organic Chemistry: IUPAC Recommendations and Preferred Names 2013*; International Union of Pure and Applied Chemistry; Royal Society of Chemistry, 2014.

(23) Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences* **1988**, *28*, 31–36.

(24) Heller, S.; McNaught, A.; Stein, S.; Tchekhovskoi, D.; Pletnev, I. InChI - the worldwide chemical structure identifier standard. *Journal of Cheminformatics* **2013**, *5*, 7.

(25) Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. 2014; `https://arxiv.org/abs/1406.1078`.

(26) Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural computation* **1997**, *9*, 1735–1780.

(27) LeCun, Y.; Haffner, P.; Bottou, L.; Bengio, Y. *Shape, contour and grouping in computer vision*; Springer, 1999; pp 319–345.

(28) Gaulton, A.; Hersey, A.; Nowotka, M.; Bento, A. P.; Chambers, J.; Mendez, D.; Mutowo, P.; Atkinson, F.; Bellis, L. J.; Cibrián-Uhalte, E., et al. The ChEMBL database in 2017. *Nucleic acids research* **2017**, *45*, D945–D954.

(29) Irwin, J. J.; Shoichet, B. K. ZINC - A Free Database of Commercially Available Compounds for Virtual Screening. *Journal of Chemical Information and Modeling* **2005**, *45*, 177–182.

(30) Kim, S.; Thiessen, P. A.; Bolton, E. E.; Chen, J.; Fu, G.; Gindulyte, A.; Han, L.; He, J.; He, S.; Shoemaker, B. A.; Wang, J.; Yu, B.; Zhang, J.; Bryant, S. H. PubChem Substance and Compound databases. *Nucleic Acids Research* **2015**, *44*, D1202–D1213.

(31) Santana, M. V.; Silva-Jr, F. P. De novo design and bioactivity prediction of SARS-CoV-2 main protease inhibitors using recurrent neural network-based transfer learning. *BMC chemistry* **2021**, *15*, 1–20.

(32) Mucllari, E.; Zadorozhnyy, V.; Pospisil, C.; Nguyen, D.; Ye, Q. Orthogonal Gated Recurrent Unit with Neumann-Cayley Transformation. 2022; `https://arxiv.org/abs/2208.06496`.

(33) Arjovsky, M.; Shah, A.; Bengio, Y. Unitary Evolution Recurrent Neural Networks. Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48. 2016; p 1120–1128.

(34) Todeschini, R.; Consonni, V. *Handbook of molecular descriptors*; John Wiley & Sons, 2008.

(35) Cortes, C.; Vapnik, V. Support-vector networks. *Machine learning* **1995**, *20*, 273–297.

(36) Cang, Z.; Wei, G.-W. Integration of element specific persistent homology and machine learning for protein-ligand binding affinity prediction. *International journal for numerical methods in biomedical engineering* **2018**, *34*, e2914.

(37) Cang, Z.; Mu, L.; Wei, G.-W. Representability of algebraic topology for biomolecules in machine learning based scoring and virtual screening. *PLoS computational biology* **2018**, *14*, e1005929.

(38) Nguyen, D. D.; Wei, G.-W. DG-GL: Differential geometry-based geometric learning of molecular datasets. *International journal for numerical methods in biomedical engineering* **2019**, *35*, e3179.

(39) Nguyen, D. D.; Xiao, T.; Wang, M.; Wei, G.-W. Rigidity strengthening: A mechanism for protein–ligand binding. *Journal of chemical information and modeling* **2017**, *57*, 1715–1721.

(40) Bramer, D.; Wei, G.-W. Multiscale weighted colored graphs for protein flexibility and rigidity analysis. *The Journal of chemical physics* **2018**, *148*, 054103.

(41) Nguyen, D. D.; Cang, Z.; Wu, K.; Wang, M.; Cao, Y.; Wei, G.-W. Mathematical deep learning for pose and binding affinity prediction and ranking in D3R Grand Challenges. *Journal of computer-aided molecular design* **2019**, *33*, 71–82.

(42) Svetnik, V.; Liaw, A.; Tong, C.; Culberson, J. C.; Sheridan, R. P.; Feuston, B. P. Random forest: a classification and regression tool for compound classification and QSAR modeling. *Journal of chemical information and computer sciences* **2003**, *43*, 1947–1958.

(43) Schapire, R. E. The boosting approach to machine learning: An overview. *Nonlinear estimation and classification* **2003**, 149–171.

(44) Basheer, I. A.; Hajmeer, M. Artificial neural networks: fundamentals, computing, design, and application. *Journal of microbiological methods* **2000**, *43*, 3–31.

(45) Caruana, R. Multitask learning. *Machine learning* **1997**, *28*, 41–75.

(46) Caruana, R. Multitask Learning: A Knowledge-Based Source of Inductive Bias. ICML. 1993.

(47) Girshick, R. Fast r-cnn In: Proceedings of the IEEE international conference on computer vision pp 1440-1448 https://doi. org/10.1109. 2015.

(48) Deng, L.; Hinton, G.; Kingsbury, B. New types of deep neural network learning for speech recognition and related applications: An overview. 2013 IEEE international conference on acoustics, speech and signal processing. 2013; pp 8599–8603.

(49) Collobert, R.; Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. Proceedings of the 25th international conference on Machine learning. 2008; pp 160–167.

(50) Capuzzi, S. J.; Politi, R.; Isayev, O.; Farag, S.; Tropsha, A. QSAR Modeling of Tox21 Challenge Stress Response and Nuclear Receptor Signaling Toxicity Assays. *Frontiers in Environmental Science* **2016**, *4*.

(51) Ramsundar, B.; Liu, B.; Wu, Z.; Verras, A.; Tudor, M.; Sheridan, R. P.; Pande, V. Is

Multitask Deep Learning Practical for Pharma? *Journal of Chemical Information and Modeling* **2017**, *57*, 2068–2076, PMID: 28692267.

(52) Wenzel, J.; Matter, H.; Schmidt, F. Predictive Multitask Deep Neural Network Models for ADME-Tox Properties: Learning from Large Data Sets. *Journal of Chemical Information and Modeling* **2019**, *59*, 1253–1268.

(53) Ye, Z.; Yang, Y.; Li, X.; Cao, D.; Ouyang, D. An Integrated Transfer Learning and Multitask Learning Approach for Pharmacokinetic Parameter Prediction. *Molecular Pharmaceutics* **2019**, *16*, 533–541.

(54) Tieleman, T.; Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning* **2012**, *4*.

(55) Kingma, D.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* **2014**,

(56) Martin, T., et al. User's guide for TEST (version 4.2)(Toxicity Estimation Software Tool) A program to estimate toxicity from molecular structure. *Washington (USA): US-EPA* **2016**, *505*.

(57) Naltional Library of Medicine. `https://chem.nlm.nih.gov/chemidplus/`.

(58) Akers, K. S.; Sinks, G. D.; Schultz, T. W. Structure–toxicity relationships for selected halogenated aliphatic chemicals. *Environmental toxicology and pharmacology* **1999**, *7*, 33–39.

(59) Zhu, H.; Tropsha, A.; Fourches, D.; Varnek, A.; Papa, E.; Gramatica, P.; Oberg, T.; Dao, P.; Cherkasov, A.; Tetko, I. V. Combinatorial QSAR modeling of chemical toxicants tested against Tetrahymena pyriformis. *Journal of chemical information and modeling* **2008**, *48*, 766–784.

(60) Martin, T. M.; Young, D. M. Prediction of the acute toxicity (96-h LC50) of organic compounds to the fathead minnow (Pimephales promelas) using a group contribution method. *Chemical Research in Toxicology* **2001**, *14*, 1378–1385.

(61) ECOTOX. `https://cfpub.epa.gov/ecotox/`.

(62) Cheng, T.; Zhao, Y.; Li, X.; Lin, F.; Xu, Y.; Zhang, X.; Li, Y.; Wang, R.; Lai, L. Computation of octanol- water partition coefficients by guiding an additive model with knowledge. *Journal of chemical information and modeling* **2007**, *47*, 2140–2148.

(63) 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. *Chemical science* **2018**, *9*, 513–530.

(64) Mobley, D. L.; Guthrie, J. P. FreeSolv: a database of experimental and calculated hydration free energies, with input files. *Journal of computer-aided molecular design* **2014**, *28*, 711–720.

(65) Pogodin, P. V.; Lagunin, A. A.; Rudik, A. V.; Filimonov, D. A.; Druzhilovskiy, D. S.; Nicklaus, M. C.; Poroikov, V. V. How to achieve better results using PASS-based virtual screening: Case study for kinase inhibitors. *Frontiers in chemistry* **2018**, *6*, 133.

(66) Menke, J.; Koch, O. Using domain-specific fingerprints generated through neural networks to enhance ligand-based virtual screening. *Journal of Chemical Information and Modeling* **2021**, *61*, 664–675.

(67) He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Proceedings of the IEEE international conference on computer vision. 2015; pp 1026–1034.

(68) Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural

networks. Proceedings of the thirteenth international conference on artificial intelligence and statistics. 2010; pp 249–256.

(69) Nowlan, S. J.; Hinton, G. E. Simplifying Neural Networks by Soft Weight-Sharing. *Neural Computation* **1992**, *4*, 473–493.

(70) Fukushima, K. Cognitron: A self-organizing multilayered neural network. *Biological cybernetics* **1975**, *20*, 121–136.

(71) Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37. 2015; p 448–456.

(72) Jiang, J.; Wang, R.; Wang, M.; Gao, K.; Nguyen, D. D.; Wei, G.-W. Boosting tree-assisted multitask deep learning for small scientific datasets. *Journal of chemical information and modeling* **2020**, *60*, 1235–1244.

(73) Karim, A.; Mishra, A.; Newton, M. H.; Sattar, A. Efficient toxicity prediction via simple features using shallow neural networks and decision trees. *Acs Omega* **2019**, *4*, 1874–1888.

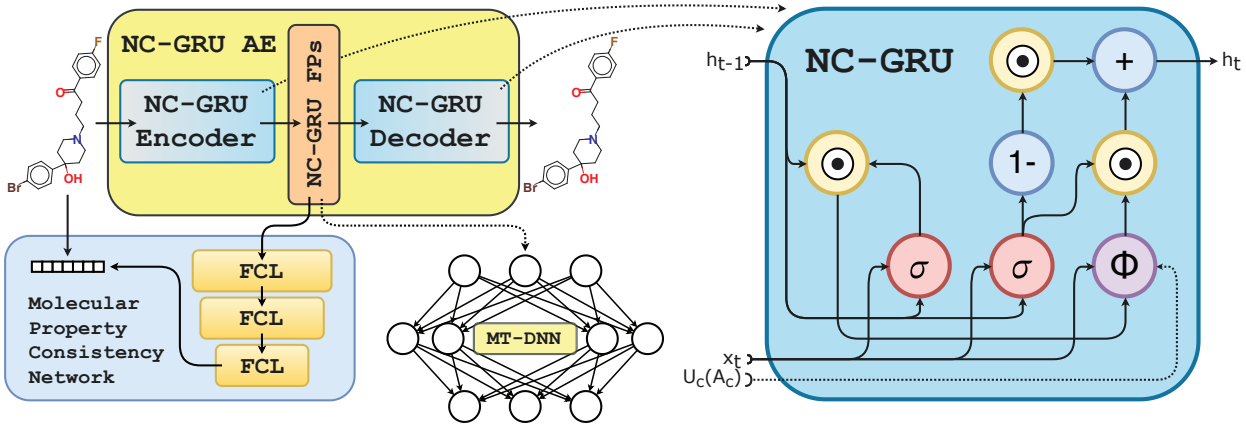
(74) Wu, K.; Zhao, Z.; Wang, R.; Wei, G.-W. TopP–S: Persistent homology-based multi-task deep neural networks for simultaneous predictions of partition coefficient and aqueous solubility. *Journal of computational chemistry* **2018**, *39*, 1444–1454.

(75) 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. *Journal of chemical information and modeling* **2019**, *59*, 3370–3388.

(76) Shen, W. X.; Zeng, X.; Zhu, F.; Qin, C.; Tan, Y.; Jiang, Y. Y.; Chen, Y. Z., et al. Out-of-the-box deep learning prediction of pharmaceutical properties by broadly learned

knowledge-based molecular representations. *Nature Machine Intelligence* **2021**, *3*, 334–343.

TOC Graphic


NC-GRU AE
NC-GRU Encoder
NC-GRU FPs
NC-GRU Decoder
FCL
FCL
FCL
Molecular Property Consistency Network
MT-DNN
NC-GRU
h_{t-1}
h_t
x_t
$U_c(A_c)$
1-
σ
σ
Φ
+